Volume 42 Number 3 September 2018 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Information and Communication Technology

Guest Editors:

Huynh Thi Thanh Binh

Ichiro Ide

1977

Editorial Boards

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatica is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatica is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

https://doi.org/10.31449/inf.v42i3.2465 Informatica **42** (2018) 285–289 **285**

IJCAI 2018 – Chinese Dominance Established

Matjaž Gams
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: matjaz.gams@ijs.si

Editorial

In July 2018 in Stockholm, ICML, AAMAS, ICCBR and SoCS joined with IJCAI and ECAI to establish the first major worldwide AI event. This paper is about the resulting IJCAI-ECCAI event [1].

The 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence merged with the other events to form a single conference. Around 7000 participants divided their time between these conferences over 14 days as one fee covered the entrance to all the events. As a consequence, several researchers attended several conferences, which in itself was a major achievement. Namely, the conferences and even the individual sections of the conferences are becoming so specialized that AI researchers are becoming oblivious to the achievements being made in a related area, leading to specialization and small incremental improvements, thereby deterring major innovations. Fortunately, in 2018 there was a serious attempt to reintegrate the field.

For the organizers, the super-event joint conferences represented a huge effort, but everything ran smoothly – albeit with a couple of small exceptions, as usual. One of them was the initial robot dance, where a Nao robot performed a predefined sequence of moves, which the human dancer enriched with dynamics and scope. The glitch was a loss of sound during the event (deliberate or by accident?). Added to this, the lack of any AI in the performance was a huge issue for many of the participants; in particular, the absence of true AI, one of the central themes of the conference. However, the artistic impression was there. Perhaps not surprisingly, as the small Nao robot was clearly physically and dynamically very much inferior to the flexible human dancer, a kind of reverse of David and Goliath seemed to be taking place. Also, the big 1000+ lecture rooms were organized in such a way that at no time was everybody sitting down, instead there seemed to be 5-10 people in motion at any moment. That aside, Stockholm is a traditional, open, metropolitan city that has hosted conferences for up to several tens of thousands of participants before, and the AI organizers have extensive experience as well; so by any measure the event must be considered as an organizational success.

The IJCAI-ECCAI joint event involved a record 3470 submitted papers: 37 % more than in 2017, while the 2017 event was 11% up on the previous year, confirming the steady growth from 2007 on. AI continues to progress as a scientific field and as an area of human interest.

The first major technical impression in 2018 was that Chinese dominance has finally been established. Eclatantly! In 2017, 37% of the papers were Chinese, while a year later this figure was 46%. Only 9% increase, one might say, but the 2017 conference was in Australia, with strong Chinese ties, while Stockholm is in Europe, and it was a joint European and international IJCAI conference, meaning around half of the event was basically a European conference. Despite that, European and American papers constituted around 20% each, while several authors, in particular from the USA, were also Chinese. Astonishment and admiration are the right words to describe this Chinese success.

The more detailed numbers are as follows: from the 710 accepted papers (21% acceptance rate), 325 came from China, 129 from EU (UK 37, France 22, Italy 18, Germany 15, Austria 12), USA 122, Singapore 26, Australia 23, Japan 17, Israel 13, etc.

When asked if it is reasonable to limit non-European papers at least for the ECCAI conference, say to 50%, several of the researchers expressed concern that it would mean that several of the best non-European papers would then end up being presented at other conferences. Several of the Chinese papers were indeed of high quality, demonstrating Chinese innovation, good education and the major support for AI in China. There were some concerns that the Chinese papers often follow a pattern with a specific idea, lots of complicated mathematics and an unverified empirical comparison. But that is true for many other papers as well. It should be noted, in addition, that due to several national European research policies, it is often nearly worthless for domestic evaluations to publish a paper at IJCAI or ECAI, since all that counts for these researchers are journal publications. The absence of more high-quality

European papers might therefore be partially attributed to the strange European scientific policies. Some of these issues were discussed at the panels, as presented in Figures 1 and 2.

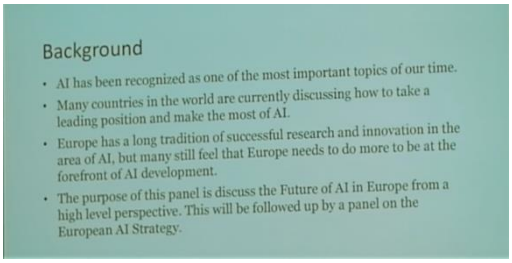


Figure 1: At a panel about European AI, the importance of the field for European progress was clearly established.

In some presentation, e.g., the one shown in Figure 1, the European position and the self-evaluation were not exactly in accordance with the percentage of conference papers. Some other positions even sounded a bit like a clip from a galaxy far, far away. But in reality, the panels were of high quality and several essential issues and initiatives were raised. Several panelists mentioned that there is no AI coordination in Europe, even though the EU is still the no. 1 world economy. In terms of AI funding, the USA prevails over China 2:1, and China prevails over the EU, again by 2:1. Such estimates might be misleading since the nominal comparison took place – instead, real economy (how many kilograms of sugar or of steel) already puts China above USA in terms of scientific funding.

There are two important differences between the USA and the EU: the USA executes bold international policies, whereas the EU finds its soft approach is sometimes hurting its economy and society. The EU used to be no.1 in computer science; now it is no. 3. Lots of this falling behind was not necessary at all; instead there are subjective leadership reasons for the decline, e.g., the EU patent system is enormously complicated and bureaucratic compared to the American one. Another problem: the UK has the best European AI based on many criteria, and so Brexit will make this situation worse for the EU. Whereas top EU projects like H2020 represent world-class research, and the EU is still leading in many areas of business and science, the strong scientific funding for key areas as well as policies to support them are lacking.

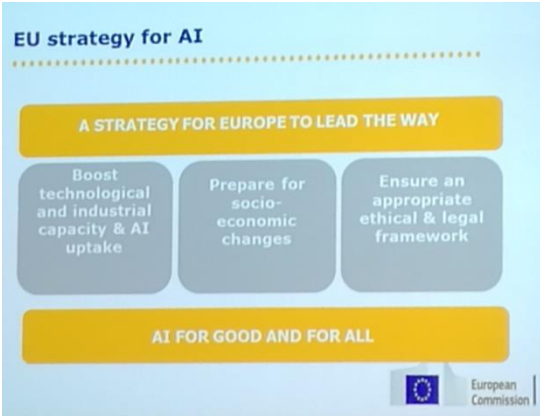


Figure 2: EU strategy involves three elements: science/technology, socio-economic changes and the social framework.

While the EU is as concerned with legal issues as it is with the research, China has significantly improved its AI efforts through governmental and private funding, and there is no major rift in the government. The democracies of several European countries and the USA are torn apart because of ideological and political antagonisms, instead of focusing on technical progress. For Chinese researchers, the road to success and obtaining a good position at home is to publish at major AI conferences, in major journals and join established researcher teams in the USA or Europe. For Europeans, it is possible to follow the Chinese path, but no European country offers a several-times-higher salary for researchers returning home, like China does. While the presidents of superpowers from the USA to Russia declare the tremendous importance of AI in relation to world dominance, the percentage of papers best demonstrates who supports the field the most. This is not to say that all major countries are not increasing their AI funds significantly. For example, the EU has presented its plans at IJCAI (Figure 3): first, a 70% increase, followed by a 100% and then another 100%. The US Department of Defense (DoD) established the Joint AI Center (JAIC). It will host the DoD's 600 AI projects with an estimated $1.7 billion over 6 years. As predicted, AI will likely change the nature of welfare, along with several other fields. However, without sufficient AI research, nobody can expect to maintain its leading position in the world.

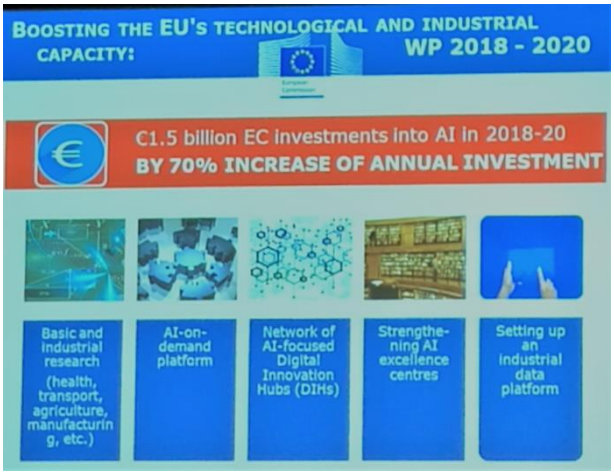


Figure 3: The EU will significantly increase AI funding. Finally. Will national governments follow?

A closer look at the EU plan reveals that there are several new ideas, as presented in Figure 4. Among others, the EU will fund the open AI platform, which is at least partially influenced by Elon Musk's, which by the way won the first 5 vs 5 Dota2 game with expert players (some small additional limitations). The EU plan was probably the major AI strategy presented at the conference. While China does it on its own and the USA allocates most funds to military applications, the EU is focused on a public, general, AI-boosting plan to benefit everybody. That is for sure great news, not only for AI in Europe, but for humanity as a whole.

Figure 4: The EU strategy introduces several integrating EU components, including the AI toolbox and the Network of Digital Innovation Hubs. Unfortunately, many of the most advanced AI hubs are in the UK.

Several new mechanisms like CLAIRE are already active (https://claire-ai.org/): "an initiative by the European AI community that seeks to strengthen European excellence in AI research and innovation."

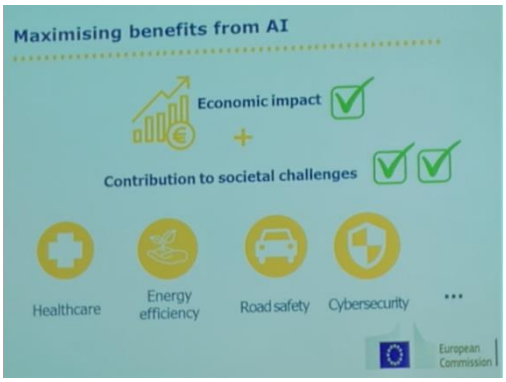


Figure 5: The EU expects that AI progress will bring several benefits, from the economic impacts to solutions to the societal challenges. There are several areas that will see major advances in the near future, such as healthcare.

"If Europe were to fall behind in AI technology, we would be likely to face challenging economic consequences, an academic brain drain, reduced transparency, and increasing dependency on foreign technologies, products and values. The CLAIRE initiative presents a proposal to avoid that."

"The CLAIRE initiative aims to establish a pan-European network of Centres of Excellence in AI, strategically located throughout Europe, and a new, central facility with a state-of-the-art, "Google-scale", CERN-like infrastructure – the CLAIRE Hub – that will promote new and existing talent and provide a focal point for the exchange and interaction of researchers at all stages of their careers, across all areas of AI. The CLAIRE Hub will not be an elitist AI institute with a permanent scientific staff, but an environment where Europe's brightest minds in AI meet and work for limited periods of time. This will increase the flow of knowledge among European researchers and back to their home institutions."

Maybe, we should also remember the times when science was not a business, when we researched not for the purpose of cash, but for reasons of fundamental curiosity, a desire to improve our knowledge. Some spirit of that kind is still observable at the conferences and was also demonstrated, for example, by the computer chess tournament. During the breaks many participants occasionally stopped by and observed the most interesting matches. Komodo won the World Computer Chess Championship 2018 after a play-off with GridGinkgo. In third place was Jonny, due to a win over Leela Chess Zero. The latter was observed with much interest, due to having implemented AlphaZero for the PC. It was not a match for the best programs, instead it played out

very differently – intuitively, lucidly and error-prone. Obviously, it lacked the power of the Google computers to validate its fancy ideas, often in the form of sacrifices. Figure 6 shows the Komodo team, who received the Shannon Trophy (and replica) from the chairman of the ICGA David Levy.

Figure 6: Komodo was again the computer chess champion on PCs. Leela Chess Zero, a PC version of the Alpha Zero, played lucidly, but had no chance against the hard logic of Komodo.

In Stockholm 2018 the social meetings of societies were boosting the exchange of information and cooperation, be it inside the EU or international societies. For example, the EurAI meeting (Figure 7), IJCAI AI societies meeting, IFIP meetings, etc.

The IJCAI report should first of all be about scientific achievements. In 2018 there was distinct, albeit rather expected, progress. Indeed, there were plenty of reasonably novel improvements, and indeed the major theme was a challenging one: How to grow a mind, a true AI - but that was it. Quite enough for many, but a bit too classical for others. Furthermore, the AI influence on our everyday life has already achieved much greater impact than generally anticipated in the public opinion: every day AI makes around 100 trillion decisions, meaning it is thoroughly embedded into our society. Coupled with other ICT achievements, human society long ago developed into an information society – an integration of humans and ICT systems, and an integration of human society and technology. This is one of the reasons why nobody understands what is actually going on – social scientists understand society, while engineers and technical scientists understand technological systems, and finally nobody understand the two embedded and integrated into one unity – kind of Borg stuff, just that the unifying essence is the web and ICT and AI services. Another analogy is related to computer chess – when humans play based on their own brains, the inferiority and inability to understand complex relations are evident. Only coupled with powerful computers and advanced AI programs can we hope to decipher the societal changes and trends, and propose good solutions.

With regards to the novel applications, Tambe's group stood out from many – their security AI designing daily schedules for airports, harbors and other relevant facilities is employed at several locations worldwide. It has even been given to 60 wildlife parks to cope better with poachers. That is one of the successful applications, accompanied by a huge mass of new research systems, e.g., a novel HW and SW embedded system connected to the patient's spine that enables a paralyzed patient to stand up.

New classes of applications are emerging, e.g., in visual tasks. DNNs can transform a human face into another, even create a new face never seen before. An animal, say a horse, can be camouflaged into zebra stripes and it can move freely around in a simulated video. Systems speak perfectly and listen better than humans; they can sharpen a picture or translate from voice online. Google search is using DNNs to capture the best answer to a question.

On the other hand, there are seemingly bizarre simple problems that researchers have a hard time dealing with. While it is generally accepted that DNNs outperform humans in visual classification tasks, it is still a big problem to transfer one ML system based on examples from a specific hospital and specific scanning devices to another. The technical differences are small, causing human experts no problem, but for the DNNs these small details significantly impair the quality. Until recently, that is. At the conference, several solutions related to transfer learning, general AI and also real AI were presented and discussed. Why should AI systems not learn like children, gathering knowledge and learning from there on with a small number of examples, even a single one?

Figure 7: Children learn in a very different way to AI. Why not copy them?

There is shallow, i.e., current AI, deep AI (also claimed as shallow AI), real AI, and fake AI. The last one refers to chatbots, i.e., virtual assistants, where human operators often jump in communications and leave users under the impression that it is AI on its own. The real AI was one of the major themes of the conference, which is quite a big difference from the previous conferences, where the primary goal was to complete the tasks better than expert humans, be it chess or detecting malignant tissue patterns. Now the task is different – perform at the level of children aged a couple of years. While supervised learning clearly achieves top performance, compared to humans it needs far too many examples, which is not acceptable, at least for the slow humans. Similarly, reinforcement learning needs way too many trials. Furthermore, machines do not have common sense compared even to young children.

In terms of the ban on autonomous weapons, more and more societies and countries are joining the ban. EurAI, as the union of all AI European societies and the second largest AI society in the world, also supports the ban.

Figure 7: More and more societies and countries are joining the ban on autonomous weapons. EurAI, as the union of all AI European societies, also joined the efforts.

In 2018, the debate on banning autonomous weapons was held in the UN and in the European Parliament: https://www.stopkillerrobots.org/2018/07/parliaments-2/. The list of institutions supporting the ban is here: https://www.stopkillerrobots.org/coalition/.

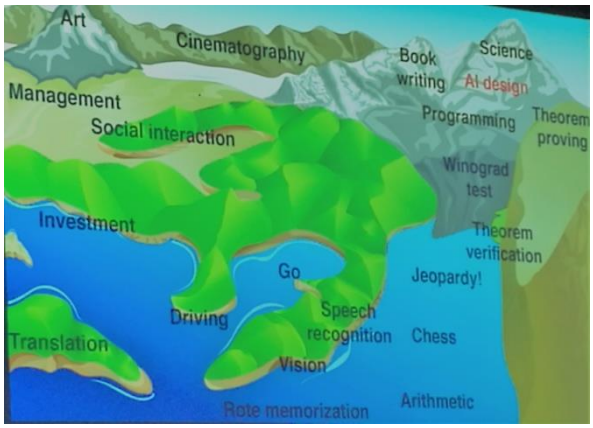


Figure 8: Tegmark's view of the AI field.

Another interesting approach is to generate general AI, as was the case in the 2018 IJCAI conference. Currently, the majority opinion among AI researchers is that general AI is possibly 10 years away. It is probably not that the AI community lacks computer power or finances; it is the novel ideas that we are striving for. There is also a reasonable consensus that AI can, could, should and will help humans solve major human societal problems. Scientists should avoid the politics, especially the discrepancies between different ideological or political tracks, and defer from attacking colleagues along these lines. Science should be kept as separate as possible from politics and ideology. With these words from Tegmark (Figure 8) we look into a bright EU AI, AI and human future.

References

[1] IJCAI 2018 conference (https://www.ijcai-18.org/).

Special issue on "The Eighth International Symposium on Information and Communication Technology – SoICT 2017"

Since 2010, the Symposium on Information and Communication Technology – SoICT has been organized annually. The symposium provides an academic forum for researchers to share their latest research findings and to identify future challenges in computer science. The best papers from SoICT 2015 and SoICT 2016 have been extended and published in the Special issue "SoICT 2015" and "SoICT 2016" of the Informatica Journal, Vol.40, No.2 (2016) and Vol. 41, No. 2 (2017). In 2017, SoICT was held in Nha Trang, Vietnam, during December 7–8. The symposium covered four major areas of research including Artificial Intelligence and Big Data, Information Networks and Communication Systems, Human-Computer Interaction, and Software Engineering and Applied Computing.

Among 132 submissions from 22 countries, 64 papers were accepted for presentation at SoICT 2017. Among them, the following six papers were carefully selected, after further extension and additional reviews, for inclusion in this special issue.

The first paper, *"Spectrum utilization efficiency of elastic optical networks utilizing coarse granular routing"* by Hai-Chau Le and Ngoc T. Dang investigated an elastic optical network that uses coarse granular routing based on coarse granular node architecture. The network takes advantages of both elastic optical networking and coarse granular routing technologies to cope with the trade-off between the link cost and the node cost in order to build a spectrum-and-cost efficient solution for future Internet backbone networks. The authors have evaluated the hardware scale requirement and the spectrum utilization efficiency of the network with typical modulation formats under various network and traffic conditions.

The second paper, "*Time-stamp incremental checkpointing and its application for an optimization of execution model to improve performance of CAPE*" by Van Long Tran, Eric Renault, Viet Ha Hai, and Xuan Huyen Do presents an improvement of Discontinuous Incremental Checkpointing, and a new execution model for CAPE using new techniques of checkpointing. It contributes to improve the performance and make CAPE even more flexible.

The third paper, "*SHIOT: A novel SDN-based framework for the heterogeneous Internet of Things"* by Hai-Anh Tran, Duc Tran, Linh-Giang Nguyen, Quoc-Trung Ha, Van Tong, and Abdelhamid Mellouk developed an SDN-based framework called SHIOT which relies on the ontology for examining the end-user requests and applies an SDN controller to classify flow scheduling over the task level.

The fourth paper, "*USL: A domain-specific language for precise specification of use cases and its transformations"* by Chu Thi Minh Hue, Dang Duc Hanh, Nguyen Ngoc Binh, and Le Minh Duc introduces a domain-specific language named the "Use case Specification Language (USL)" to precisely specify use cases. The authors define the abstract syntax of USL using a metamodel together with OCL wellformedness rules and then provide a graphical concrete syntax for the usability goal. This paper also defines precise semantics for USL by mapping USL models to Labelled Transition Systems (LTSs). It opens a possibility to transform USL models to software artifacts such as test cases and design models.

The fifth paper, *"Effective deep multi-source multi-task learning frameworks for smile detection, emotion recognition and gender classification"* by Dinh Viet Sang and Tran Bao Cuong proposes effective multi-task deep learning frameworks which can jointly learn representations for three tasks: smile detection, emotion recognition, and gender classification. The frameworks can be learned from multiple sources of data with different kinds of task-specific class labels.

The sixth paper, *"Alignment-free sequence searching over whole genomes using 3D random plot of query DNA sequences" by* Da-Young Lee, Hae-Sung Tak, Han-Ho Kim, and Hwan-Gue Cho proposes a new alignment-free sequence comparison and search method to overcome the limitations of the alignment-based model.

We hope that readers interested in Information and Communication Technology will find this Special Issue a useful collection of papers.

Huynh Thi Thanh Binh
Ichiro Ide

Spectrum Utilization Efficiency of Elastic Optical Networks Utilizing Coarse Granular Routing

Hai-Chau Le and Ngoc T. Dang
Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
Computer Communication Labs, The University of Aizu, Aizu-wakamatsu, Japan
E-mail: chaulh@ptit.edu.vn, ngocdt@ptit.edu.vn

Keywords: elastic optical network, optical cross-connect, spectrum selective switch, routing and spectrum assignment

Received: March 29, 2018

In this paper, we have investigated an elastic optical network that uses coarse granular routing based on our recently developed coarse granular node architecture. The developed coarse granular optical cross-connect (OXC) architecture that enables routing bandwidth-flexible lightpaths coarse-granularly is based on coarser granular selective spectrum switches. The network takes the advantages of both elastic optical networking and coarse granular routing technologies to cope with the trade-off between the link cost and the node cost in order to build a spectrum-and-cost efficient solution for future Internet backbone networks. We have evaluated the hardware scale requirement and the spectrum utilization efficiency of the network with typical modulation formats under various network and traffic conditions. We also compared the spectrum utilization of our network to that of corresponding traditional WDM network and conventional elastic optical network. Numerical results verified that, similar to conventional elastic optical network, the proposed network offers a substantial spectrum saving comparing to traditional WDM network.

Povzetek: Prispevek uvede izvirno elastično optično omrežje in analizira lastnosti kot učinkovitost.

1 Introduction

The ever-increasing Internet traffic growth has been continuously spurred by newly emerged high-performance and bandwidth-killer applications such as 4k/HD/ultra-HD video, e-Science and cloud/grid computing [1, 2]. To cope with the explosive traffic increment and to support further mobility, flexibility and bandwidth heterogeneity, the necessity of cost-efficient and bandwidth-abundant flexible optical transport networks has become more and more critical [3, 4]. To scale up to Terabit/s, current optical transport networks based on current WDM technology with a fixed ITU-T frequency grid will encounter serious issues due to the stranded bandwidth provisioning, inefficient spectral utilization, and high cost [3]. Recent research efforts on optical transmission and networking technologies that are oriented forward more efficient, flexible, and scalable optical network solutions [4] can be categorized into two different approaches that are: 1) improving the link resource utilization/flexibility and 2) minimizing the node system scale/cost.

The first approach which aims to enhance the spectrum utilization and the network flexibility is currently dominated by the development of elastic optical networking technology [5-12]. Elastic optical networks (EON) realize spectrum- and energy-efficient optical transport infrastructure by exploiting bitrate-adaptive spectrum resource allocation with flexible spectrum/frequency grid and distance-adaptive modulation [8, 9]. They are also capable of providing dynamic spectrum-effective and bandwidth-flexible end-to-end lightpath connections while offering Telcos (IT/communication service providers) the ability to scale their networks economically with the traffic growth and the heterogeneity of bandwidth requirement [10, 11]. However, EON is still facing challenges owing to the lack of architectures and technologies to efficiently support bursty traffic on flexible spectrum. It also requires more complicated switching systems and more sophisticated network planning and provisioning control schemes [12].

On the other hand, the second approach targets the development of cost-effective, scalable and large scale optical switching systems [13-18]. One of the most attractive direction is the use of coarse granular optical path (lightpath) switching [16-17] that can be realizable with optical/spectrum selective switching technologies [18]. Spectrum selective switches (SSSs) are available with multiple spectrum granularities which are defined as the number of switching spectrum bands. It is demonstrated that, with a common hardware technology (i.e. MEMS, PLC, LCoS, …), the hardware scale is increased dramatically as finer granular SSSs are applied. Coarser granular SSSs are simpler and more cost-effective but, their routing flexibility is limited more severely. Unfortunately, this routing limitation may seriously affect the network performance, especially in case of dynamic wavelength path provision. In other words, node hardware scale/cost reduction only can be attained at a cost of certain routing flexibility restriction. Hence, it is desirable to enhance the node routing

flexibility while still keeping the hardware reduction as large as possible.

Based on that, in order to exploit elastic optical networking and coarse granular switching for a realizing cost-efficient, spectrum effective and flexible optical transport network, we have recently developed a single-layer optical cross-connect architecture based on coarse granular switching spectrum selective switches. Elastic optical network that employs the developed OXC architecture is still capable of exploiting elastic optical networking technology while attaining a substantial hardware reduction. We have also evaluated the network spectrum utilization in various network scenarios such as single modulation format (BPSK, QPSK, 8QAM and 16QAM) and distance-adaptive scheme. Numerical evaluations verified that, like a conventional elastic optical network, the proposed network can obtain a significant spectrum saving comparing to the corresponding traditional WDM network. A preliminary version of this work with the proposal and limited basic numerical effectiveness evaluation of a bandwidth-flexible and coarse granular optical cross-connect architecture was presented at the SoICT conference [19].

2 Elastic optical network utilizing coarse granular routing

2.1 Developed coarse granular routing OXC architecture [19]

Most existing optical cross-connect systems are realized by optical selective switch technology which is one of the most popular and mature optical switching technologies. For constructing a high-port count OXC, multiple spectrum selective switches can be cascaded to create a higher port count SSS to overcome the limitation of commercially available SSS port count which is currently 20+ and unlikely will be substantially enhanced cost-effectively in the near future [4, 18]. Therefore, larger scale optical cross-connect system requires more and/or higher port count SSSs. Moreover, spectrum selective switches are still costly and complicated devices. SSS cost/complexity strongly relies on the number of switching spectrum bands per fiber (also called the spectrum granularity). Finer granular SSSs are more complicated as well as have greater hardware scale and consequently, become more expensive.

Based on that observation, in order to exploit elastic optical network technology while keeping the hardware scale reasonably small, we have recently developed a coarse granular routing elastic optical cross-connect architecture (denoted as GRE network) for realizing flexible bandwidth large scale optical transport networks [19]. Figure 1 shows the developed OXC system in which, instead of using fine granular SSSs in traditional bandwidth-variable OXC in elastic optical networks, coarse granular spectrum selective switches are implemented to build a cost-efficient high-port count OXC system. Unlike neither traditional WDM networks that divide the spectrum into individual channels with the

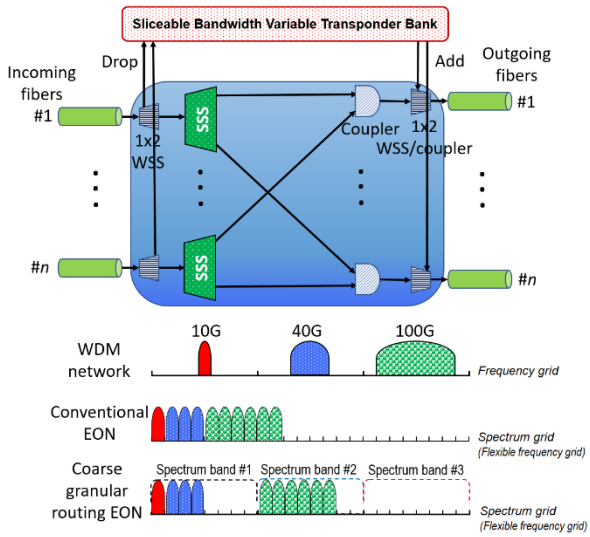


Figure 1: Coarse granular routing OXC architecture [19].

fixed channel spacing of either 50 GHz or 100 GHz specified by ITU-T standards nor elastic optical networks that employ a flexible frequency grid with a fine granularity (i.e. 12.5 GHz), the developed coarse granular routing elastic optical network employs the same flexible frequency grid but it routes lightpaths at the spectrum band level, so called "coarse" granular routing entity – GRE, through coarse granular OXCs; all spectrum slots of a band must be routed together as a single entity.

2.2 Hardware scale requirement

Practically, the cost and the control complexity of WSS/SSS-based systems depend strongly on the switch scale. Hence, switch scale minimization plays a key role for creating cost-effective large-scale WSS/SSS-based OXCs. Among recently available commercial optical switch technologies for constructing wavelength selective switch and/or spectrum selective switch systems, MEMS-based system are known as one of the most popular and widely adopted technology. Hence, to estimate the effectiveness of our recently developed OXC architecture, for simplicity, we just consider MEMS-based spectrum selective switches whose hardware scale mainly depends on the number of necessary elemental MEMS mirrors. Furthermore, without the loss of generality, adding/ dropping portions which can be simple 1x2 SSSs or couplers are also neglected. The switch scale of OXC systems, consequently, is quantified by the total MEMS mirrors required by SSS components.

We assume that the transmission bandwidth of a fiber is C_{fiber}, the channel spacing based on ITU-T frequency grid of traditional WDM network is G_{WDM} (G_{WDM}=100 GHz) and EON channel spacing is G_{EON} ($G_{EON} << G_{WDM}$). The number of wavelengths per fiber, W_{WDM}, in WDM network can be calculated as,

$$W_{WDM} = \frac{C_{fiber}}{G_{WDM}} \quad (1)$$

while the number of spectrum slots per fiber, S, of elastic optical network is given by,

$$S = \frac{C_{fiber}}{G_{EON}} \quad (2)$$

Let W denote the size of coarse granular routing entity (i.e. GRE granularity), the number of spectrum slots per GRE, and let S be the total number of spectrum slots that can be accommodated in a fiber; $1 \leq W \leq S$ and S is divisible by W and we have $L=S/W$ ($1 \leq L \leq S$) is the number of switching spectrum slots per fiber. Each mirror of a MEMS-based selective spectrum switch is dedicated to a spectrum slot (or spectrum band) and so, each SSS needs L MEMS mirrors. Note that all spectrum slots of a GRE are simultaneously switched by a mirror. Hence, total number of MEMS mirrors required in WDM OXC, elastic OXC and the proposed GRE architecture are calculated as following,

$$M_{WDM} = nW_{WDM}\left(1 + \left\lceil \frac{n-1}{M} \right\rceil\right) \quad (3)$$

$$M_{EON} = nS\left(1 + \left\lceil \frac{n-1}{M} \right\rceil\right) \quad (4)$$

$$M_{GRE} = nL\left(1 + \left\lceil \frac{n-1}{M} \right\rceil\right) \quad (5)$$

where n is the input/output fiber number (n>0), M is the maximal selective switch size (i.e. port count) and L is the GRE granularity. Table 1 summarizes the switch scale calculating formulas. The formulations also imply that the total number of necessary mirrors of an SSS is decreased as the applied GRE granularity becomes greater or it means that applying coarser granular SSSs (SSSs with greater W) will help to reduce the switch scale of OXC systems.

OXC Architecture	Switching component		Switch scale (Total mirror number)
	Switch element	*Total number*	
Conventional WDM	WSS	$n\left(1 + \left\lceil \frac{n-1}{M} \right\rceil\right)$	$nW_{WDM}\left(1 + \left\lceil \frac{n-1}{M} \right\rceil\right)$
Developed OXC	Coarser granular SSS		$n\frac{S}{W}\left(1 + \left\lceil \frac{n-1}{M} \right\rceil\right)$
Elastic OXC	SSS		$nS\left(1 + \left\lceil \frac{n-1}{M} \right\rceil\right)$

Table 1: Switch scale calculation.

Figure 2 shows the switch scale requirement of the developed OXC architecture, in terms of MEMS mirrors, with respect to both the number of input/output fibers (the port count) and the number of switching spectrum bands per fiber. The graph demonstrates that the hardware scale increases as the number of input fibers becomes greater. The hardware scale increment becomes much more significant if more number of switching bands per fiber (finer GRE granularity) is applied. Hence, a great deal of hardware scale/cost reduction can be achieved if the GRE granularity is limited at a reasonable value. It implies that coarse granular routing elastic optical network (using coarse granular SSSs) can be considered as a promising solution for creating cost-effective and bandwidth-abundant transport networks.

Moreover, Figure 3 shows the hardware scale comparison of the three comparative OXC architectures that are traditional OXC, elastic OXC and coarse

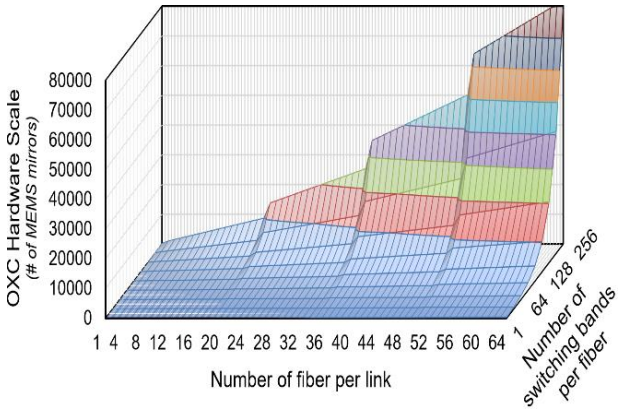


Figure 2: Hardware scale requirement of spectrum selective switch-based OXC.

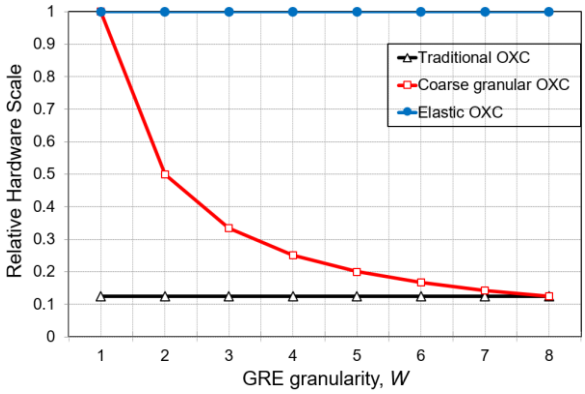


Figure 3: Hardware scale comparison.

granular OXC when the WDM channel spacing is 100 GHz and the spectrum slots of EON is 12.5 GHz. Due to the use of large channel spacing, i.e. 100 GHz or 50 GHz, traditional OXC needs the smallest hardware scale. On the other hand, thanks to the reduction of the number of switching spectrum bands, coarse granular OXC needs fewer number of switching elements comparing to conventional elastic optical cross-connect. Obviously, the hardware scale reduction offered by coarse granular OXC is enhanced, especially when coarser granular routing is applied (greater GRE granularity).

2.3 Network routing operation

Unlike conventional OXCs in WDM or EON networks, the developed GRE node suffers from a intra-node routing limitation due to the use of coarser granular spectrum selective. Figure 4 illustrates the routing principle of the developed coarse granular routing optical cross-connect architecture. In elastic optical network which uses the developed coarse granular routing node architecture (so called coarse granular routing elastic optical network), lightpaths of a spectrum band can be added/dropped flexibly and dynamically by 1x2 SSSs/optical coupler equipped for incoming and outgoing fibers and sliceable bandwidth variable transponders with the spectrum band capacity. Different from conventional elastic optical networks in which spectrum slots of each lightpath can be routed separately, in this network, whole spectrum slots of a spectrum band

from an incoming fiber must be switched together as one entity due to the coarse granular routing restriction of spectrum selective switches. It means that all lightpaths which are assigned to spectrum slots of the same spectrum band have to be routed to a common output fiber. This restriction imposed by the spectrum band granularity of SSSs limits the routing flexibility of the proposed OXC architecture. The node routing flexibility depends on the SSS spectrum granularity. In coarse granular routing elastic optical network, finer SSS granularity can be applied to improve the node routing flexibility, however, utilizing finer granular SSSs may cause a rapid increase in hardware-scale/cost. Hence, the SSS granularity must be carefully determined while considering the balance the node routing flexibility against the hardware scale/cost.

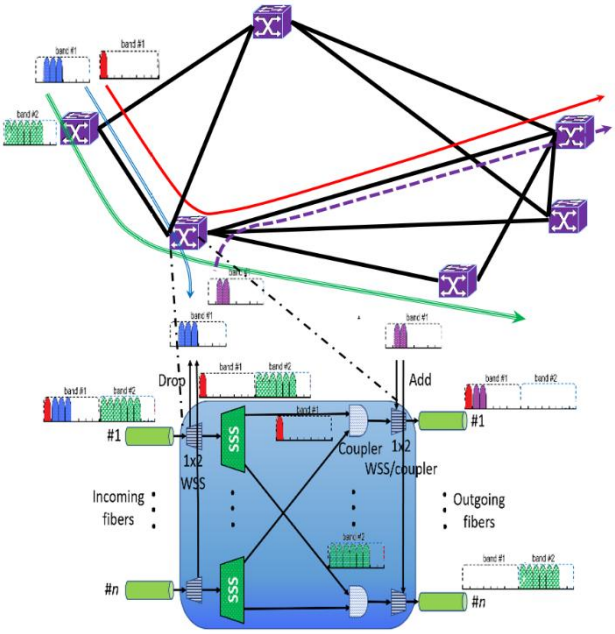


Figure 4: Coarse granular routing principle.

On the other hand, similar to conventional elastic optical networks, the coarse granular routing elastic optical network also can support single or multiple modulation formats flexibly and dynamically. Each lightpath can be assigned to a pre-determined modulation format (single modulation format scenario) or an appropriate modulation format according to its distance (called distance-adaptive scenario). In distance-adaptive scheme, for a given traffic capacity, modulating optical signal with a higher-order format offers higher capacity per spectrum slot and consequently, requires less number of spectrum slots. It means that applying higher-order modulation format obtains higher spectrum efficiency but its optical transparent reach is shortened and consequently, more frequent regeneration and/or more regeneration resources are required. Inversely, utilizing lower-order modulation formats might lower the spectrum slot capacity and hence, may cause an increment in the required spectrum slot number. Hence, impact of the modulation format assignment scenarios on the network spectrum utilization needs to be clarified.

3 Spectrum usage evaluation

3.1 Theoretical spectrum utilization analysis

In this section, we evaluate the spectrum utilization of three comparative optical networks including WDM, traditional EON and coarse granular routing elastic optical networks. Without the loss of generality, we assumed the following parameters. The channel spacing based on ITU-T frequency grid of traditional WDM network is 100 GHz (G_{WDM}=100 GHz, the most popular frequency grid) and the lowest order modulation format (i.e. BPSK) is applied. Elastic optical network is deployed with a typical channel spacing of 12.5 GHz (G_{EON}=12.5 GHz) and five modulation format assignment scenarios including four single modulation format (BPSK, QPSK, 8QAM and 16QAM) and a distance-adaptive schemes.

1) Point-to-point link

In this part, we simply estimated the spectrum utilization of a single point-to-point link with 3 comparative technologies including WDM, EON and our coarse granular routing EON (denoted as GRE). We assumed that the considered link includes Hs,d hops and has the total distance of $D_{s,d}$ where (s, d) is the source and destination node pair of the link, and requested bitrate of the connection on the link is $R_{s,d}$ (Gbps).

Based on that, let C_{WDM} be the channel capacity of BPSK WDM, the number of spectrum slots needed in the conventional WDM network, $NS_{WDM}(s, d)$, can be calculated as,

$$NS_{WDM}(s,d) = \left\lceil \frac{R_{s,d}}{C_{WDM}} \right\rceil H_{s,d}. \quad (6)$$

Hence, the total WDM spectrum is,

$$S_{WDM}(s,d) = G_{WDM} \left\lceil \frac{R_{sd}}{C_{WDM}} \right\rceil H_{s,d}. \quad (7)$$

For conventional elastic optical network, the spectrum slot number required in a single modulation format scheme (which uses only one modulation format of optical signals) is given by,

$$NS_{EON-MOD}(s,d) = \left\lceil \frac{R_{s,d}}{C_{EON-MOD}} \right\rceil H_{s,d} \quad (8)$$

where, *MOD* denotes the selected modulation format (it will be replaced by BPSK, QPSK, 8QAM or 16QAM) and $C_{EON\text{-}MOD}$ is the corresponding slot capacity. From Equation (8), the necessary spectrum of single modulation format elastic optical link can be evaluated as,

$$S_{EON-MOD}(s,d) = G_{EON} \left\lceil \frac{R_{s,d}}{C_{EON-MOD}} \right\rceil H_{s,d}. \quad (9)$$

Let α be the spectrum grooming ratio ($0 < \alpha \leq 1$); $\alpha = \frac{x}{GRE}$ where *GRE* is the GRE granularity, the capacity of coarse granular routing entity, and x is the average number of spectrum slots which carry the traffic in a coarse granular routing entity. Consequently, the number

of spectrum slots and the corresponding total spectrum required for coarse granular routing EON link are respectively calculated as,

$$NS_{GRE-MOD}(s,d) = \frac{1}{\alpha}\left\lceil \frac{R_{s,d}}{GRE \times C_{EON-MOD}} \right\rceil H_{s,d}, \quad (10)$$

and,

$$S_{GRE-MOD}(s,d) = \frac{GRE \times G_{EON}}{\alpha}\left\lceil \frac{R_{s,d}}{GRE \times C_{EON-MOD}} \right\rceil H_{s,d}. \quad (11)$$

On the other hand, for the distance-adaptive scheme of both conventional EON and our GRE networks, the modulation format of each lightpath is determined individually and assigned dynamically according to total distance of the lightpath. Therefore, if we assume that the simplest modulation format assignment strategy, which assigns the possible highest order of modulation format, is used, the total spectrum slot number required by the distance adaptive scheme of EON and coarse granular routing EON networks are,

$$NS_{EON-adap}(s,d) = \begin{cases} \left\lceil \frac{R_{s,d}}{C_{EON-16QAM}} \right\rceil H_{s,d} & if\ D_{s,d} \le L_{16QAM} \\ \left\lceil \frac{R_{s,d}}{C_{EON-8QAM}} \right\rceil H_{s,d} & if\ L_{16QAM} < D_{s,d} \le L_{8QAM} \\ \left\lceil \frac{R_{s,d}}{C_{EON-QPSK}} \right\rceil H_{s,d} & if\ L_{8QAM} < D_{s,d} \le L_{QPSK} \\ \left\lceil \frac{R_{s,d}}{C_{EON-BPSK}} \right\rceil H_{s,d} & otherwise, \end{cases} \quad (12)$$

and,

$$NS_{GRE-adap}(s,d) = \begin{cases} \frac{1}{\alpha}\left\lceil \frac{R_{s,d}}{GRE \times C_{EON-16QAM}} \right\rceil H_{s,d} & if\ D_{s,d} \le L_{16QAM} \\ \frac{1}{\alpha}\left\lceil \frac{R_{s,d}}{GRE \times C_{EON-8QAM}} \right\rceil H_{s,d} & if\ L_{16QAM} < D_{s,d} \le L_{8QAM} \\ \frac{1}{\alpha}\left\lceil \frac{R_{s,d}}{GRE \times C_{EON-QPSK}} \right\rceil H_{s,d} & if\ L_{8QAM} < D_{s,d} \le L_{QPSK} \\ \frac{1}{\alpha}\left\lceil \frac{R_{s,d}}{GRE \times C_{EON-BPSK}} \right\rceil H_{s,d} & otherwise. \end{cases} \quad (13)$$

From Equations (12) and (13), the required spectrum utilization of elastic optical link and that of coarse granular routing EON are estimated accordingly by,

$$S_{EON-adap}(s,d) = G_{EON} NS_{EON-adap}(s,d) \quad (14)$$

and,

$$S_{GRE-adap}(s,d) = GRE \times G_{EON} NS_{GRE-adap}(s,d). \quad (15)$$

2) Spectrum utilization of the network

Given a network topology $G=\{V, E\}$ in which V is the set of nodes, $|V|=n$, and E is set of links. For each node pair (s, d) ($(s,d) \in V\text{x}V$), we assume that the traffic load requested from the source node, s, to the destination node, d, is $R_{s,d}$, the hop count and the distance of the route connecting s and d are $H_{s,d}$ and $D_{s,d}$ respectively.

Based on the calculations given in Equations (7) and (9), total spectrum required in conventional WDM network is,

$$S_{WDM} = \sum_{\substack{(s,d)\in V\text{x}V \\ s\neq d}} G_{WDM}\left\lceil \frac{R_{s,d}}{C_{WDM}} \right\rceil H_{s,d}, \quad (16)$$

and the spectrum utilization of elastic optical networks for single modulation format scheme is given by,

$$S_{EON-MOD} = \sum_{\substack{(s,d)\in V\text{x}V \\ s\neq d}} G_{EON}\left\lceil \frac{R_{s,d}}{C_{EON-MOD}} \right\rceil H_{s,d}. \quad (17)$$

Similarly, from Equation (11), we have the total spectrum utilization of coarse granular routing elastic optical network for single modulation format scheme as following,

$$S_{GRE-MOD} = \sum_{\substack{(s,d)\in V\text{x}V \\ s\neq d}} \frac{GRE \times G_{EON}}{\alpha}\left\lceil \frac{R_{s,d}}{GRE \times C_{EON-MOD}} \right\rceil H_{s,d}. \quad (18)$$

Moreover, in distance-adaptive scheme, elastic optical networks including both conventional network and our developed network are able to assign modulation format dynamically. In fact, there are many different modulation assignment strategies, i.e. shortest path first (or least spectrum), least generating resource,… Depending on the applied strategy, the implementing portions of available modulation formats can be varied. If we assume that α, β, γ and δ are coefficients which determine the distribution of the selected modulation formats (BPSK, QPSK, 8QAM and 16QAM) in the network respectively, $\alpha\ge0$, $\beta\ge0$, $\gamma\ge0$, $\delta\ge0$ and $\alpha+\beta+\gamma+\delta=1$. Based on Equations (17) and (18), the required spectrum of distance-adaptive conventional elastic optical network and that of coarse granular routing EON network can be calculated as,

$$S_{EON-adap} = \alpha S_{EON-BPSK} + \beta S_{EON-QPSK} + \gamma S_{EON-8QAM} + \delta S_{EON-16QAM} \quad (19)$$

$$S_{GRE-adap} = \alpha S_{GRE-BPSK} + \beta S_{GRE-QPSK} + \gamma S_{GRE-8QAM} + \delta S_{GRE-16QAM} \quad (20)$$

This means that the performance of distance adaptive networks is in the middle comparing to other single modulation format elastic networks.

From Equations (16)-(20), the length of lightpaths, in term of both hop count and distance, significantly affects the usage of spectrum; longer paths are, more spectrum is required. It should be minimized to optimize the resource usage in elastic optical networks. In other words, the shortest paths should be used for lightpaths. However, note that implementing the shortest paths simply may result in a substantial spectrum collision.

3.2 Numerical results and discussion

To estimate the performance efficiency of the developed coarse granular routing elastic optical network, we utilized the following parameters for numerical evaluations. The frequency grid of WDM network is 100 GHz and the spectrum slot bandwidth of EON and GRE networks is 12.5 GHz. Two tested network topologies are pan-European optical transport network, COST266, and US backbone network, USNET (see Figure 5). Traffic load is represented by the total traffic demand requested between node pairs which is assigned randomly according to a uniform distribution in the range from 50 Gbps to 500 Gbps (for each traffic load, 100 samples were tested and the average values were then plotted).

In the numerical experiments, we have also assumed comparative elastic optical networks provide four typical

modulation formats that are BPSK, QPSK, 8QAM and 16QAM. Consequently, there are five experimental network scenarios that are four single modulation format schemes (BPSK, QPSK, 8QAM, and 16QAM) and a distance-adaptive scheme. The coarse granular switching group capacity, GRE (the number of spectrum slots per group), is set as a variable. In fact, we tested the GRE granularity of 2, 4, and 8 (in case GRE=1, GRE network is equivalent to conventional EON). The obtained results of the corresponding WDM network are used as a benchmark (its graph is always 1); all obtained results for EON and GRE networks are compared to that of the corresponding WDM network and the relative data will be demonstrated.

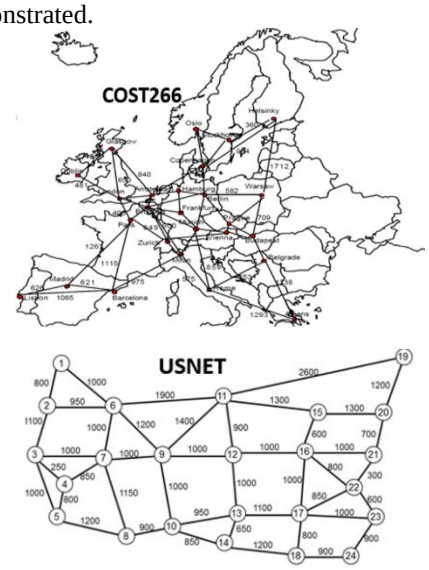
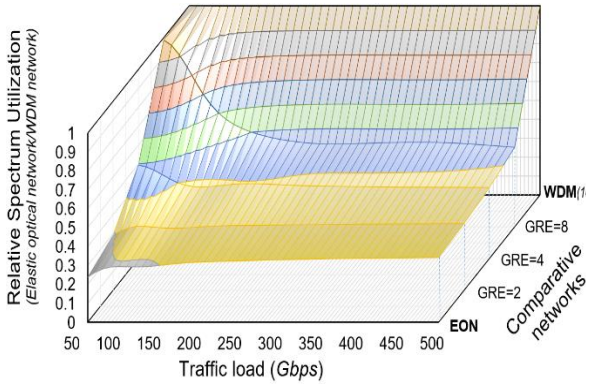
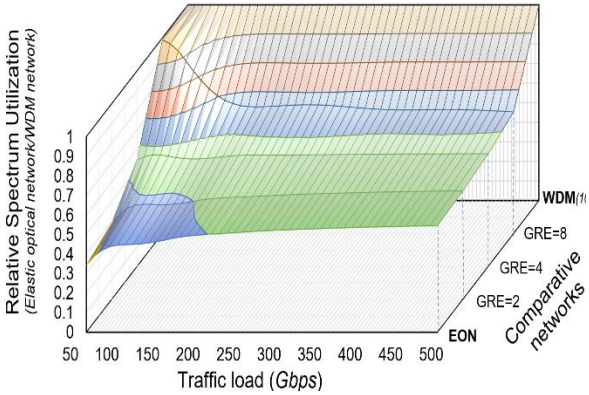


Figure 5: Tested network topologies.

Firstly, the spectrum usage comparison in the case of distance-adaptive scheme for the three comparative networks in COST266 and USNET topologies is illustrated in Figure 6. The obtained results show that both the GRE network and conventional elastic optical network offer a significant spectrum saving comparing to the traditional WDM network; up to 65% (45%) spectrum reduction can be achieved for COST266 (USNET) network topology with the traffic of 500 Gbps, thanks to the deployment of the flexible grid and dynamic modulation format assignment. It also demonstrates that the relative spectrum utilization of EON and GRE networks tends to decreased slightly as the traffic load becomes greater or finer granular routing is applied (smaller GRE granularity). That is because large traffic load can fill up huge channel spacing as used in conventional WDM networks and thus, using finer frequency grid does not help much to reduce the spectrum utilization. Note that, in this distance adaptive scheme, the spectrum utilization savings are less than those for 16QAM single modulation format scheme due to the possibility of implementing lower order modulation format to cope with the distance of required traffic without using any regenerating resource.

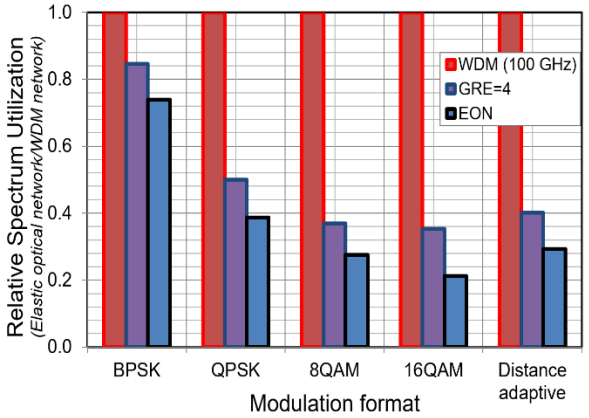


a) COST266

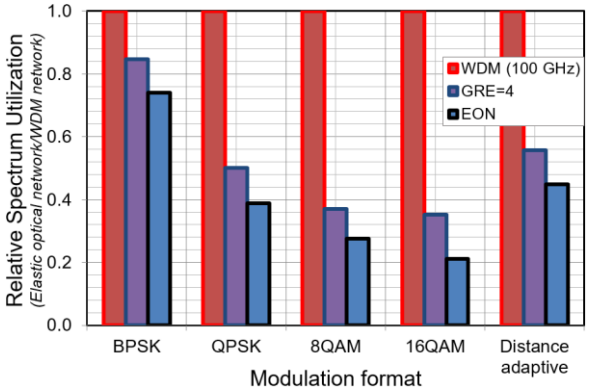


b) USNET

Figure 6: Spectrum utilization comparison for distance-adaptive scheme.



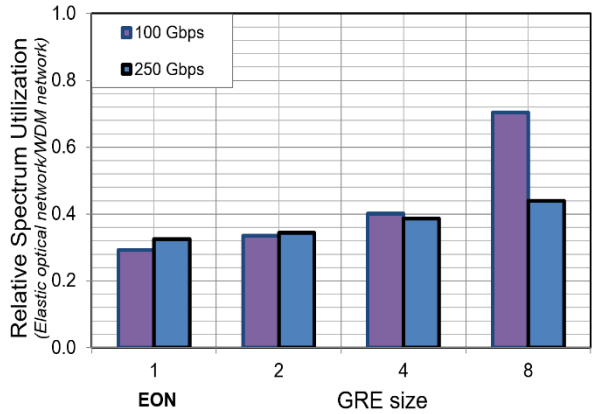
a) COST266



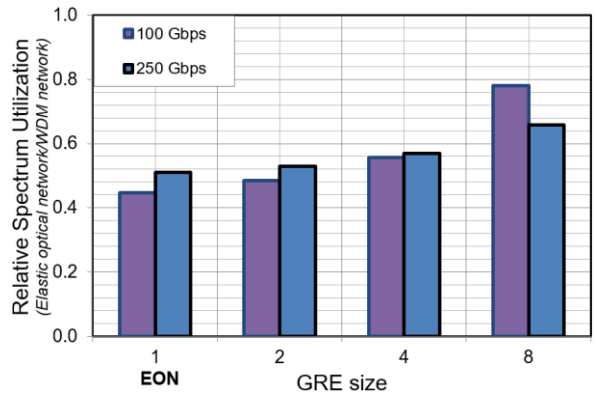
b) USNET

Figure 7: Impact of modulation formats.

Moreover, to verify the impact of the flexible modulation format assignment on the network performance, we compared five different network scenarios including four single modulation format schemes (BPSK, QPSK, 8QAM and 16QAM) and distance-adaptive scheme with the traffic load of 100 Gbps. The comparative results are described in Figure 7. It is demonstrated that employing higher order modulation formats offers better spectrum saving. Even the developed GRE network can reduce the hardware scale, the spectrum utilization of our network (as GRE=4) is more than that of EON due to the routing flexibility limitation. This also implies the importance of flexible modulation format assignment in saving spectrum while dealing the trade-off between the node routing flexibility (node cost) and the link resource requirement.



a) COST266



b) USNET

Figure 8: Dependence of the network spectrum usage on the GRE granularity.

Finally, Figure 8 shows the dependence of spectrum utilization on the GRE granularity applied when the traffic load is fixed at 100 Gbps and 250 Gbps. Again, it is shown that finer granular routing (smaller GRE granularity) offers better network performance, in terms of spectrum utilization, especially for small traffic load. The reason is that small traffic load may not fill up whole the spectrum band switched in the GRE network. Finer granular routing is expected to reduce the spectrum utilization, however, it may result in an explosive increase in the hardware scale. Hence, in the network point of view, it is desirable to balance the spectrum usage and the hardware scale requirements.

4 Conclusion

We have introduced a coarse granular routing elastic optical network that employs the developed coarse granular spectrum selective switch-based optical cross-connect architecture. By imposing coarse granular spectrum selective switching, the developed network is still able to exploit elastic optical networking technology while attaining a significant hardware reduction. To evaluate the performance of the coarse granular routing elastic optical network, we have clarified its spectrum utilization in various network scenarios, single modulation format (including BPSK, QPSK, 8QAM and 16QAM) and distance adaptive schemes, under different traffic conditions. We also compared the spectrum utilization of the network to that of corresponding traditional WDM network and conventional elastic optical network. Numerical results verified that, similar to conventional elastic optical network, the proposed network offers a substantial spectrum saving, says up to 65%, comparing to traditional WDM network. The developed network provides a promising solution to deal with the trade-off between node cost and link cost for creating cost-effective and spectrum-efficient future Internet backbone networks.

5 Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2015.39.

6 References

[1] Cisco Visual Networking Index: Forecast and Methodology, Cisco system, 2014–2019. Retrieved from http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[2] E. B. Desurvire (2006). Capacity demand and technology challenges for lightwave systems in the next two decades. *Journal of Lightwave Technology*, IEEE, vol. 24, No. 12, pp. 4697-4710. https://doi.org/10.1109/JLT.2006.885772

[3] J. Berthold, A. Saleh, L. Blair, J. Simmons (2008). Optical networking: Past, present, and future. *Journal of Lightwave Technology*, IEEE, vol. 26, No. 9, pp. 1104-1118. https://doi.org/10.1109/JLT.2008.923609

[4] K. Sato, H. Hasegawa (2009). Optical Networking Technologies That Will Create Future Bandwidth-Abundant Networks. *Journal of Optical Communications and Networking*, IEEE/OSA, vol. 1, no. 2, pp. A81-A93. https://doi.org/10.1364/JOCN.1.000A81

[5] A. Jukan and J. Mambretti (2012). Evolution of Optical Networking Toward Rich Digital Media Services. *Proceedings of the IEEE*, IEEE, vol. 100, no. 4, pp. 855-871.

https://doi.org/10.1109/JPROC.2011.2182076

[6] G. Bosco, V. Curri, A. Carena, P. Poggiolini, and F. Forghieri (2011). On the performance of Nyquist-WDM terabit superchannels based on PM-BPSK, PM-QPSK, PM-8QAM or PM-16QAM subcarriers. *Journal of Lightwave Technology*, IEEE, vol. 29, No.1, pp. 53–61. https://doi.org/10.1109/JLT.2010.2091254

[7] G. Zhang, M. De Leenheer, A. Morea and B. Mukherjee (2013). A Survey on OFDM-Based Elastic Core Optical Networking. *IEEE Communications Surveys & Tutorials*, IEEE, vol. 15, no. 1, pp. 65-87. https://doi.org/10.1109/SURV.2012.010912.00123

[8] M. Jinno, H. Takara, B. Kozicki, Y. Tsukishima, Y. Sone, and S. Matsuoka (2009). Spectrum-Efficient and Scalable Elastic Optical Path Network: Architecture, Benefits, and Enabling Technologies. *IEEE Communications Magazine*, IEEE, vol. 47, pp. 66-73. https://doi.org/10.1109/MCOM.2009.5307468

[9] O. Gerstel, M. Jinno, A. Lord and S. J. B. Yoo (2012). Elastic optical networking: a new dawn for the optical layer?. *IEEE Communications Magazine*, IEEE, vol. 50, no. 2, pp. s12-s20. https://doi.org/10.1109/MCOM.2012.6146481

[10] A. Lord, P. Wright and A. Mitra (2015). Core Networks in the Flexgrid Era. *Journal of Lightwave Technology*, IEEE, vol. 33, no. 5, pp.1126-1135. https://doi.org/10.1109/JLT.2015.2396685

[11] M. Jinno, B. Kozicki, H. Takara, A. Watanabe, Y. Sone, T. Tanaka and A. Hirano (2010). Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network. *IEEE Communications Magazine*, IEEE, vol. 48, no. 8, pp.138-145. https://doi.org/10.1109/MCOM.2010.5534599

[12] B. Chatterjee, N. Sarma and E. Oki (2015). Routing and Spectrum Allocation in Elastic Optical Networks: A Tutorial. *IEEE Communications Surveys & Tutorials*, IEEE, vol. PP, no. 99, pp. 1. https://doi.org/10.1109/COMST.2015.2431731

[13] T. Zami, D. Chiaroni (2012). Low contention and high resilience to partial failure for colorless and directionless OXC. *Proceedings of Photonics in Switching*, OSA, paper Fr-S25-O15. Retrieved from https://ieeexplore.ieee.org/document/6608250

[14] I. Kim, P. Palacharla, X. Wang, D. Bihon, M. D. Feuer, S. L. Woodward (2012). Performance of Colorless, Non-directional ROADMs with Modular Client-side Fiber Cross-connects. *Proceedings of Optical Fiber Communication Conference (OFC2012)*, OSA, paper NM3F.7. https://doi.org/10.1364/NFOEC.2012.NM3F.7

[15] Y. Li, L. Gao, G. Shen, L. Peng (2012). Impact of ROADM colorless, directionless and contentionless (CDC) features on optical network performance. *Journal of Optical Communication and Networking*, IEEE, vol. 4, No. 11, pp. B58-B67. https://doi.org/10.1364/JOCN.4.000B58

[16] H.-C. Le, H. Hasegawa, K. Sato (2014). Performance evaluation of large-scale multi-stage hetero-granular optical cross-connects. *Optics Express*, OSA, vol. 22, no. 3, pp. 3157-3168. https://doi.org/10.1364/OE.22.003157

[17] Y. Taniguchi, Y. Yamada, H. Hasegawa, and K. Sato (2012). A novel optical networking scheme utilizing coarse granular optical routing and fine granular add/drop. *Proceedings of OFC/NFOEC*, OSA, pp. JW2A.2. https://doi.org/10.1364/NFOEC.2012.JW2A.2

[18] R. Hirako, K. Ishii, H. Hasegawa, K. Sato, H. Takahashi, M. Okuno (2011). Development of Single PLC-Chip Waveband Selective Switch that Has Extra Ports for Grooming and Termination. *Proceedings of the 16th Opto-Electronics and Communications Conference*, IEEE, pp. 492-493. Retrieved from https://ieeexplore.ieee.org/document/6015223

[19] Hai-Chau Le, Thanh Long Mai, Ngoc T. Dang (2017). Spectrum Utilization of Coarse Granular Routing Elastic Optical Networks. *Proceedings of SoICT'17: Eighth International Symposium on Information and Communication Technology*, pp. 197-203. https://doi.org/10.1145/3155133.3155180

Time-stamp Incremental Checkpointing and its Application for an Optimization of Execution Model to Improve Performance of CAPE

Van Long Tran
Samovar, Télécom SudParis, CNRS, Université Paris-Saclay - 9 rue Charles Fourier, Évry, France
E-mail: van_long.tran@telecom-sudparis.eu and www.telecom-sudparis.eu
Hue Industrial College - 70 Nguyen Hue street, Hue city, Vietnam
E-mail: tvlong@hueic.edu.vn and www.hueic.edu.vn

Éric Renault
Samovar, Télécom SudParis, CNRS, Université Paris-Saclay - 9 rue Charles Fourier, Évry, France
E-mail: eric.renault@telecom-sudparis.eu and www.telecom-sudparis.eu

Viet Hai Ha
College of Education, Hue University - 34 Le Loi street, Hue city, Vietnam
E-mail: haviethai@gmail.com and www.dhsphue.edu.vn

Xuan Huyen Do
College of Sciences, Hue University - 77 Nguyen Hue street, Hue city, Vietnam
E-mail: doxuanhuyen@gmail.com and www.husc.edu.vn

Keywords: OpenMP, OpenMP on cluster, CAPE, Checkpointing-Aided Parallel Execution, Checkpointing, Incremental checkpointing, DICKPT, TICKPT

Received: March 29, 2018

CAPE, which stands for Checkpointing-Aided Parallel Execution, is a checkpoint-based approach to automatically translate and execute OpenMP programs on distributed-memory architectures. This approach demonstrates high-performance and complete compatibility with OpenMP on distributed-memory systems. In CAPE, checkpointing is one of the main factors acted on the performance of the system. This is shown over two versions of CAPE. The first version based on complete checkpoints is too slow as compared to the second version based on Discontinuous Incremental Checkpointing. This paper presents an improvement of Discontinuous Incremental Checkpointing, and a new execution model for CAPE using new techniques of checkpointing. It contributes to improve the performance and make CAPE even more flexible.

Povzetek: Predstavljena je izboljšava CAPE - paralelno izvajanje, usmerjeno s podporo redundance.

1 Introduction

In order to minimize programmers' difficulties when developing parallel applications, a parallel programming tool at a higher level should be as easy-to-use as possible. MPI [1], which stands for Message Passing Interface, and OpenMP [2] are two widely-used tools that meet this requirement. MPI is a tool for high-performance computing on distributed-memory environments, while OpenMP has been developed for shared-memory architectures. If MPI is quite difficult to use for non programmers, OpenMP is very easy to use, requesting the programmer to tag the pieces of code to be executed in parallel.

Some efforts have been made to port OpenMP on distributed-memory architectures. However, apart from our solution, no solution successfully met the two following requirements: 1) to be fully compliant with the OpenMP standard and 2) high performance. Most prominent approaches include the use of an SSI [3], SCASH [4], the use of the RC model [5], performing a source-to-source translation to a tool like MPI [6, 7] or Global Array [8], or Cluster OpenMP [9].

Among all these solutions, the use of a Single System Image (SSI) is the most straightforward approach. An SSI includes a Distributed Shared Memory (DSM) to provide an abstracted shared-memory view over a physical distributed-memory architecture. The main advantage of this approach is its ability to easily provide a fully-compliant version of OpenMP. Thanks to their shared-memory nature, OpenMP programs can easily be compiled and run as processes on different computers in an SSI. However, as the shared memory is accessed through the network, the synchronization between the memories involves an important overhead which makes this approach hardly scalable. Some experiments [3] have shown that the larger the number of threads, the lower the performance. As a result, in order to reduce the execution time overhead involved by the use of an SSI, other approaches have been proposed. For example, SCASH only maps the shared variables of the processes onto a shared-memory area at-

tached to each process, the other variables being stored in a private memory, and the RC model uses the relaxed consistency memory model. However, these approaches have difficulties to identify the shared variables automatically. As a result, no fully-compliant implementation of OpenMP based on these approaches has been released so far. Some other approaches aim at performing a source-to-source translation of the OpenMP code into a MPI code. This approach allows the generation of high-performance codes on distributed-memory architectures. However, not all OpenMP directives and constructs can be implemented. As yet another alternative, Cluster OpenMP, proposed by Intel, also requires the use of additional directives of its own (ie. not included in the OpenMP standard). Thus, this one cannot be considered as a fully-compliant implementation of the OpenMP standard either.

Concerning to bypass these limitations, we developed CAPE [10, 15] which stands for Checkpointing-Aided Parallel Execution. CAPE is a solution that provides a set of prototypes and frameworks to automatically translate OpenMP programs for distributed memory architectures and make them ready for execution. The main idea of this solution is using incremental checkpoint techniques (ICKPT) [11, 12] to distribute the parallel jobs and their data to other processes (the fork phase of OpenMP), and collect the results after the execution of the jobs from all processors (the join phase of OpenMP). ICKPT is also used to deal with the exchange of shared data automatically.

Although CAPE is still under development, it has shown its ability to provide a very efficient solution. For example, a comparison with MPI showed that CAPE is able to reach up to 90% of the MPI performance [13, 14]. This has to be balanced with the fact that CAPE for OpenMP requires the introduction of few `pragma` directives only in the sequential code, i.e. no complex code from the user point of view, while writing a MPI code might require the user to completely refactorise the code. Moreover, as compared to other OpenMP for distributed-memory solutions, CAPE is fully compatible with OpenMP [13, 15].

This paper presents an improvement of DICKPT – a checkpoint technique for CAPE, and a new execution model applied these new checkpoints, that improves the performance and the flexibility of CAPE. A part of these results were also presented and published at the SoICT's 2017 conference [16]. The paper is organized as follows: the next section describes CAPE mechanism, capabilities and restrictions in details. Section 3 presents a development of checkpointing that are used in CAPE. Then, Section 4 presents the design and the implementation of the new execution model based on the new checkpointing techniques. The analysis and evaluation of both new checkpointing and execution model are presented in Section 5. Section 4 shows the result of the experimental analysis. Finally, Section 5 concludes the paper and presents our future works.

2 CAPE principles

In order to execute an OpenMP program on distributed-memory systems, CAPE uses a set of templates to translate an OpenMP source code into a CAPE source code. Then, the generated CAPE source code is compiled using a traditional C/C++ compiler. At last, the binary code can be executed independently on any distributed-memory system supporting the CAPE framework. The different steps of the CAPE compilation process for C/C++ OpenMP programs is shown in the Figure 1.

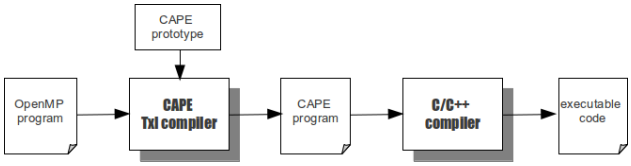


Figure 1: Translation of OpenMP programs with CAPE.

2.1 Execution model

The CAPE execution model is based on checkpoints that implement the OpenMP fork-join model. This mechanism is shown in Figure 2. To execute a CAPE code on a distributed-memory architecture, the program first runs on a set of nodes, each node being run as a process. Whenever the program meets a parallel section, the master node distributes the jobs among the slave processes using the Discontinuous Incremental Checkpoints (DICKPT) [12, 13] mechanism. Through this approach, the master node generates DICKPTs and sends them to the slave nodes, each slave node receives a single checkpoint. After sending checkpoints, the master node waits for the results to be returned from the slaves. The next step is different depending upon the nature of the node: the slave nodes receive their checkpoint, inject it into their memory, execute their part of the job, and sent back the result to the master node by using DICKPT; the master node waits for the results and after receiving them all, merges them before injection into its memory. At the end of the parallel region, the master sends the resulting checkpoint to every slaves to synchronize the memory space of the whole program.

2.2 Translation from OpenMP to CAPE

In the CAPE framework, a set of functions has been defined and implemented to perform the tasks devoted to DICKPT, typically, distributing checkpoints, sending/receiving checkpoints, extracting/injecting a checkpoint from/to the program's memory, etc. Besides, a set of templates has been defined in the CAPE compiler to perform the translation of the OpenMP program into the CAPE program automatically and make it executable in the CAPE framework. So far, nested loops and shared-data variable constructs are not supported yet. However, this is not regarded as an issue as this can be solved at the level

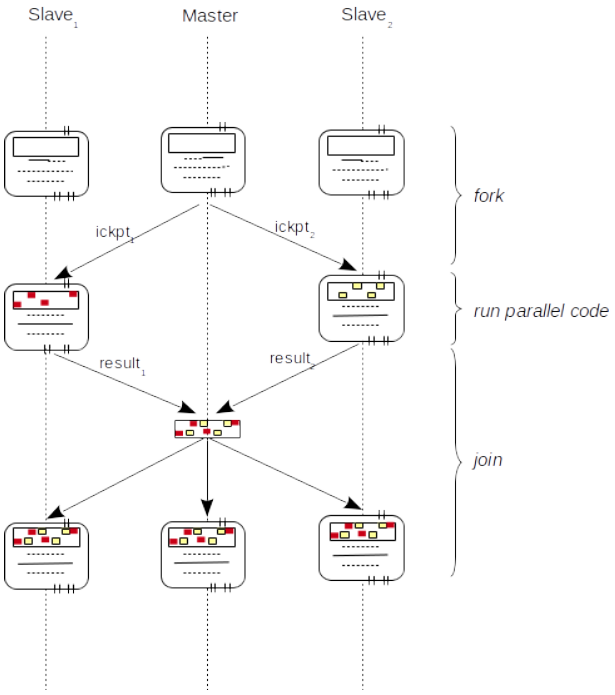


Figure 2: CAPE execution model.

of the source-to-source translation and does not require any modifications in the CAPE philosophy. In this end, CAPE can only be applied to OpenMP programs matching the Bernstein's conditions [17].

After the translations operated by the CAPE compiler, the OpenMP source code is free of any OpenMP directives and structures. Figure 3 presents an example of code substitution for the specific case of the `parallel for` construct. This example is typical of those we implemented for the other constructs [7]. The automatically generated code is based on the following functions that are part of the CAPE framework:

- `start( )` sets up the environment for the generation of DICKPTs.
- `stop( )` restores the environment used for the generation of DICKPT.
- `create(file)` generates a checkpoint with name `file`.
- `inject(file)` injects a checkpoint into the memory of the current process.
- `send(file, node)` sends a checkpoint from the current process to another.
- `wait_for(file)` waits for checkpoints and merges them to create another one.
- `merge(file1,file2)` merges two checkpoints together.

```
# pragma omp parallel for
  for ( A ; B ; C )
      D ;
```

↓ automatically translated into ↓

```
if ( master ( ) )
   start ( )
   for ( A ; B ; C )
        create ( before )
        send ( before, slave_x )
   create ( final )
   stop ( )
   wait for ( after )
   inject ( after )
   if ( ! last parallel ( ) )
           merge ( final, after )
           broadcast ( final )
else
   receive ( before )
   inject ( before )
   start ( )
   D
   create ( after_i )
   stop ( )
   send ( after_i, master )
   if ( ! last parallel ( ) )
           receive ( final )
           inject ( final )
   else
           exit
```

Figure 3: Template for the `parallel for` with incremental checkpoints.

- `broadcast(file)` sends a checkpoint to all slave nodes.
- `receive(file)` waits for and receives a checkpoint.

2.3 Discontinuous incremental checkpointing on CAPE

Checkpointing is the technique that saves the images of a process at a point during its lifetime, and allows it to be resumed at the saving's time if necessary [11, 18]. Using checkpointing, processes can resume their execution from a checkpoint state when a failure occurs. So, no need to take time to initialize and execute it from the begin. These techniques are introduced since two decades ago. Nowadays, they are researched and used widely on fault-tolerance, applications trace/debugging, roll-back/animated playback, and process migration.

Basically, checkpointing techniques can be categorized

into two groups: completed checkpoints and incremental checkpoints. Completed checkpointing [18, 19, 20] saves all information regarding the process at the points that it generate checkpoints. The advantages of this technique is reducing the time of generation and restoration. However, the checkpoint's size is too large. Incremental checkpointing [11, 21, 22, 23, 12, 24] only saves the modified information as compared to the previous checkpoint. This technique reaches advantages of reducing checkpoint's overhead and checkpoint's size, so it is in widely used in distributed computing. Besides, using data compression to reduce checkpoint's size [11, 21, 24], it is also focus on the techniques that detect modified data but reach the minimum of size. Some typical techniques are using page-based protection to identify the pages in memory that have been modified [11, 22, 23], using word-level granularity [21, 12], using block encoding [22], using user-directed and memory exclusion [11], using live variable analysis [24].

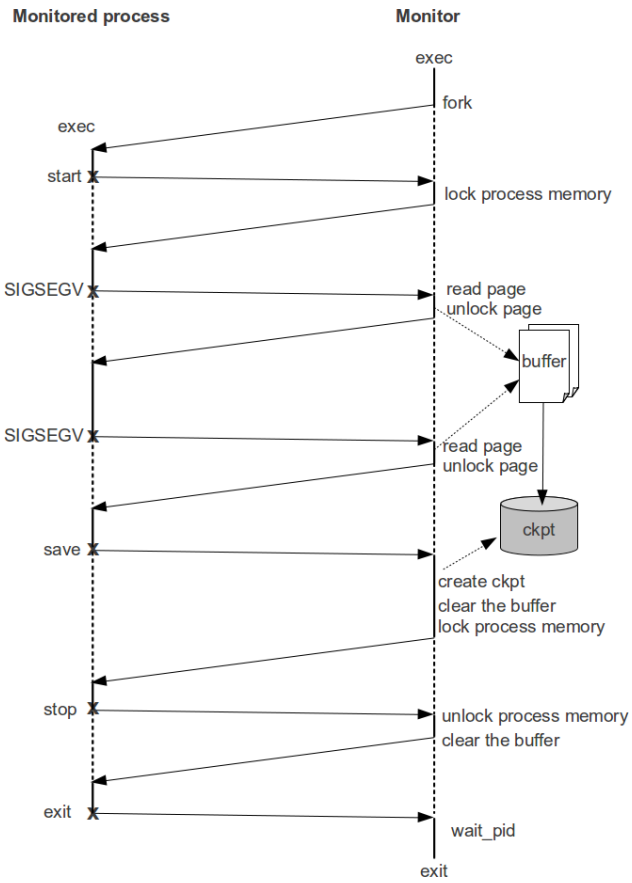


Figure 4: Principle of DICKPT in cases of checkpointing.

In CAPE, Discontinuous Incremental Checkpointing (DICKPT) is a development based on incremental checkpointing, that contains two kinds of data, register information and modified data of the process. In which, the first one is copied from all register data of the process, and the second one is identified based on write-protection techniques.

Figure 4 shows the steps to monitor and generate a checkpoint of a process on CAPE. It is done by an other process making use of the $ptrace$ Unix system call. The idea of these steps is that, at the beginning of the parallel region, the monitor sets all page of monitor process at write-protected. Whenever the monitored process wants to write into any write-protected page, a $SIGSEGV$ signal is generated. Then, the monitor saves the data of this page, removes the write-protection and lets the monitored process write into the page. At the end of the region, the monitor compares the saved data with the current data of monitored process page. The difference are extracted and saved into checkpoint file.

2.4 Remarks

The good performance of CAPE as compared to those of MPI and the full compliance to the OpenMP specifications [13, 15, 14] have made CAPE a good alternative to port OpenMP on distributed-memory architectures. So far, the implementation of CAPE is not complete, some disadvantages can be listed:

1. DICKPT saves all modified data of process, including temporary and private variables. This is an unnecessary synchronization inside an OpenMP program.

2. As shown in Figure 2, the master node might act as a bottleneck while waiting for checkpoints from the salves, merging checkpoints and/or sending back data to slaves for memory synchronization.

3. To distribute jobs to slaves, the master node generates a number of checkpoints that depends upon the number of slave nodes and so that each slave node receives a checkpoint (see Figure 7). This method can reach a high-level of optimization. However, it might not be enough flexible for some cases like 1) the number of slaves may not be identified at compile time, 2) the OpenMP source code should be modified to detect when the master generates the checkpoint and 3) the dynamic scheduling of OpenMP cannot be implemented using this method.

4. After distributing the jobs, the slave nodes execute the divided jobs while the master does nothing until the reception of the resulting checkpoints from the slaves, which clearly wastes resources.

5. For synchronization, the checkpoints should be sent by order in order to resume exactly the last state of process.

3 Time-stamp incremental checkpointing (TICKPT)

Time-stamp Incremental Checkpointing (TICKPT) is an improvement of DICKPT by adding new factor – time-stamp – into incremental checkpoints and by removing un-

necessary data based on data-sharing variable attributes of OpenMP program.

Basically, TICKPT contains three mandatory elements including register's information, modified region in memory of the process, and their time-stamp. As well as DICKPT, in TICKPT, the register's information are extracted from all registers of the process in the system. However, the time-stamp is added to identify the order of the checkpoints in the program. This contributes to reduce the time for merging checkpoints and selecting the right element if located at the same place in memory. In addition, only the modified data of shared variables are detected and saved into checkpoints. It makes checkpoint's size significantly reduced depending on the size of private variables of the OpenMP program.

To present the order of checkpoints in a program, time-stamps have to represent the order of the instructions when it is executed. For the general case, an activation tree [25] can be used to identify the sequence of function call in a program. For CAPE, checkpoints are always generated in same level of functions, so that the program counter can be used to ensure simplicity. However, if the instruction is a `loop`, the program counter is combined with the loop iteration to represent the order of the loop exactly.

To detect modified data, the write-protection mechanism is used. However, only the shared variables are written down in the checkpoint file. The matter in here is how to detect private and shared variables.

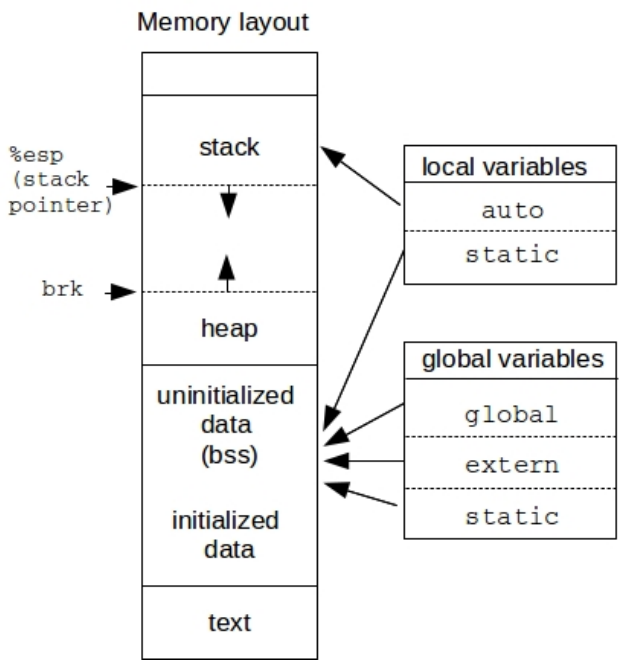


Figure 5: Allocation of OpenMP program's variables in virtual process memory.

In an OpenMP program, data-sharing variable attributes can be set up either, implicitly or explicitly [2]. All variables declared outside an `#pragma omp parallel` directive are implicitly shared. This includes all global and `static` local variables allocated in `heap` and `data` segments of the process's memory, and local variables allocated on the `stack` (see Figure 5). The variables in `heap` and `data` segments can easily be identified by their address. For the variables on the `stack`, we save the stack pointer before entering the `#pragma omp parallel` region. Variables declared before the stack pointer are shared. The others, are private.

To explicitly, change the status of a variable, the programmer can use data-sharing attributes like OpenMP directive `#pragma omp threadprivate (list of variables)` and relative clauses. The OpenMP data-sharing clauses are shown in Table 1.

Clauses	Description
default(none|shared)	Specifying the default behavior of variables
shared(list)	Specifying the list of shared variables
private(list)	Specifying the list of private variables
firstprivate(list)	Allowing to access value of the list of private variables in the first time
lastprivate(list)	Allowing to share value of the list of private variables at the end of parallel region
copyin(list)	Allowing to access value of threadprivate variables
copyprivate(list)	Specifying the list of private variables that should be shared among all threads.
reduction(list, ops)	Specifying the list of variables that are subject to a reduction operation at the end of the parallel region.

Table 1: OpenMP data-sharing clauses.

4 A new execution model for CAPE

In order to improve the performance of CAPE and its flexibility, we designed a new execution model that extends the one presented in Section 2.1. In this new execution model, DICKPT is replaced by TICKPT. Figure 6 illustrates the model which can be described as follows:

1. At the beginning of the program, all nodes in the system execute the same sequential code.

2. When a parallel region is reached, the master process creates a set of incremental checkpoints. The number of incremental checkpoints depends upon the number of tasks in the parallel region. Each incremental checkpoint contains the state of the program to be

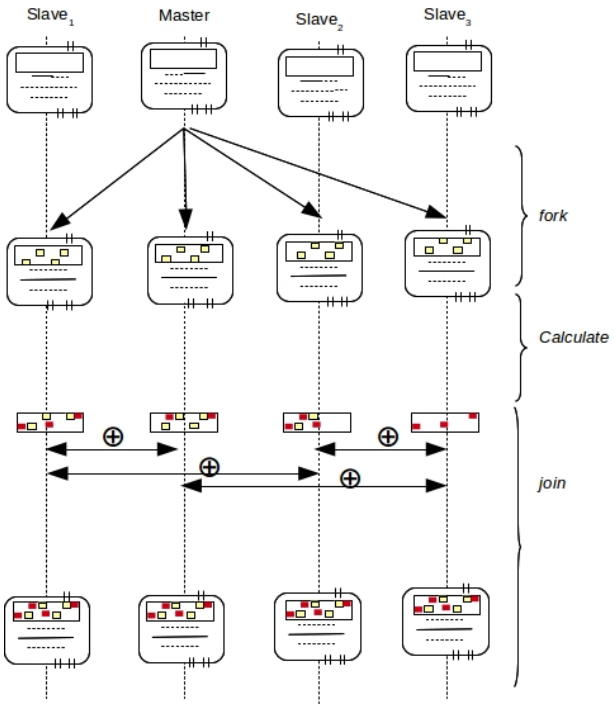


Figure 6: The new execution model for CAPE.

used to resume its execution in another process at the saved time.

3. The master process scatters the set of incremental checkpoints. Each node receives some of the checkpoints generated by the master process. This step is illustrated in the Figure 8.

4. The received incremental checkpoints are injected into the slave processes' memories.

5. The slave processes resume their execution.

6. Results on slave processes are extracted by identifying the modified regions and saved as an incremental checkpoint.

7. Incremental checkpoints of each process is sent back to the master node. Incremental checkpoints are combined altogether to generate a single checkpoint. This step can be distributed among the processes if need be.

8. The final combined incremental checkpoint is injected in the master process' memory and the master process can resume its execution.

Changing the execution model implies changing the translation templates. Figure 9 presents the template for the `#pragma omp parallel for` directive that adapts to the new execution model. The other OpenMP directives can be designed in a similar way. For this template, CAPE operates as follows:

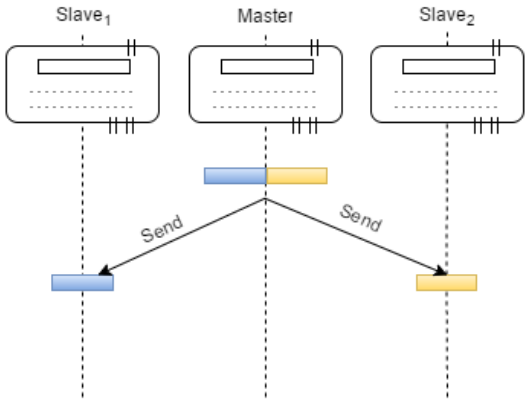


Figure 7: Scheduling method in CAPE-2.

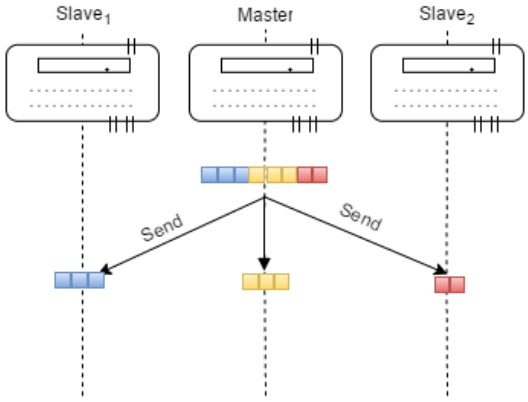


Figure 8: Scheduling method with the new execution model.

– $generate\_dickpt(before_i)$ (line 3): at each loop iteration, the master process generates an incremental checkpoint.

– $scatter(before, \&recv_n, master)$ (line 4): the master process scatters the checkpoints to the available processes, including itself. Each process receives some of the checkpoints ($recv_n$).

– $inject(recv_n)$ (line 5): each checkpoint is injected into the target process' memory.

– the execution is resumed on instruction D (line 6).

– $generate\_dickpt(after_n)$ (line 7): each process generates an incremental checkpoint that saves the result of its execution.

– $allreduce(after_n, \&after, [< ops >])$ (line 8): the $after_n$ checkpoint of process n is sent to the other processes. Checkpoints are combined, calculated and saved in a new $after$ checkpoint. With TICKPT, the order of checkpoints is presented in each of them, so this is performed using the Recursive Doubling algorithm [26] as illustrated in Figure 10.

– $inject(after)$ (line 9): incremental checkpoint $after$

```
# pragma omp parallel for
  for ( A ; B ; C )
      D ;
```

↓ automatically translated into ↓

```
1 if ( master ( ) )
2   for (A ; B ; C)
3     generate_dickpt (before_i)
4 scatter(before, &recv_n ,master)
5 inject(recv_n )
6 D
7 generate_dickpt (after_n)
8 allreduce(after_n,&after,[<ops>])
9 inject(after)
```

Figure 9: Prototype for the `parallel for` with the new execution model.

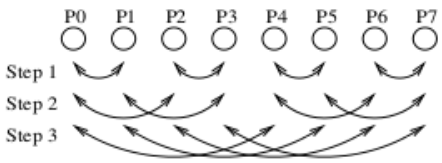


Figure 10: Recursive doubling for allreduce.

is injected into the application's memory to synchronize the state of the program on all nodes.

5 Analysis and evaluation

5.1 From DICKPT to TICKPT

As presented in Section 3, TICKPT is an evolution of DICKPT. It creates and adds time-stamps into checkpoints to make them more flexible and to reduce synchronization time when applying on CAPE. In addition, it removes unnecessary data to reduce checkpoint's size. For the synchronization time, we will analyse and evaluate the whole performance of CAPE. For checkpoint's size, we consider the amount of the modified data generated by TICKPT and DICKPT after having executed the piece of code in Figure 11 in each node, with various values for N.

Data contained in A, B and C variables are changed. The DICKPT counts them all, while TICKPT only counts data in variable C. Therefore, the amount of modified data significantly reduced with TICKPT as shown in Figure 12.

5.2 Analysis of the new execution model

Moving from a scheduling of CAPE processes based on the number of nodes (Figure 7) to a scheduling based on

```
#define N 1000
...
int A[N], B[N], C[N], i;
...
#pragma omp parallel for
                private(A,B) shared(C)
for(i = 0; i <N; i++){
       A[i] = i;
       B[i] = N - i;
       C = A[i] + B[i] ;
}
```

Figure 11: A piece of OpenMP code used to consider the amount of modified data with the two checkpoint techniques.

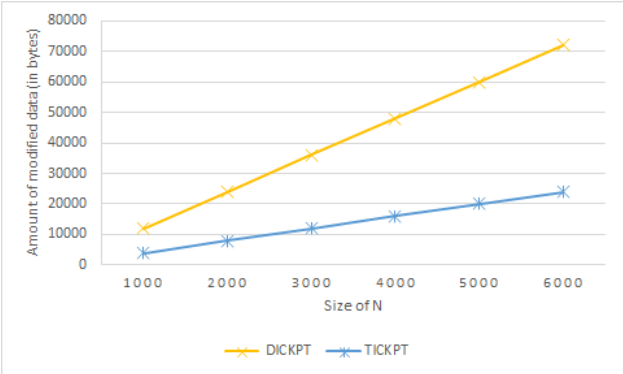


Figure 12: Amount of modified data (in bytes) generated by both methods.

the number of jobs (Figure 8) makes CAPE more flexible at least for the three following reasons:

1. The number of available processes can be identified at runtime. The master node can distribute the jobs to all available processes.

2. All OpenMP scheduling mechanisms such as `static` and `dynamic` can be implemented on CAPE. This is because the master node generates a number of checkpoints depending on the number of jobs. First step, one checkpoint can be sent to each slave node to execute a divided job. When the slave node finishes, it is sent the next checkpoint that has not been executed yet.

3. There is no need to modify the OpenMP source code to detect the location of the master process that generated the incremental checkpoints and sent them to the slave nodes.

For performance analysis and evaluation, considering that both initial and sequential codes of the program are executed in the same way in any processes of the system, only the execution time of the parallel regions has been considered.

Let t_f be the execution time of the fork phase, t_c be the computation time to execute the divided jobs and t_j be the

time for the join phase, ie. the time to synchronize data after executing the divided jobs at all nodes. For each parallel region, the execution time can be computed using equation 1.

$$t = t_f + t_c + t_j \tag{1}$$

Note that t_f is similar for both methods as both consider the work-shared steps and the generation of incremental checkpoints, and an incremental checkpoint only consist of very few bytes, ie. the time for the fork phase is close to zero.

In the previous execution model, the master process was not involved in the computation phase, and resources ware wasted. Let n be the number of jobs and p be the number of processes. Assume that each process takes one unit of time to execute a job, and the number of jobs is equally divided equally among the processes. The value for t_c becomes:

$$t_c = \left\lceil \frac{n}{p-1} \right\rceil \tag{2}$$

With the new execution model, all processes are involved in the computation phase so that t_c is equal to:

$$t_c = \left\lceil \frac{n}{p} \right\rceil \tag{3}$$

t_j is also impacted by the new execution model. In the previous model, the value for t_j is equal to the time for the slave processes to send their results to the master node for combination plus the time to receive the final checkpoint and inject it into the process' memory. This work is done sequentially. Thus, the time to send or receive a checkpoint is given by:

$$t_j = 2(p-1) \tag{4}$$

With the new execution model, the Recursive Doubling algorithm [26] is applied to communicate between all processes, so that t_j becomes:

$$t_j = \lceil log_2(p) \rceil \tag{5}$$

Computation time t_c is the most important factor that affects the execution time of a parallel region. From equations (2) and (3), it is easy to demonstrate that t_c for the previous execution model is always larger than t_c for the new execution model, ie. the execution time for CAPE is reduced with this new execution model. And the resources are used more efficiently.

Besides, the use of the Recursive Doubling algorithm during the join phase with the new execution model allows saving time when synchronizing data between processes. This is highlighted by comparing equations (4) and (5) with the previous execution model and the new execution model respectively.

6 Experiments

In order to measure the impact of the new execution model on the performance, as mathematically analyzed in Section 5, some experiments were conducted. These experiments were performed on 4-node and 16-node clusters. Each node includes an Intel core i3-2100, a dual-core 4-thread CPU running at 3.10 GHz and 2 GB of RAM. These computers are connected using a 100 Mb/s Ethernet network. To avoid external influence as much as possible, the entire system was dedicated to the tests during all of the performance of the measurement campaign.

The program used as the basis for these experiments is the classic matrix-matrix multiplication. The sizes of the matrices are increased from 1600×1600, 3200×3200 to 9600×9600. Each program is executed at least 10 times to measure the total execution times and a confidence interval of at least 98% has been always achieved for the measures. Data reported here are the means of the 10 measures.

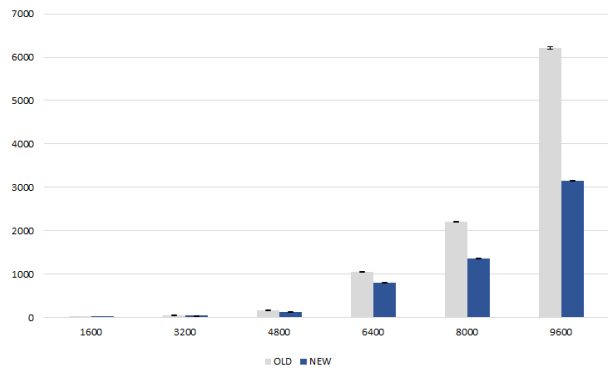


Figure 13: Total execution time (in seconds) for both models on 4-node clusters.

Figure 13 and 14 present the total execution time of CAPE on 4-node and 16-node clusters respectively. As can be seen from these figures, the execution time of both models are shown, the gray color (OLD) represents the previous execution model, and the blue one (NEW) represents the new execution model. The horizontal axis shows the size of the matrix and the vertical axis shows the execution time in seconds.

For the 4-node cluster, as compared with the previous model, the execution time of the new model is reduced significantly and the reduction is inversely proportional to the size of the matrix. The larger the size of the matrix, the shorter in time. This is due to the fact that there are only three nodes executing the divided jobs on the previous execution model. The master just divides and distributes jobs to slaves, and then waits for the results return. It does not participate in the computational part. In contrast, with the new execution model, master node receives and executes a part of the divided jobs. Therefore the computation time (t_c) on this model is much lower than on previous model, especially on the cluster with only 4 nodes.

For the 16-node cluster, the result in Figure 14 shows the

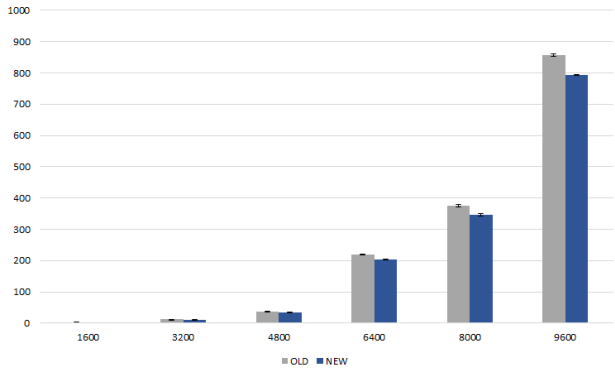


Figure 14: Total execution time (in seconds) of two models on 16-node clusters.

same trend, ie. the new execution model is better than the previous one. However, the distance between the execution time of both models is closer. It maintains a saving time around 10%. This is due to the larger number of nodes that leads to less time to compute the divided jobs. Therefore, the total saving time of t_c in this case is smaller.

Figure 15 presents the execution time of the fork (t_f), computation (t_c) and join (t_j) phases for both previous and new execution models on the master node with the 16-node cluster. The matrix size 9600 $\times$ 9600 is selected in this case.

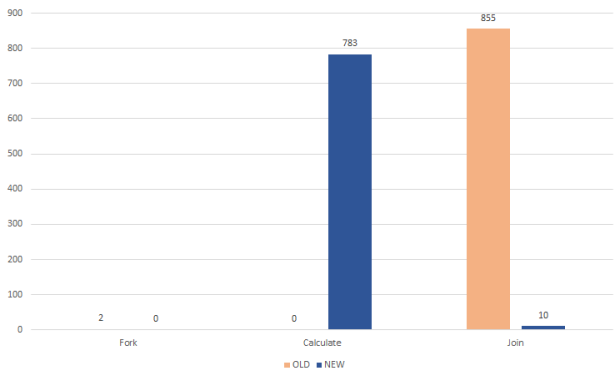


Figure 15: Execution time (in seconds) fork, computation and join phases for both previous and new execution models on the master node.

In the previous model, after the fork operation, the master node waits for the results from the slave nodes. Therefore, the value for t_c is equal to zero. For the same phase using the new execution model, the master node participates in the execution together with the slave nodes, so that t_c is much larger than zero. However, the new execution model uses the free resources of the master node to compute a part of the jobs of the whole program. This does not increase the whole execution time, but also contributes to improve the global efficiency of the system. The join phase comes right after the computation phase in the previous execution model. At this time, the master node waits for the results from the slave nodes and the synchronization of data. With the new execution model, this time is dedicated to the synchronization of data. Therefore, t_j is much smaller for the new execution model as compared with the previous execution one.

Indeed, both the theoretical analysis and the practical experiments on clusters composed of 4 nodes and 16 nodes when comparing the previous execution model and the new execution model show that the resources of the system are used more efficiently and the execution time is significantly reduced (decreased at least by 10%). This shows that the new execution model is a good direction to pursue the development of CAPE in the future.

7 Conclusion and future works

In this paper, we presented and analysed the disadvantages of previous version of CAPE. We also proposed a new method name TICKPT to improve the previous checkpointing technique. Then, TICKPT is applied to improve the previous execution model. Both theoretical analysis and experimentation showed that checkpoint's size and risk of bottlenecks in execution model are reduced significantly while the performance and the flexibility of CAPE are improved.

For the future, we will keep on developing CAPE using this new execution model. We will also try to determine how to combine checkpoints more efficiently to implement OpenMP's shared-data environment variables.

Acknowledgement

We thank the technical program committee members of SoICT 2017 conference for their reviews and comments that greatly improved the manuscript. We also thank Prof. Zhenjiang Hu, chair of the Software Engineering section for the discussion at the conference, that provided us ideas to extend this article.

References

[1] Message Passing Interface Forum (2014) MPI: A Message-Passing Interface Standard, `http://mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf`.

[2] OpenMP ARB (2013) OpenMP application program interface version 4.0, `http://www.openmp.org`.

[3] Morin, Christine and Lottiaux, Renaud and Vallée, Geoffroy and Gallard, Pascal and Utard, Gaël and Badrinath, Ramamurthy and Rilling, Louis (2003) Kerrighed: a single system image cluster operating system for high performance computing, *Euro-Par 2003 Parallel Processing*, Springer,

pp. 1291–1294. https://doi.org/10.1007/978-3-540-45209-6_175.

[4] Sato, Mitsuhisa and Harada, Hiroshi and Hasegawa, Atsushi and Ishikawa, Yutaka (2001) Cluster-enabled OpenMP: An OpenMP compiler for the SCASH software distributed shared memory system, *Scientific Programming*, Hindawi, pp. 123–130. http://doi.org/10.1155/2001/605217.

[5] Karlsson, Sven and Lee, Sung-Woo and Brorsson, Mats (2002) A fully compliant OpenMP implementation on software distributed shared memory, *High Performance Computing—HiPC 2002*, Springer, Berlin, pp. 195–206. https://doi.org/10.1007/3-540-36265-7_19.

[6] Basumallik, Ayon and Eigenmann, Rudolf (2005) Towards automatic translation of OpenMP to MPI, *Proceedings of the 19th annual international conference on Supercomputing (SC)*, ACM, pp. 189–198. https://doi.org/10.1145/1088149.1088174.

[7] Dorta, Antonio J and Badía, José M and Quintana, Enrique S and de Sande, Francisco (2005) Implementing OpenMP for clusters on top of MPI , *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, Springer, pp. 148–155. https://doi.org/10.1007/11557265_22.

[8] Huang, Lei and Chapman, Barbara and Liu, Zhenying (2005) Towards a more efficient implementation of OpenMP for clusters via translation to global arrays, *Parallel Computing*, Elsevier, pp. 1114–1139. https://doi.org/10.1016/j.parco.2005.03.015.

[9] Hoeflinger, Jay P (2006) Extending OpenMP to clusters, *White Paper*, Intel Corporation.

[10] Renault, Éric (2007) Distributed Implementation of OpenMP Based on Checkpointing Aided Parallel Execution, *A Practical Programming Model for the Multi-Core Era*, Springer, pp. 195–206. https://doi.org/10.1007/978-3-540-69303-1_22.

[11] Plank, James S and Beck, Micah and Kingsley, Gerry and Li, Kai (1994) Libckpt: Transparent checkpointing under unix, *White Paper*, Computer Science Department. https://pdfs.semanticscholar.org/bd21/4d6a94edf9bd4f97b4467c545dafd8138e8a.pdf.

[12] Ha, Viet Hai and Renault, Éric (2011) Discontinuous Incremental: A new approach towards extremely lightweight checkpoints, *Computer Networks and Distributed Systems (CNDS)*, IEEE, pp. 227–232. https://doi.org/10.1109/CNDS.2011.5764578.

[13] Ha, Viet Hai and Renault, Eric (2011) Design and performance analysis of CAPE based on discontinuous incremental checkpoints, *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, IEEE, pp. 862-867. https://doi.org/10.1109/PACRIM.2011.6033008.

[14] Tran, Van Long and Renault, Eric and Ha, Viet Hai (2016) Analysis and evaluation of the performance of CAPE, *IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress*, IEEE, pp. 620–627. https://doi.org/10.1109/UIC-ATC-ScalCom-CBDCom-IoP-SmartWorld.2016.0104.

[15] Ha, Viet Hai and Renault, Eric (2011) Improving performance of CAPE using discontinuous incremental checkpointing, *High Performance Computing and Communications (HPCC)*, IEEE, pp. 802–807. https://doi.org/10.1109/HPCC.2011.114.

[16] Tran, Van Long and Renault, Éric and Do, Xuan Huyen and Ha, Viet Hai (2017) Design and implementation of a new execution model for CAPE, *Proceedings of the Eighth International Symposium on Information and Communication Technology (SoICT's 2017)*, ACM, pp. 453–459. https://doi.org/10.1145/3155133.3155199.

[17] Bernstein (1966) Analysis of Programs for Parallel Processing, *IEEE Transaction on Electronic Computers*, IEEE, pp. 757–763. https://doi.org/10.1109/PGEC.1966.264565.

[18] Cores, Iván and Rodríguez, Mónica and González, Patricia and Martín, María J (2016) Reducing the overhead of an MPI application-level migration approach, *Parallel Computing*, Elsevier, pp. 72–82. https://doi.org/10.1016/j.parco.2016.01.012.

[19] Li, C-CJ and Fuchs, W Kent (1990) Catch-compiler-assisted techniques for checkpointing, *Fault-Tolerant Computing (FTCS)*, IEEE, pp. 74–81. https://doi.org/10.1109/FTCS.1990.89337.

[20] Chen, Zhengyu and Sun, Jianhua and Chen, Hao (2016) Optimizing Checkpoint Restart with Data Deduplication, *Scientific Programming*, Hindawi. https://doi.org/10.1155/2016/9315493.

[21] Plank, James S and Xu, Jian and Netzer, Robert HB (1995) Compressed differences: An algorithm for fast

incremental checkpointing, *Technical Report CS-95-302*, University of Tennessee.

[22] Hyochang, NAM and Jong, KIM and Hong, Sung Je and Sunggu, LEE (1997) Probabilistic checkpointing, *Proceedings of IEEE 27th International Symposium on Fault Tolerant Computing*, IEEE. https://doi.org/10.1109/FTCS.1997.614077.

[23] Mehnert-Spahn, John and Feller, Eugen and Schoettner, Michael (2009) Incremental checkpointing for grids, *Proceedings of the Linux Symposium*, Montreal, Quebec, Canada, pp. 201–220.

[24] Cores, Iván and Rodríguez, Gabriel and González, Patricia and Osorio, Roberto R and others (2013) Improving scalability of application-level checkpoint-recovery by reducing checkpoint sizes, *New Generation Computing*, Springer, pp. 163–185. https://doi.org/10.1007/s00354-013-0302-4.

[25] Alfred, V.Aho and Monica, S. Lam and Ravi, Sethi and Jeffrey, D. Ullman (2006) *Compilers Principles, Techniques,& Tools*, Addion Wesley.

[26] Thakur, Rajeev and Rabenseifner, Rolf and Gropp, William (2005) Optimization of collective communication operations in MPICH, *International Journal of High Performance Computing Applications*, Sage Publications, pp. 49–66. https://doi.org/10.1177/1094342005051521.

SHIOT: A Novel SDN-based Framework for the Heterogeneous Internet of Things

Hai-Anh Tran, Duc Tran, Linh-Giang Nguyen, Quoc-Trung Ha and Van Tong
School of Information and Communication Technology
Hanoi University of Science and Technology, Vietnam
E-mail: anhth@soict.hust.edu.vn

Abdelhamid Mellouk
Networks and Telecommunications Department
University of Paris-Est Creteil, France
E-mail: mellouk@u-pec.fr

Keywords: software defined networking, internet of things, Lagrange relaxation

Received: February 26, 2018

A new measurable and quantifiable world is created by the Internet of Things. However, the variety of IoT components, i.e., devices, access technologies and applications, which are deployed on the same core infrastructure with a common network policy have led to an unexpected issue of heterogeneity. Such issue directly dismisses the interoperability in the IoT, and hence, significantly decreasing the QoS of a given IoT service. In this paper, we develop a SDN-based framework, called SHIOT to address the above challenge. SHIOT relies on the ontology for examining the end-user requests and applies a SDN controller to classify flow scheduling over the task level. We also utilize the Lagrange relaxation theory to optimize the routing mechanism. Extensive experiments demonstrate that SHIOT is able to support stressed networks and offers a significant advantage over the traditional framework that is integrated without SDN.

Povzetek: Zasnovana je izboljšava heterogenega interneta stvari na osnovi SDN okvira in ontologij.

1 Introduction

The practical involvement of the Internet of Things (IoT) in our world is indisputable. The IoT contributes to the world's economy and improves the quality of life. There are 9 billion networking devices and the number is expected to reach 24 billion by 2020 [11]. The devices can be sensors, actuators, cameras, smartphones, which are placed in different access networks using different access technologies (e.g., bluetooth, wifi, zigbee, cellular, MANET). There are also a variety of IoT applications, implemented on top of the access networks. However, all these various elements, i.e., devices, access technologies, applications have to be built on a common network infrastructure with traditional equipments and communication protocols. The latter is not designed to support high-level of scalability, high amount of traffic and mobility in different IoT tasks. For this reason, it is required to develop diverse policies for the core network to adapt to the heterogeneous IoT. This challenge can not be accomplished without the help of a technology solution, namely Software Defined Networking (SDN) [12, 27].

SDN is considered as a network architecture that is able to enhance the flexibility of traditional core network. The idea of programmable networks facilitates network evolution, in which the forwarding devices are decoupled from the so-called control layer. This structure makes the behavior of the core network more adaptive to the quality of service, required by different access technologies, applications and devices. In addition, the centralized architecture in SDN gives the capability of collecting data and uses this information to improve network policies instead of manually transforming these high-level policies into low-level configuration commands.

In this paper, we proposed a SDN-based framework, named **SHIOT** (**S**dn-based framework for the **H**eterogeneous **IoT**) to tackle the heterogeneity issue in the IoT. The goal is to provide a transparent bridge between user-interface layer and other low-level layers in order to enhance the user convenience, while reducing the low-level complexity. To this end, we have developed an open ontology to analyze semantics of incoming requests. We also introduce a global optimization for routing layer of the core network by using a heuristic algorithm based on Lagrange relaxation theory. The characteristics of SDN help us to the above requirements without altering the untouchable core Internet network.

The paper is structured as follows: Section 2 presents the background and related works. Section 3 analyzes and describes in detail the SHIOT framework. Section 4 shows the experimental results. Finally, section 5 is dedicated to conclusions and future works.

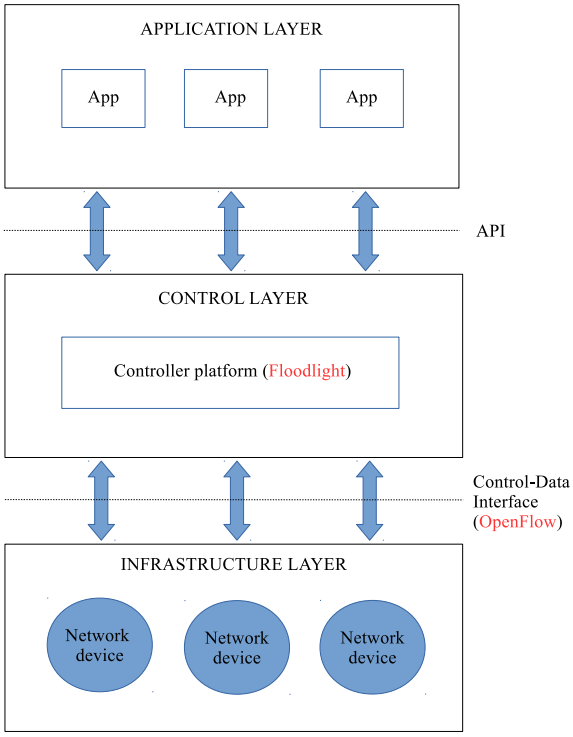


Figure 1: A general SDN architecture

2 Background and related works

2.1 Software defined networking

Over the past decade, the need for services that span multiple IoT application domains is growing in order to realize the efficiency gains, promised by the IoT. End-users, however, have to face the heterogeneity issue, which arises when a wide variety of devices, wireless communication solutions, and access technologies are implemented in the IoT. SDN is considered as a good solution to handle such issue because of its centralized and programmable controller that enables simple programmatic control of the network data-path.

The main idea of SDN is to separate the control and data planes. The control plane creates and modifies the forwarding rules, which are subsequently sent to network devices. The network devices (e.g., switches and routers) in turn, just forward packages based on the received rules. The controller is therefore considered as control logic precept to examine the overall network behavior. Using SDN-based controller, network administrators can easily program, manipulate and configure network protocols in a centralized way. Fig. 1 shows a general architecture of SDN, which consists of three main layers: Application layer, Control layer and Infrastructure layer.

In order to deploy a SDN architecture, it is essential to have an interface that ensures the communication between the data and control plane. Such interface is called Southbound Interface (SBI) and should be standardized. SBI defines a protocol to facilitate the diversity of network devices and controller softwares. There are variety of SBI protocols (e.g. ForCES [8]), but the most typical one seems to be OpenFlow [19]. An OpenFlow-enabled networking switch needs to maintain a forwarding flow table that has three types of information: rules, actions associated with each rule, and the statistics that count the number of packets and bytes for the flow. The OpenFlow-enabled switch aslo creates a secure channel to communicate with the SDN controller. We have chosen OpenFlow in the present work because it is capable of reducing management complexity, handling high bandwidth and implementing new policies when required. A detailed description on the advantages of OpenFlow can be found in [19].

2.2 SDN-enabled IoT: a review

The section gives a brief overview on the applicability of SDN in the IoT to improve the system performance and overcome the heterogeneity issue.

Huang et al. [13] developed an M2M based framework, which involves M2M nodes, a gateway to handle the nodes that does not support M2M protocol and a controller to carry out network management. Once the routing table is changed, the controller notifies such change and sends it to the different nodes. Therefore, the durability of the IoT network is improved. However, the authors only applied the same routing policy for all types of flows and did not consider their QoS.

Martinez-Julia and Skarmeta [18] utilized SDN that allows different objects from different networks to communicate with each other using IPv6. The IoT controller is also inserted to simplify the control operations of the various objects. After establising forwarding rules, such controller sends these rules to the SDN controller and other networking devices. Such mechanism enables the communication between the herotereous objects. Li et al. [17] and Omnes et al. [21] exploited SDN to contruct a framework that meets the requirement of the diversity and dynamics in the IoT. Vilalta et al. [28] developed an end-to-end orchestration for IoT services using an SDN/NFV-enabled edge node under SDN.

In [15], Jararweh et al. introduced an architecture model, namely SDIoF based on the combination of SDStore [5] and SDSec [2]. This architecture consists of three main components: 1) The physical layer where all the assets and hardware devices in the system reside. This layer is classified into several clusters such as sensor network cluster and database pool cluster; 2) The control layer acts as a middleware. It involves IoT controller, SDN controller, SDStore controller, and SDSec controller that are entirely software-based controllers to abstract the management operations from underling physical layer; 3) The application layer combines many fine-grained user applications, which simplify the end-user's accessing to the stored data through the Northbound APIs (N-APIs). Unfortunately, the authors in [15, 17, 21, 28] only gave the general and theoretical ideas. Their proposals have not been proven and assessed

through experiments.

Wei et al. [25] introduced a hash-based distributed strategy while integrating SDN into IoT in order to solve the problem of storage limitation of forwarding nodes. In their work, the multi-dimension selection method was utilized for finding the suitable storage. The hash space was formed by using the IoT data flow. However, the authors did not consider the QoS requirements related to the different IoT applications/services.

Chakrabarty et al. [3] developed the so-called Black SDN with the focus on various security issues. Black SDN was used to secure both the meta-data and payload within each layer of an IoT communication packet. The authors considered the SDN centralized controller as a trusted third party to ensure secure routing and optimize the system performance management. Olivier et al. [20] proposed a SDN-based architecture for the IoT that consists of multiple domains, where the SDN controllers are installed. The authors agued that such architecture is able to guarantee the security of the entire network. However,there is no simulation or evaluation to justify their architecture and outcomes. Sharma et al. [26] introduced DistBlockNet, a distributed secure SDN for the IoT, where blockchain technology was exploited to verify a version of the flow-rule table. However, in their experiments, they did not consider the average end-to-end delay, the most important QoS parameter that has a significant impact on the user experience while running an IoT service.

It is clear that SDN is an emerging architecture, which is able to facilitate network management in order to improve network performance and monitoring. However, there is a lack of evaluation and testing to assess its use in the IoT. This paper aims to address this question by developing SHIOT framework to bridge the gap between end-users and the underlying layers. SHIOT will be discussed in detail in the following section.

3 The proposed SHIOT

This section presents SHIOT framework, a layered SDN controller that acts as a middleware to bring the transparency to users from the top layer to the bottom ones. The controller (Fig. 2) is developed using an open source platform called *Floodlight* [1]. We decided to use the Floodlight Open SDN Controller due to the fact that it is an enterprise-class, Apache-licensed, Java-based OpenFlow Controller. Floodlight is supported by a community of developers including a number of engineers from Big Switch Networks. Although the project is discontinued, its latest version is sufficient for our development.

Since the IoT scenario is very dynamic, we create a database, called *State info DB* to store on the fly the state information of the network such as its topology, the link state, the joining or leaving nodes, etc. The end-user's commands that determine what he/she needs are placed at the highest layer of abstraction. We also construct a web service that provides RESTful APIs, so that the end-user may use such service with any platform and on any device. The transparency, considered in this paper is defined as the level of independence from applications, devices and networks, which are used to accomplish the required tasks. For example, the user's command is to count the number of people in Room01. The controller analyzes this command and indentifies the relevant service. Such service can be either to capture the video in Room01 and count the number of people inside or to activate the counting sensor on the door of the room. The devices can be either the camera or counting sensors. In other words, the role of SDN controller is to map the devices to a specific application in order to perform the couting process. Lower layer then selects network and path to route the data flows. Finally, all these decisions are sent down to Communication layer, and installed on the selected devices. The operation of Communication layer has been processed by a network emulator, known as *mininet* [6] in our testbed.

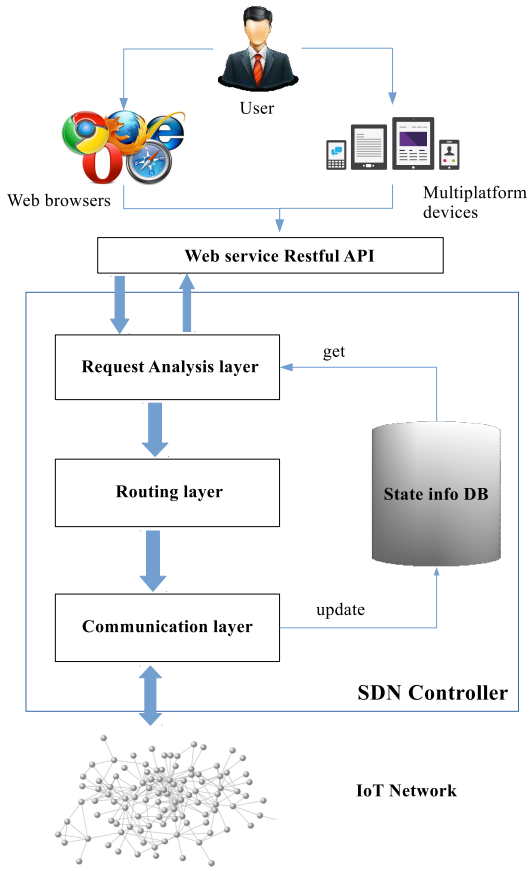


Figure 2: The proposed SHIOT framework architecture.

3.1 Request analysis layer

As mentioned above, there is always a challenge in the IoT due to the diversity in access networks and devices. Hence, the Request Analysis layer is used to provide an abstract

layer to end-users, so that the IoT may work independently from the underlying layers. In other words, the Request Analysis layer bridges the gap between user requests and underlying networking devices. The present work constructs an ontology and utilizes semantic technologies to describe the IoT context as well as the devices and their characteristics. We set our focus in the IoT that is deployed in E-healthcare system. Fig. 3 shows the constructed ontology, which involves three main classes as follows.

1. **Applications**: We considers five different healthcare applications:
   - Monitoring: This is used to capture and record the various healthcare indicators including the physiological (i.e., ECG, EMG, EEG), chemical (sweat, glucose, saliva), and optical (oximetry, the properties of tissues) metrics.
   - Therapeutic: The goal is to monitor the treatment of a given disease. This consists of medication (drug delivery patches), stimulation (chronic pain relief) and emergency (defibrillator).
   - Fitness and Wellness: The application aims to observe the motion and location indicators such as physical activity, calorie count, GPS information and indoor localization.
   - Behavioral: This application is used to maintain regular surveillance over the patient's activities (fall, sleep, exercise), emotions (anxiety, stress, depression) and diet (calorie intake, eating habits).
   - Rehabilitation: This application is used to monitor the rehabilitation of patients like speech (language development) and camera (technology for blinds).
2. **Devices**: There are two main types of devices, i.e., on-body contact sensors and peripheral non-contact sensors.
3. **Positions**: The positions where the devices locate include the laboratory, operating rooms, casualty rooms, consulting rooms, day rooms, emergency rooms, pharmacy, high dependency unit, maternity ward.

Based on the above classes in the ontology, the *Positions* and *Devices* are used to determine the sender and receiver nodes. We also assign a predefined maximum latency to each requested application in *Applications* class. This assignment is based on [14, 23, 22]. In order to validate the scalability of SHIOT, the current ontology includes 250 devices, deployed in 60 different rooms. However, such ontology can be extended and customized to be suitable for the use of any other organizations.

The delta delay Δ_{delay}, that was selected as the QoS metrics in this work, will be one of the three outputs, extracted from this layer. The two remaining outputs will be the sender and receiver nodes. Specifically, we construct two modules in the Request Analysis layer. As illustrated in Fig. 4, the *User expression analysis* module simply scans the expression text, sent from the various applications and detects keywords that are related to the constructed ontology. The *Ontology analysis* module in turn uses these keywords as input and relies on the *Ontology data* to determine the appropriate applications and the forwarding IoT devices. Finally, the outputs, i.e., Δ_{delay}, *the sender node* and *receiver node* will be sent to the Routing layer.

3.2 Routing layer

Based on the results, computed in the Request Analysis layer, the role of the Routing layer is to find an optimal path to route the data from the selected source to destination nodes. This is to ensure that the latency time is not larger than a predefined threshold Δ_{delay} and thus, optimizing the cost function.

To this end, we implemented a routing algorithm that is based on the concept of aggregate cost. The optimal cost is found using the Lagrange relaxation theory. Such algorithm offers a significant advantage because it can give a lower bound on the theoretically optimal solution. The experimental results showed that the difference between the obtained cost and the lower bound is naturally quite narrow. In addition, the proposed routing algorithm takes into account the trade-off between the end-to-end (E2E) delay and quality of the found path. Gary et al. [10] proved that a routing problem is NP-complete in the case where the number of QoS metrics that should be minimized is more than or equal to 2. Therefore, the paper tries to define a simpler problem instead of tackling the more complex problem. Particularly, the delay along the path should be acceptable, and the cost of the path should be as low as possible. The intuitive motivation of the routing task is to find a path that is minimal in terms of cost, provided that the delay is under a given bound. The delay bound is determined in the Request Analysis layer and the routing problem is formulated as a Delay Constrained Least Cost path problem (DCLC) [24].

The DCLC problem can be described as follows: A communication network is modeled as a directed and connected graph $G = (V, E)$, where E denotes a set of directed links and V represents a set of nodes (e.g., switches, routers), connected by directed links. Any node is reachable from any other node in this graph. Every directed link $e = (u, v) \in E$ has a delay $D(e)$ and a cost $C(e)$ associated with it. The link delay, i.e., $D(e)$ is measured when a packet is passing through link e. The link cost, i.e., $C(e)$ represents some other metrics, required to optimize such as the loss-rate, bandwidth, jitter, etc. The link cost is computed using Eq. 1 as

$$C(e) = w_1 * l_e + w_2 * b_e \tag{1}$$

where w_1 and w_2 are weights corresponding to the loss-rate (l_e) and bandwidth (b_e) metrics of link e, respectively.

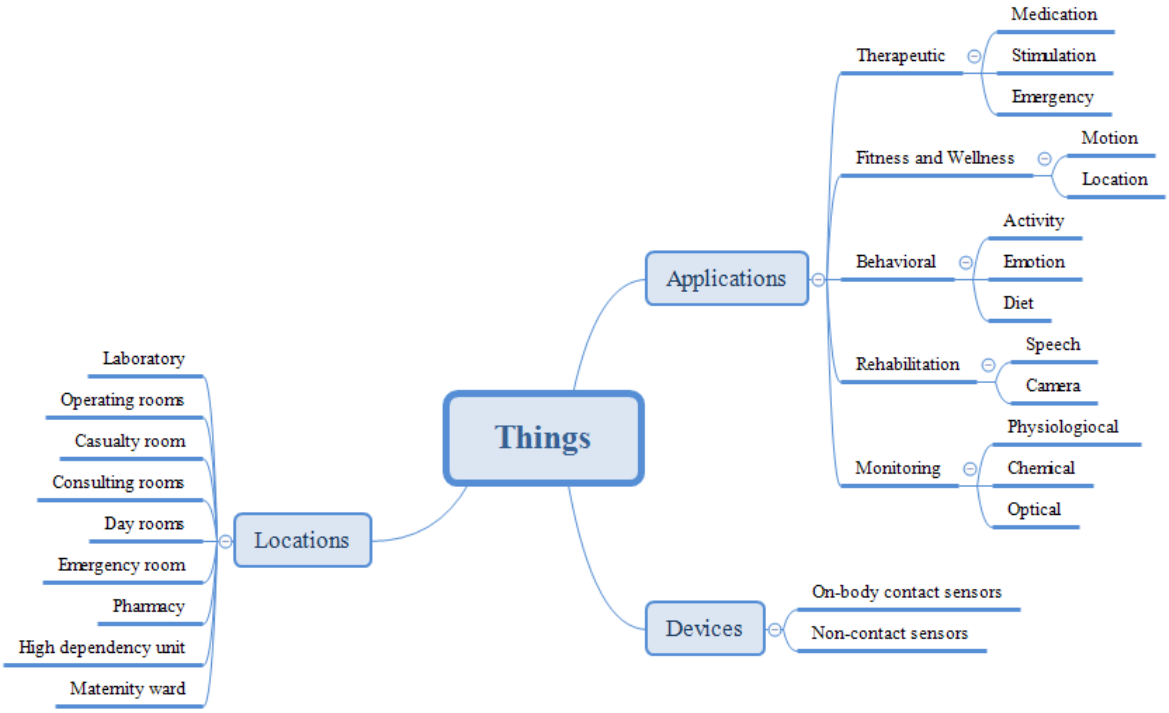


Figure 3: Ontology of the E-healthcare system

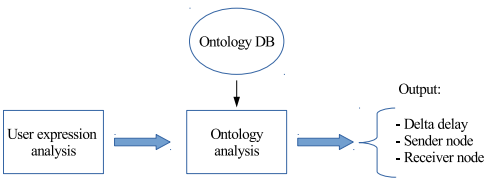


Figure 4: Request analysis module

The constrained minimization problem is represented as follow:

$$\min_{r \in R'(x,y)} \sum_{e \in r} C(e) \tag{2}$$

where $R'(x, y)$ is the set of routing paths r from source x to destination y for which the end-to-end delay is bounded by Δ_{delay}. Δ_{delay} is determined by the Resquest Analysis layer. We also have $R'(x, y) \subseteq R(x, y)$. A $r \in R(x, y)$ is in $R'(x, y)$ if and only if

$$\sum_{e \in r} D(e) \leq \Delta_{delay}$$

In order to solve the DCLC problem, we utilize a heuristic algorithm based on the Lagrange relaxation theory. This is considered as a common technique for determining lower bounds and finding solutions for this problem. The idea is based on the minimization of the modified cost function, where the cost and delay in terms of a parameter γ are combined to form an aggregate weight for each link as follows.

$$C_\gamma = C_{\text{old}} + \gamma.d \tag{3}$$

where d is the delay, C_{old} is the cost that is calculated using Eq. 1. For a given γ, the optimal path, denoted as r_γ, can be found by using Dijkstra's algorithm w.r.t. the cost c_γ, obtained in Eq. 3. If the total delay of the path r_γ, denoted as $D(r_\gamma)$, is equal or less than Δ_{delay}, it is the optimal solution. Otherwise, if $D(r_\gamma) > \Delta_{delay}$, the value of γ is increased in order to increase the influence of the delay factor in the cost function (see Eq. 3). The relationship between parameter γ, the cost and the delay of a given path can be illustrated by the following lemmas.

Lemma 1. *If* $0 \leq \gamma_1 \leq \gamma_2$ *then* $D(r_{\gamma 1}) \leq D(r_{\gamma 2})$ *and* $C(r_{\gamma 1}) \geq C(r_{\gamma 2})$.

Proof. See [9]. □

Lemma 1 shows that a larger γ will lead to a larger cost and a smaller delay. This implies that as long as the resulting shortest path does not violate the predefined delta delay, a smaller γ will definitely result to a better solution. The next lemma is used to find the smallest γ value (i.e., γ related to the shortest path that does not violate Δ_{delay}).

Lemma 2. *If* $\gamma_1 < \gamma_2$, $D(r_{\gamma 1}) \neq D(r_{\gamma 2})$, $\gamma' = \frac{C(r_{\gamma 1}) - C(r_{\gamma 2})}{D(r_{\gamma 2}) - D(r_{\gamma 1})}$, *then* $C(r_{\gamma 1}) \geq C(r_{\gamma'}) \geq C(r_{\gamma 2})$, $D(r_{\gamma 1}) \leq D(r_{\gamma'}) \leq D(r_{\gamma 2})$.

Proof. See [9]. □

Lemma 2 shows that with $\gamma' = \frac{C(r_{\gamma 1})-C(r_{\gamma 2})}{D(r_{\gamma 2})-D(r_{\gamma 1})}$, the shortest path $r_{\gamma'}$ must have a delay between the delays of $r_{\gamma 1}$ and $r_{\gamma 2}$, and the cost is also between the costs of these two paths. The above two lemmas imply that the *least cost path* r_c has to be first computed using Dijkstra's algorithm w.r.t. the cost. If its delay is not greater than Δ_{delay}, then it must be the optimal solution. Otherwise, the *least delay path* r_d, i.e., the path, found by using Dijkstra's algorithm w.r.t. the delay should be obtained. If its delay is greater than the Δ_{delay}, no optimal solution can be achieved. If none of the above conditions are true, the algorithm begins an iterative procedure. In each iteration, r_d is updated with a better solution having a lower delay and r_c is updated with a better solution having lower cost.

Algorithm 1 Lagrange relaxation-based routing algorithm

Require: $source$, $dest$, C, D, Δ_{delay}
Ensure: Optimal path
$r_c \leftarrow$ Dijkstra$(source, dest, C)$
if $D(r_c) \leq \Delta_{delay}$ **then return** r_c
end if
$r_d \leftarrow$ Dijkstra$(source, dest, D)$
if $D(r_d) > \Delta_{delay}$ **then return** "No solution"
end if
while true **do**
  $\gamma := \frac{C(r_c)-C(r_d)}{D(r_d)-D(r_c)}$
  $\mathcal{P} \leftarrow$ Dijkstra$(source, dest, C_\gamma)$
  if $C_\gamma(\mathcal{P}) = C_\gamma(r_c)$ **then return** r_d
  else if $D(\mathcal{P}) \leq \Delta_{delay}$ **then**
    $r_d \leftarrow \mathcal{P}$
  else
    $r_c \leftarrow \mathcal{P}$
  end if
end while

The heuristic algorithm (see Algorithm 1) is described in detail as follows: First, we utilize the original cost function (Eq. 1) and find the *least cost path* using Dijkstra's algorithm. If the delay of this path meets the delay requirement Δ_{delay}, it is the optimal path. Otherwise, we find the *least delay path* and examine whether the delay of this path is greater than Δ_{delay}. We may then decide to start the loop or to stop the algorithm as there is no optimal solution that can be found. The γ parameter is computed as

$$\gamma := \frac{C(r_c) - C(r_d)}{D(r_d) - D(r_c)}$$

Such parameter is updated after each iteration. The Dijkstra's algorithm is used w.r.t the new value of cost C_γ. If the cost value of the new path is found to be equal to the cost value of the *least cost path*, the optimal path should be the *least delay path*. If not, if the delay of the new path is found to be smaller than Δ_{delay}, the *least delay path* is updated as the new path. Otherwise, the *least cost path* is considered as the new path. The loop is repeated until the optimal path is found.

4 Experiments

4.1 Experimental setup

In order to conduct the experiments, we implemented a testbed, which is illustrated in the Fig. 5. The user is able to send requests to the E-healthcare system using Restful APIs from any platforms and devices. In the present work, we developed an Android application that automatically creates and sends the RESTful requests. A load balancing mechanism with PC coordinator is also implemented to support high rate of requests. Concretely, the coordinator applies the Round Robin algorithm to distribute the requests to three *Request Analysis PCs* (RA1-3). These three RA PCs analyze the incoming requests and search for the appropriate outputs in the *Ontology DB* database. In the SDN controller, these outputs are then fed to the routing algorithm that determines the appropriate way to control the simulated network. In this paper, we utilize *mininet* [6] to emulate the network topology, which involves a number of different nodes representing the Openflow-enabled switches and IoT devices.

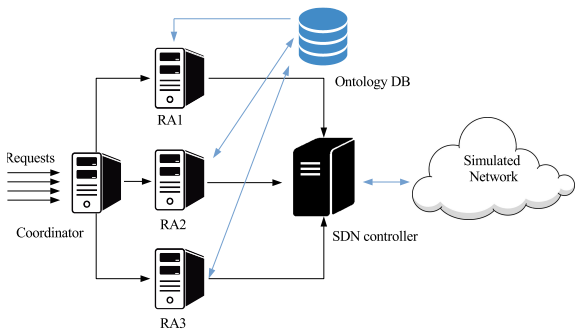


Figure 5: The testbed, implemented with load balancing mechanism

4.2 Analyzing the request analysis layer

First, we assess the scalability of the Request Analysis layer by varying the number of sending requests per second and evaluating the round trip time (RTT). In Fig. 6, it is obvious that the load balancing mechanism is able to support high request rate. It achieves a RTT of 2.2 seconds when the rate reaches 500 requests per second, while the RTT without load balancing mechanism is 26.8 seconds. This in turn, proves the scalability of the *Request Analysis layer*.

In order to evaluate the accuracy of the Request Analysis layer, we execute 10000 requests, and send them to five applications (2000 requests per application).

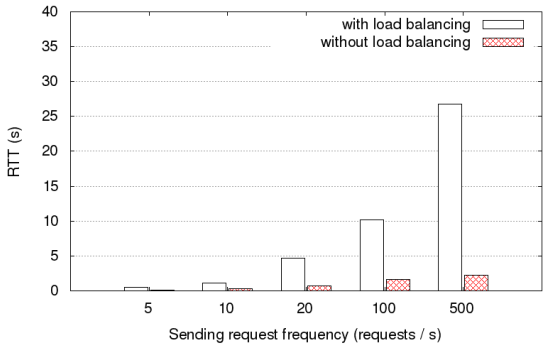


Figure 6: RTT corresponding to the Request Analysis layer, which is deployed with and without load balancing mechanism.

Applications	Number of user requests	Number of well-classified requests	Prop.
Monitoring	2000	1928	96.4 %
Therapeutic	2000	1987	99.3 %
Fitness and Wellness	2000	1979	98.9 %
Behavioral	2000	1893	94.6 %
Rehabilitation	2000	1912	95.6 %

Table 1: Accuracy related to the classification of user requests in the five applications

Table 1 shows that majority of user requests have been classified exactly for all five applications. The faults mostly come from the *Behavioral* application. This is due to the fact that the Request Analysis layer is based on the processing of text strings. The *Behavioral* application on the other hand, includes a variety of activity and emotion descriptions such as fall, sleep, exercise, anxiety, stress, depression, etc. Hence, it is more difficult to classify the user tasks. However, an accuracy of 94.6% is still acceptable for this kind of application.

4.3 Analyzing the routing layer

Concerning the performance of the routing layer, we implemented several other methods that also focus on the DCLC problem. These methods are as follows.

- The Constrained Bellman-Ford (**CBF**) routing algorithm [29], which is based on a breadth-first search that is able to update the lowest cost path in each visited node. CBF runs until either the highest constraint is exceeded or it cannot improve the paths anymore.
- The Multi-Criteria Routing algorithm (**MCR**) developed by Lee et al. in [16]: This routing algorithm is based on heuristics of ranked metrics in the network. A loop is repeated to determine the shortest path for each metric until the best path is found, or it fails for all metrics.
- The routing algorithm proposed by Cheng et al. [4] that combines the problems of finding the least cost and least delay paths by modifying the cost function. Such algorithm is abbreviated as MCF in the present work. It aims to compute a simple metric from multiple requirements using the weighted combination of the various QoS metrics.

The above mentioned algorithms are compared with the proposed algorithm (abbreviated as LARE in this paper) using the following measures:

- *Number of fails*, which is the average number of unreachable nodes, which occurs when the path cannot be found at a given delta delay Δ_{delay}. This measure aims at assessing the efficiency of a given algorithm in finding the destination nodes.
- *Delay*, which is the average delay time of the path from a source node to a reachable destination node.
- *Cost*, which is considered as the average cost of the paths from a source node to all reachable destination nodes.

Concerning the network topology, we first validate the routing algorithms and their functionality with a simple network including 17 nodes. We then carried out the experiments with NTT, a 37 node network topology [7] that is modeled using the exact characteristics of a real-world network. Finally, we construct by hand a huge network with 150 nodes (abbreviated as *150N* topology) to evaluate the scalability of the proposed framework.

To obtain the performance related to the various routing algorithms, this work varies the value of delta delay (Δ_{delay}) from 100 ms to 1000 ms in both the NTT and 150N topologies. As it can be seen from Figs. 7 and 8, when Δ_{delay} is smaller than 200 ms, the number of unreachable nodes is considerably high. This number decreases as the delay constraint is increased. All the paths are only found after 400 ms.

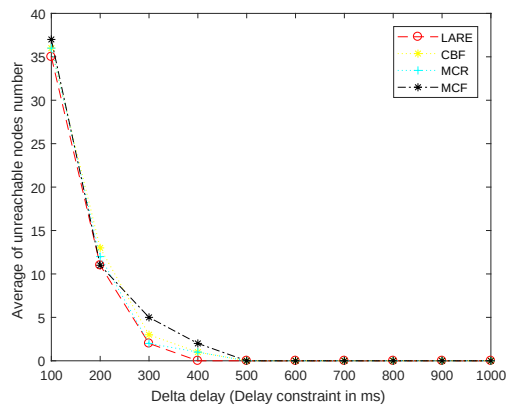


Figure 7: Average number of unreachable nodes in NTT topology

Figs. 9 and 10 show the average delay curves of the various algorithms in NTT and 150N topologies. As illustrated, with the low values of delta delay (ranging from 100 ms to

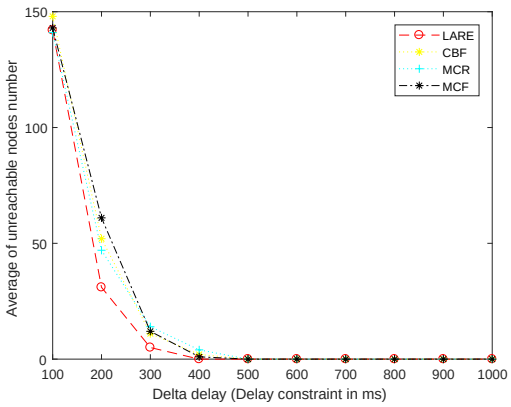


Figure 8: Average number of unreachable nodes in 150N topology

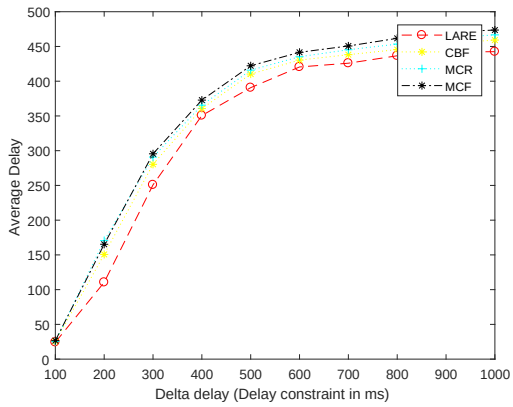


Figure 10: Average delay of the various routing algorithms in 150N topology

300 ms), the number of reachable nodes is small. The destination nodes are close to the source nodes, keeping the delay values at very low level. After 400 ms delta delay, when all the paths are found, the average delay becomes stable.

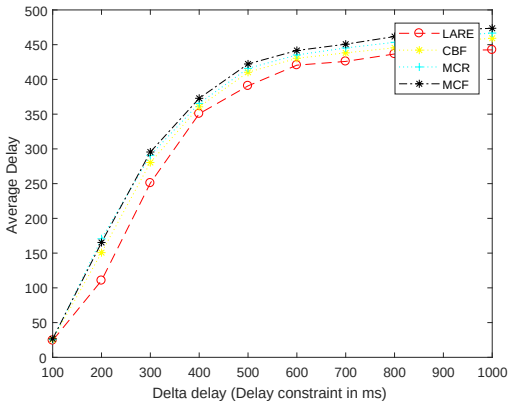


Figure 9: Average delay of the various routing algorithms in NTT topology

It is clear that the LARE algorithm provides the best result in terms of average delay in both the network topologies. The scalability of LARE has been proven in 150N (see Fig. 10), a large network having extremely high node density, where the performance gap between the proposed algorithm and other methods become more significant. At 1000 ms delta delay, LARE gives an average delay of 456 ms, while those of other methods are more than 660 ms. The MCF algorithm provides the highest average delay in both cases due to its difficulty to select an appropriate aggregate weights when combining the QoS metrics.

Figs. 11 and 12 shows the average cost of the paths found by the various algorithms in two network topologies. As explained above, at the beginning ($\Delta_{delay} < 300ms$) it is impossible to find paths to all destination nodes. Since the obtained paths are very short, the costs become relatively small. The algorithms find more paths as we increase the delta delay. When all destination nodes are found ($\Delta_{delay} \geq 300ms$), a higher delta delay would result in a lower cost. It is obvious that LARE algorithm gives almost similar outcomes to MCF in both the network topologies. This is exactly what we expected because MCF focuses on optimizing the cost value. MCR is the worst performer, since it only finds the best path for one metric, while ignoring the cost value. Especially, in the 150N topology (Fig. 12), LARE achieves an average cost of 19.5 at the delta delay of 1000 ms, while the MCR produces an average cost of 30.1.

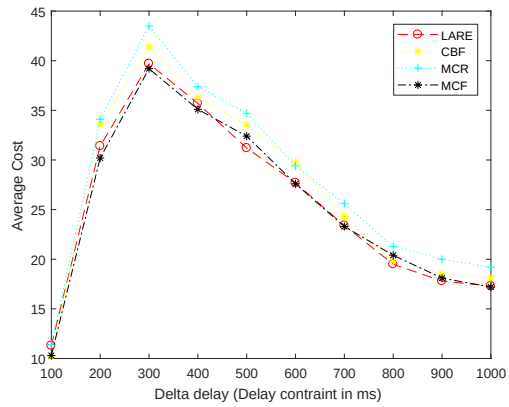


Figure 11: Average cost of the various routing algorithms in NTT topology

4.4 Analyzing the overall SHIOT framework

This section aims to validate the capability of supporting stressed network of the proposed framework and compared it with the traditional system, which relies on the simple best-effort policy and is implemented without SDN. Specifically, we try to stress the system by gradually in-

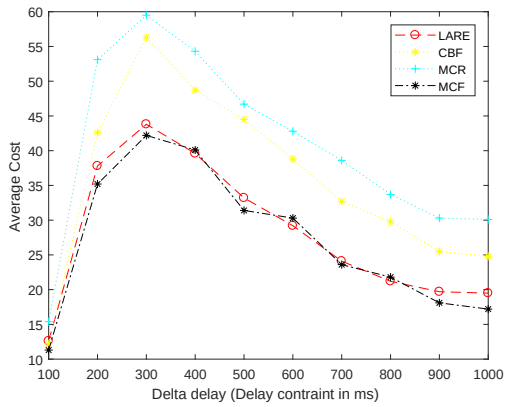


Figure 12: Average cost of the various routing algorithms in 150N topology

creasing the rate of sending requests from 100 to 2000 requests per second.

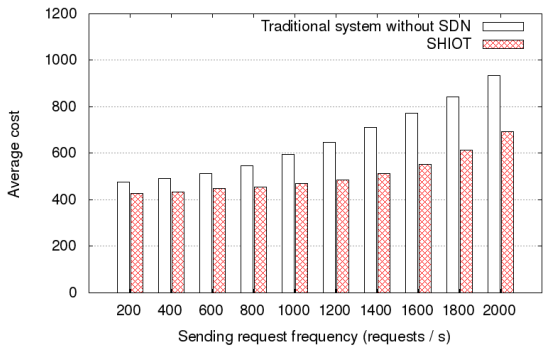


Figure 13: Capability of supporting stressed network of SHIOT and the traditional system in terms of delay.

We set the delta delay to the value of 1000 ms to ensure that the paths to all the destination nodes are found. As shown in Figs. 13 and 14, SHIOT is obviously better than the traditional couterpart in terms of delay and cost. Even in the most stressed state (i.e., 2000 requests per second), SHIOT is able to provide an average delay of 694 ms and a cost of 33, while those, achieved by the traditional system are 935ms and 69, respectively. The performance difference may be due to the simple policy (the best-effort policy) that is implemented in the core network of the traditional system. SHIOT, on the other hand, possess a layered architecture that is able to deal with the high rate of requests, sent from the various applications.

Finally, we evaluate the system performance while running the video application. Specifically, video flows are generated (video streaming) using an open source software, named VLC. Such flows are sent from one node to another in the simulated network. The experiment lasts five minutes. The delay and jitter are computed for each chosen path. In Table 2, we can see that SHIOT outperforms the conventional system. Its average end-to-end delay is about

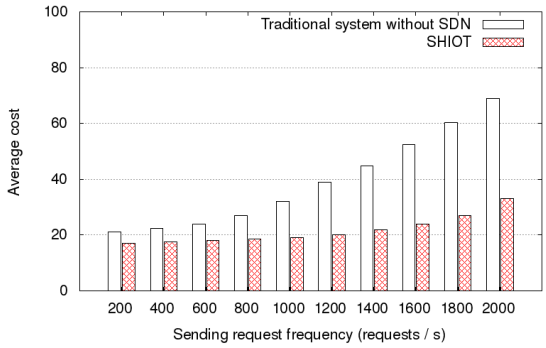


Figure 14: Capability of supporting stressed network of SHIOT and the traditional system in terms of cost.

3.5 s while that of the traditional system is 7.3 s. Similar observation can be obtained in terms of jitter. This is exactly what we expected because SHIOT has the ability to differentiate the different types of data flow (e.g. video, audio, information data).

5 Conclusions

In this paper, we have proposed a layered SDN framework, named SHIOT, to address the heterogeneity issue in the IoT. SHIOT is based on an open ontology to classify the incoming user requests. This framework also utilizes the Lagrange relaxation theory to find the optimal path in order to forward these requests to the destination nodes. In general, SHIOT can be considered as a remedy to bridge the gap between abstract high-level tasks and other low-level networks/devices. Experimental results showed that SHIOT yielded better performance when compared with the traditional system, that is deployed without SDN. It is also proved to be efficient and effective in handling tasks required by the various applications. Concerning the future works, we are in the process of evaluating SHIOT, which is implemented in the real devices (e.g. Openflow-enabled switches).

6 Acknowledgments

This work is supported by the Vietnamese Ministry of Science and Technology (MOST) under the Grant No. NDT.14.TW/16.

References

[1] The floodlight open sdn controller project. http://www.projectfloodlight.org/floodlight/.

[2] M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, A. Rindos, et al. Sdsecurity: A software defined security experimental framework. In *Communication Workshop (ICCW), 2015 IEEE Inter-*

Time (s)	SHIOT		Traditional system	
	delay (s)	jitter (s)	delay (s)	jitter (s)
20	7.3	0.93	8.3	0.9
40	9.2	0.84	8.7	0.7
60	9.1	0.52	9.3	0.8
80	7.7	0.68	8.5	0.6
100	4.3	0.34	7.6	0.62
120	5.2	0.26	7.2	0.57
140	3.5	0.33	6.8	0.61
160	4.5	0.41	7.1	0.63
180	3.8	0.27	7.4	0.61
200	4.7	0.43	7.6	0.58
220	3.9	0.39	7.2	0.62
240	3.5	0.52	7.3	0.63
260	3.2	0.45	7.8	0.60
280	3.3	0.42	7.2	0.62
300	3.5	0.33	7.1	0.59

Table 2: Comparison of SHIOT and the traditional system, while running video streaming application

national Conference on, pages 1871–1876. IEEE, 2015. DOI: https://doi.org/10.1109/ICCW.2015.7247453.

[3] S. Chakrabarty, D. W. Engels, and S. Thathapudi. Black sdn for the internet of things. In *Mobile Ad Hoc and Sensor Systems (MASS), 2015 IEEE 12th International Conference on*, pages 190–198. IEEE, 2015. DOI: https://doi.org/10.1109/MASS.2015.100.

[4] S. Chen and K. Nahrstedt. On finding multi-constrained paths. In *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, volume 2, pages 874–879. IEEE, 1998. DOI: https://doi.org/10.1109/ICC.1998.685137.

[5] A. Darabseh, M. Al-Ayyoub, Y. Jararweh, E. Benkhelifa, M. Vouk, and A. Rindos. Sdstorage: a software defined storage experimental framework. In *Cloud Engineering (IC2E), 2015 IEEE International Conference on*, pages 341–346. IEEE, 2015. DOI: https://doi.org/10.1109/IC2E.2015.60.

[6] R. L. S. de Oliveira, A. A. Shinoda, C. M. Schweitzer, and L. R. Prete. Using mininet for emulation and prototyping software-defined networks. In *Communications and Computing (COLCOM), 2014 IEEE Colombian Conference on*, pages 1–6. IEEE, 2014. DOI: https://doi.org/10.1109/ColComCon.2014.6860404.

[7] G. Di Caro, F. Ducatelle, and L. Gambardella. AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, 16(5):443–455, 2005. DOI: https://doi.org/10.1002/ett.1062.

[8] A. Doria, J. H. Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern. Forwarding and control element separation (forces) protocol specification. Technical report, 2010. DOI: https://doi.org/10.17487/RFC3746.

[9] G. Feng and C. Doulgeris. Fast algorithms for delay constrained leastcost unicast routing. *shortened version presented on INFORMS*, 2001.

[10] M. R. Gary and D. S. Johnson. Computers and intractability: A guide to the theory of np-completeness, 1979.

[11] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013. DOI: https://doi.org/10.1016/j.future.2013.01.010.

[12] F. Hu, Q. Hao, and K. Bao. A survey on software-defined network and openflow: from concept to implementation. *IEEE Communications Surveys & Tutorials*, 16(4):2181–2206, 2014. DOI: https://doi.org/10.1109/COMST.2014.2326417.

[13] H. Huang, J. Zhu, and L. Zhang. An sdn_based management framework for iot devices. In *Irish Signals & Systems Conference 2014 and 2014 China-Ireland International Conference on Information and Communications Technologies (ISSC 2014/CIICT 2014). 25th IET*, pages 175–179. IET, 2013. DOI: http://dx.doi.org/10.1049/cp.2014.0680.

[14] R. ITU-T. P. 862. *Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs*, 2001.

[15] Y. Jararweh, M. Al-Ayyoub, E. Benkhelifa, M. Vouk, A. Rindos, et al. Sdiot: a software defined based internet of things framework. *Journal of Ambient Intelligence and Humanized Computing*, 6(4):453–461, 2015. DOI: https://doi.org/10.1007/s12652-015-0290-y.

[16] W. Lee and M. Hluchyj. Multi-criteria routing subject to resource and performance constraints. In *ATM Forum*, volume 94, page 0280, 1994.

[17] J. Li, E. Altman, and C. Touati. A general sdn-based iot framework with nvf implementation. *ZTE communications*, pages 1–11, 2015.

[18] P. Martinez-Julia and A. F. Skarmeta. Extending the internet of things to ipv6 with software defined networking. Technical report, Euchina-fire, 2014.

[19] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2):69–74, 2008. DOI: https://doi.org/10.1145/1355734.1355746.

[20] F. Olivier, G. Carlos, and N. Florent. New security architecture for iot network. *Procedia Computer Science*, 52:1028–1033, 2015.

[21] N. Omnes, M. Bouillon, G. Fromentoux, and O. Le Grand. A programmable and virtualized network & it infrastructure for the internet of things: How can nfv & sdn help for facing the upcoming challenges. In *Intelligence in Next Generation Networks (ICIN), 2015 18th International Conference on*, pages 64–69. IEEE, 2015. DOI: https://doi.org/10.1109/ICIN.2015.7073808.

[22] I. Rec. Y. 1541: Network performance objectives for ip-based services. *International Telecommunication Union, ITU-T*, 2003.

[23] I. Recommendation. 910,"subjective video quality assessment methods for multimedia applications," recommendation itu-t p. 910. *ITU Telecom. Standardization Sector of ITU*, 1999.

[24] D. S. Reeves and H. F. Salama. A distributed algorithm for delay-constrained unicast routing. *IEEE/ACM Transactions on Networking (TON)*, 8(2):239–250, 2000. DOI: https://doi.org/10.1109/90.842145.

[25] W. Ren, Y. Sun, T.-Y. Wu, and M. S. Obaidat. A hash-based distributed storage strategy of flowtables in sdn-iot networks. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–7. IEEE, 2017. DOI: https://doi.org/10.1109/GLOCOM.2017.8254507.

[26] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park. Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks. *IEEE Communications Magazine*, 55(9):78–85, 2017. DOI: https://doi.org/10.1109/MCOM.2017.1700041.

[27] V. R. Tadinada. Software defined networking: Redefining the future of internet in iot and cloud era. In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on*, pages 296–301. IEEE, 2014. DOI: https://doi.org/10.1109/FiCloud.2014.53.

[28] R. Vilalta, A. Mayoral, D. Pubill, R. Casellas, R. Martínez, J. Serra, C. Verikoukis, and R. Muñoz. End-to-end sdn orchestration of iot services using an sdn/nfv-enabled edge node. In *Optical Fiber Communication Conference*, pages W2A–42. Optical Society of America, 2016. DOI: https://doi.org/10.1364/OFC.2016.W2A.42.

[29] R. Widyono et al. *The design and evaluation of routing algorithms for real-time channels*. International Computer Science Institute Berkeley, 1994.

USL: A Domain-Specific Language for Precise Specification of Use Cases and Its Transformations

Chu Thi Minh Hue[1], Dang Duc Hanh[2], Nguyen Ngoc Binh[3] and Le Minh Duc[4]
Department of Software Engineering, VNU University of Engineering and Technology
[1]Hung Yen University of Technology and Education, Vietnam
[2]Corresponding author
[3]Visiting professor, Hosei University, Japan
[4]Hanoi University, Vietnam
E-mail: {huectm.di12 | hanhdd | nnbinh}@vnu.edu.vn, duclm@hanu.edu.vn

Keywords: use cases, pre- and postcondition, labelled transition systems, model transformations, domain-specific languages

Received: March 29, 2018

A use case model is often represented by a UML use case diagram and loosely structured textual descriptions. The use case model expressed in such a form contains ambiguous and imprecise parts. This prevents integrating it into model-driven approaches, where use case models are often taken as the source of transformations. In this paper, we introduce a domain-specific language named the Use case Specification Language (USL) to precisely specify use cases. We define the abstract syntax of USL using a metamodel together with OCL wellformedness rules and then provide a graphical concrete syntax for the usability goal. We also define a precise semantics for USL by mapping USL models to Labelled Transition Systems (LTSs). It opens a possibility to transform USL models to software artifacts such as test cases and design models. We focus on a transformation from a USL model to a template-based use case description in order to illustrate our method. A language evaluation of USL is also performed in this paper.

Povzetek: Zasnovan je domensko specifični jezik USL za natančno specifikacijo primerov in transformacij.

1 Introduction

Use case is a software artifact that is commonly used for capturing and structuring the functional requirements. A use case is defined as "the specification of sequences of actions, including variant sequences and error sequences, that a system, subsystem, or class can perform by interacting with outside objects to provide a service of value" [1]. As a requirements artifact, the use case model is commonly specified by a UML use case diagram and loosely structured textual descriptions [2]. A key benefit of this use case specification is that it is easy for non-technical stakeholders to learn and use. However, the use case models expressed in this form often contain ambiguous and imprecise parts. This prevents the models from being used directly in model-driven approaches, as a transformation source to produce other analysis and design models. An important challenge here is how to achieve a balance between two seemingly conflicting goals: to specify use case sufficiently precise for model transformation purposes, while achieving the ease-of-use required by non-technical stakeholders.

To this end, a considerable number of works, including [3, 4, 5, 6, 7] and those discussed in [8], have attempted to introduce rigor into use case description. More specifically, T. Yue et al. [3] proposed adding keywords and restriction rules into use case descriptions and then using natural language processing techniques in order to analyze them. Unlike [3, 4, 9, 5, 6, 7], which used natural language description, the works in [10, 11] proposed a formal semantics for use case. On the other hand, UML activity and sequence diagrams are proposed in [12, 13, 14, 15] to model the control flows in use case. A number of other works [4, 16, 17] proposed using a domain specific language (DSL) to specify use case. DSL [18] is a language that is designed specifically for a certain domain to ease the task of describing concepts in the domain.

However, the main limitation of the existing work is that they do not focus on precisely capturing the relevant use case information. These include control flows, steps, system actions, actor actions, and constraints on the use case and its flows. In this paper, we propose a DSL named Use Case Specification Language (USL) to overcome this limitation. The goal of USL is to precisely specify use cases and its model transformation abilities. The USL's domain consists in the task of specifying use cases that capture the system behavior.

Our approach is to define the abstract syntax of USL by extending the metamodels of the UML use case and activity diagrams [2]. Our extension consists in a set of meta-concepts needed for the following purposes: (1) to describe the elements of a typical use case description template; (2) to represent the *basic* and *alternate flows* of a use case in the form of sequential, branched, repeating steps, or

concurrent steps; (3) to categorize *steps* and *actions* based on the interaction subjects, which include the *system*, *actors* and *included/extending use cases*; and (4) to represent *constraints* on the use case, *actions*, and *flows*.

Our precise specification of USL makes it possible to automatically transform USL models into other software artifacts using model transformation techniques. In brief, the main contributions of our work are as follows:

- A DSL named USL to precisely specify use cases. We define the abstract syntax of USL using a metamodel constrained by OCL wellformedness rules [19]. For usability, we define a graphical concrete syntax for USL.
- A formal semantics specification for USL using Labelled Transition System [20]. This semantics enables the automatic transformation of USL models into other software artifacts, such as test cases and class models.
- A support tool that includes a visual editor for constructing USL models. We use this tool and two commonly-used case studies to illustrate our method. We also evaluate USL by comparing it to other related languages.

This work makes four significant improvements from our earlier conference paper [21]. First, regarding to USL specification, we make the abstract syntax precise with the OCL wellformedness rules [19] and define a graphical concrete syntax. Second, we develop an additional case study in order to illustrate how to apply USL in practice. Third, we define a numqber of typical model transformation scenarios for USL model and explain, in more detail, the transformation into template-based use case description. Fourth, we provide an evaluation for USL.

The rest of this paper is organized as follows. Section 2 presents the background and an example for our work. Section 3 overviews our approach. Section 4 presents the USL abstract syntax and explains its formal semantics. Section 5 explains how USL models are transformed into other software artifacts. Section 6 introduces our support tool and illustrates how to apply USL to the *ATM* system case study. This section also presents an evaluation of USL. Section 7 comments on the related works. The paper is closed with the conclusions and future work.

2 Background and motivation

Figure 1 shows a simplified requirement model of a `Library system` including a UML use case model depicted in the part (a) and a UML class diagram capturing corresponding domain concepts of the system which is presented in the part (b). Our paper uses the use case *Lend Book* in the part (a) as a motivating example. This use case is invoked when the librarian executes the book lending transaction. The use case is represented in a typical template as shown in Table 1.

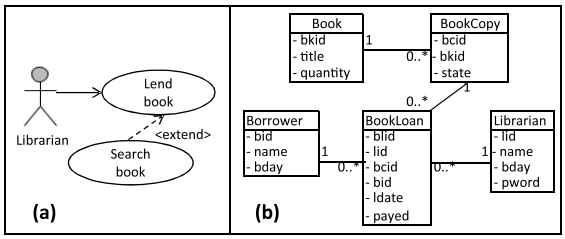


Figure 1: The simplified use case and the conceptual domain model of the `Library system`.

Table 1: A typical use case description template

Use case name: *Lend Book* **Brief description:** The Librarian processes a book loan. **Actors:** Librarian. **Precondition:** The librarian has logged into the system successful. **Postcondition:** If the use case successfully ends, the book loan is saved and a complete message is shown. In the other case, the system displays an error message. **Trigger:** The Librarian requests a book-loan process. **Special requirement:** There is no special requirement.
Basic flow
1. The Librarian selects the *Lend Book* function. 2. The system shows the Lend-book window, gets the current date and assigns it to the book-loan date. 3. The Librarian enters a book copy id. 4. The system checks the book copy id. If it is invalid, it goes to step 4a.1 5. The Librarian enters a borrower id. 6. The system validates the borrower id. If it is invalid, it goes to step 6a.1 7. The Librarian clicks the save-book-loan button. 8. The system validates the conditions to lend book. If it is invalid, the system goes to step 8a.1 9. The system saves the book loan record, then executing two steps 10 and 11 concurrently. 10. The system shows a complete message. 11. The system prints the borrowing bill.
Alternate flows
E1. request searched book 1. The Librarian selects the search function after step 4a.1. 2. The system executes the extending use case Search book. 4a. The book copy id is invalid 1. The system shows an error message, then going to step 3. 6a. The Borrower id is invalid 1. The system shows an error message, then going to step 5. 8a. The lending condition is invalid 1. The system shows an error message. 2. The system ends the use case.

A typical use case description template [22] often includes two parts, the overview information elements and the detailed description of flows. The first part consists of the following elements: the **use case name**, the use case's **brief description**, the **actors** participating in the use case, the use case's **precondition** and **postcondition**, the **trigger** that initiates the use case and the **special requirement** that describes the non-functional requirements of the use case. The second part contains two types of flows, the **basic flow** and **alternative flows**. The **basic flow** covers what normally happens when the use case is performed. Each use case description has only one **basic flow**. The **alternative flows** cover optional or exceptional behaviour as well as the variations of the normal behaviour. Both the **basic** and **alternative flows** are often further structured into steps or subflows [23, 1]. Moreover, one can smooth use case flows

to contain only a **basic flow** and some **alternate flows**.

Each step in flows consists of actions performed either by the system or actors. We refer to actors, the system, and other relation use cases as `interactive subjects`. For example, Step 1 in the **basic flow** is carried out by the Librarian actor, while Step 2 is performed by the system. A step may also contain the information to decide the next moving is another step or another flow or the starting or finishing of concurrent actions. As illustrated in Table 1, Step 2 includes three system actions, "The system shows the Lend-book window", "The system gets the current date" and "The system assigns the current time to the book-loan date". Step 5 contains a branching decision, "If it is invalid, the system goes to step 6a". Step 9 contains the starting point of two concurrent actions: "The system executes two steps 10 and 11 concurrently".

In our work, we consider sentences describing execution of an extending or an included use case as the system's actions. Our previous work [21] divides use case's actions into nine types as follows:

Actor-Input is an actor action to enter data into the system, e.g., the action "The Librarian enters a book copy id" at Step 3 in Table 1 is an `Actor-Input`.

Actor-Request is an actor action to send requests into the system, e.g., the action "The Librarian clicks the save-book-loan button" at Step 7 in Table 1 is an `Actor-Request`.

System-Display is a system action that the system performs operations with the user interface, e.g., the action "The system shows the lend-book window" at Step 2 in Table 1 is a `System-Display`.

System-Operation is a system action to validate a request and data, or process and calculate data, e.g., the action "The system gets the current date" at Step 2 in Table 1 is a `System-Operation`.

System-State is a system action to query or update its internal states, e.g., the action "The system saves the book loan record" at Step 9 in Table 1 is a `System-State`.

System-Output is a system action to send outputs to the actors, e.g., the action "The system shows an error message" at Step 1 of the alternate flow 4a in Table 1 is a `System-Output`.

System-Request is a system action to send requests to a secondary actor, e.g., the action "The system prints the borrowing bill" at Step 11 shown in Table 1 is a `System-Request`.

System-Include is a system action to include another use case.

System-Extend is a system action to extend another use case, e.g., the action "The system executes the extending use case Search book" within Step 2 of the alternate flow E1 in Table 1 is a `System-Extend`.

A use case is successfully executed only if the pre- and postcondition of the use case as well as of the actions of the current flow are satisfied.

Within the context of model-driven development, a use case model, as illustrated in Fig. 1 tends to be taken as a source model of transformations in order to obtain other software artifacts such as analysis models, design models, and test cases. However, the ambiguous and imprecise parts within use case descriptions prevents us from achieving such transformations. In order to integrate use cases into model-driven approaches, we aim to tackle the following challenges:

Capturing the overview structure. The use case model needs to preserve the overview structure of use case descriptions so that a template-based representation of use cases might be generated for non-technical stakeholders.

Specifying precisely control follows. A use case includes a set of scenarios, each of which corresponds to a control flow of the use case. Therefore, the use case model needs to preserve the information of control flows of use cases. This allows us to automatically generate artifacts like test scenarios and behaviour models.

Specifying precisely actions. The use case model needs to precisely represent actions within use case scenarios. A precise specification of actions allows us to capture use case relationships and to generate other artifacts from use cases such as class diagrams, test scenarios, and test objects.

Specifying use case constraints. For the aim to automatically generate test data, the use case model needs to preserve the constraints within use case descriptions, including the pre- and postcondition of use cases, the pre- and postcondition of use case actions, and their guard conditions.

3 Overview of the approach

Figure 2 illustrates our approach. First, we take as input a use case diagram, the textual descriptions of use cases, and a class diagram capturing the conceptual model of the system. Then, we aim to represent each use case specification as a model element of a so-called *use-case domain*. In order to define the use-case domain, we define meta-concepts w.r.t. the structural elements of the typical use-case-description template and the use case concepts as explained in Sect. 2. The meta-concepts allow us (1) to represent the basic and alternate flows of a use case in form of sequential, branched, or repeating steps, (2) to categorize use case steps and actions based on the interactive subjects including the system and actors, and (3) to represent constraints on the use case and its flows.

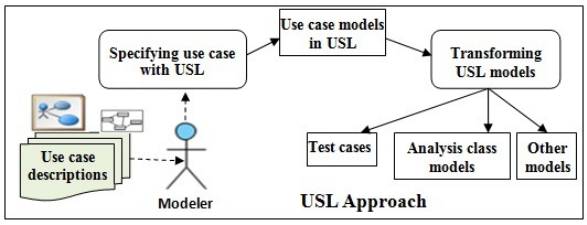


Figure 2: Overview of the USL Approach.

In order to represent textual descriptions of actions or

constraints within a use case specification, we consider them as operations on an object-oriented model w.r.t. the input conceptual model of the system. In that way, we could employ pairs of pre- and postcondition as contracts on actions in order to obtain a more precise specification of the use case. The constraints are often expressed using constraint languages such as the OCL [24], JML [25], and natural language as mentioned in [17]. In this research, we employ the OCL to represent the constraints.

Specifically, our approach is realized as follows. We propose a domain specific language named USL in order to represent use cases within the use-case domain. Further, we define a formal semantics of USL so that we could transform USL models in to other artifacts such as test cases and analysis class models. To illustrate this point, a transformation from a USL model to a template-based use case description will be explained in details in SubSect. 5.3.

4 The USL language

This section first explains the abstract syntax and the graphical concrete syntax of USL. Here, we utilize the metamodeling approach as mentioned in [26] to define USL. Then, we focus on defining a precise semantics for USL by mapping a USL model to a Labelled Transition System (LTS) [20].

4.1 The USL abstract syntax

We define the USL metamodel w.r.t. the use-case domain based on (1) UML use case specification (Chapt. 18 of [2]), (2) the Use Case Descriptions (UCDs) [1], [23], [22], and (3) the UML activity specification (Chapt.s 15, 16 of [2]). We will refer to these as the `domain sources` (1), (2), and (3), respectively.

Figure 3 shows the metamodel of USL. For brevity, we divide the metamodel into four blocks: (a), (b), (c), and (d). Figure 3-a (*i.e.*, block (a)) presents the top-level concepts. Figure 3-b presents the `FlowStep` hierarchy. Figure 3-c presents the `ControlNode` hierarchy. Figure 3-d presents the `Action` hierarchy and how it is related to the `FlowStep` hierarchy. Figure 3-e presents the concept `Constraint` and how it is used to specify `Action`, `InitialNode`, `FinalNode`, and `FlowEdge`.

To conserve space, we will not repeat here the definitions of all of the USL concepts that are described in the three domain sources. We will instead focus on a key sub-set of the concepts – those that will be used later to define the transformation of USL models. Figure 4 presents the USL model of the *Lend Book* use case as shown in Table 1. We will use this example USL model in order to illustrate our definitions.

Action (domain sources (1, 3)) represents a action that is performed either by an actor or by the system. An `Action` is characterised by the following attributes: `actionName` and `parameters`. The parameters are represented by concept `Parameter` inherited the concept `Parameter` of UML (as presented in Sect. 19.9.13 of [2]). `Action` is specialized into two main types (as illustrated in Fig. 3-d): `ActorAction` and `SystemAction`. `ActorAction` is further specialized into `ActorRequest` and `ActorInput`. `SystemAction` is specialized into `SystemOperation`, `SystemOutput`, `SystemDisplay`, `SystemState`, `SystemInclude` and `SystemExtend` that were explained in Sect. 2.

FlowStep (domain source (2)) is a sequence of `Actions` that represents a step in a basic flow or an alternate flow of the use case. It is characterised by the following attributes: `number` (the order number of step), `description` (the content of the step) and `maxloop` (the maximum iteration of the step if existing). `FlowStep` is specialized into two types (as shown in Fig. 3-b): `ActorStep` and `SystemStep`, as mentioned in Sect. 2. We define three utility functions as shown in Table 2.

Example 4.1.1. The USL model shown in Fig. 4 consists of the `FlowSteps` s1, ..., s16. Among these, s1 is an `ActorStep` and s2 is a `SystemStep`. Step s3 contains the `ActorInput` a5. Step s1 contains the `ActorRequest` a1. Step s2 contains `SystemOperation` a3. Step s10 contains the `SystemOutput` a12. Step s4 contains the `SystemState` a6. Step s11 contains the `SystemRequest` a13. Step s14 contains the `SystemExtend` a16. The `Action` a5 has `Parameter` "bcid".

Control Node (domain source (3)) represents a control action that regulates the flows across other `USLNodes`. A `ControlNode`, as illustrated in Fig. 3-c, is specialized into `InitialNode`, `FinalNode`, `DecisionNode`, `ForkNode` and `JoinNode`. These respectively represent the starting and ending points of use case, the branching points of steps, and the starting and ending points of concurrent actions in steps. To ease notation, we define two overloading functions w.r.t. `ControlNode` and a function w.r.t. `DecisionNode` as shown in Table 2.

Example 4.1.2. The USL model as shown in Fig. 4 contains nine `ControlNodes` c0, ..., c8. In particular, c0 is an `InitialNode`, c7 and c8 are different `FinalNodes`, c1, ..., c3 and c6 are `DecisionNodes`, c4 is a `ForkNode`, and c5 is a `JoinNode`.

USLNode represents all the nodes `FlowStep` or `ControlNode` that make up a USL model.

FlowEdge (domain source (3)) is a binary directed edge between two `USLNodes`. If both steps are a part of a basic flow, we call the transition a `BasicFlowEdge`. On the other hand, if both steps are a part of an alternate flow, we call the transition an `AlternateFlowEdge`. As shown in Table 2, we define two utility functions `source` and `target`, two overloading functions `guardE` and a function `isCompleted` w.r.t. the concept `FlowEdge`.

Example 4.1.3. The USL model as shown in Fig. 4 contains b1, ..., b18 as `BasicFlowEdges` and al_1, ..., al_10 as `AlternateFlowEdges`.

Variable (domain source (3)) represents variables that

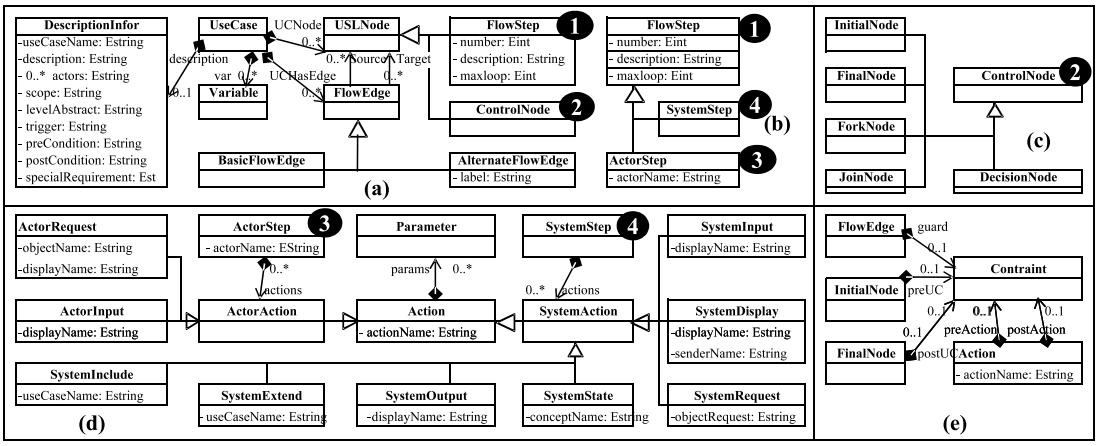


Figure 3: The USL metamodel.

Table 2: List of utility functions w.r.t. USL concepts

Utility function	Description
`firstAct`: `FlowStep` → `Action`	Returning the first `Actions` of a `FlowStep`.
`lastAct`: `FlowStep` → `Action`	Returning the last `Actions` of a `FlowStep`.
`actions`: `FlowStep` → `Actions`	Returning a set of `Actions` of a `FlowStep`.
`firstAct`: `ControlNode` → `ControlNode`	Returning the `ControlNode` itself.
`lastAct`: `ControlNode` → `ControlNode`	Returning the `ControlNode` itself.
`source`: `FlowEdge` → `USLNode`	Returning the source `USLNodes` of a `FlowEdge`.
`target`: `FlowEdge` → `USLNode`	Returning the target `USLNodes` of a `FlowEdge`.
`guardE`: `FlowEdge` → `Constraint`	Returning the guard condition.
`guardE`: `USLNode` → `USLNode` → `Constraint`	Taking the source and target `USLNodes` as input and returning the guard condition.
`isCompleted`: `FlowEdge` → `Boolean`	Determining whether or not `lastAct`(`source`(e)) has completed its execution.
`preA`: `Action` → `Constraint`	Returning the precondition of an `Action`.
`preA`: `ControlNode` → `Constraint`	If the `ControlNode` is not a `InitialNode`, returning true, else returning the `Constraint` of the `InitialNode`
`postA`: `Action` → `Constraint`	Returning the postcondition of an `Action`.
`postA`: `ControlNode` → `Constraint`	If the `ControlNode` is not a `FinalNode`, returning true, else returning the `Contraint` of the FinalNode.
`preC`: `USLModel` → `Constraint`	Returning the precondition of a `USLModel`.
`postC`: `USLModel` → `Constraint`	Returning the postcondition of a `USLModel`.
`postC`: `USLModel` → `FinalNode` → `Constraint`	Returning the postcondition of a particular `FinalNode` of a `USLModel`.

hold data values during the execution of a use case scenario. It is inherited the concept `Variable` of UML presented Sect. 15.7.25 of [2].

DescriptionInfor (domain source (2)) maintains the other textual description of use case.

Constraint (domain source (1,3)) represents constraints that are formed by use case variables: (1) the precondition of use case associated with `InitialNode`; (2) the postcondition of use case associated with `FinalNodes`; (3) guard conditions of a transition; and (4) the pre- and postcondition of an `Action`. This concept is inherited the concept `Constraint` in UML, shown in Sect. 7.6 of [2]. As depicted in Table 2, we define utility functions w.r.t. `Constraints` to get the pre- and postcondition of actions and use case.

Example 4.1.4. The USL model as shown in Fig. 4 contains g1, ..., g6 as guard conditions and p1, ..., p6 as postconditions of `Actions`.

We define a set of OCL wellformedness rules as restrictions on the USL metamodel. These rules are defined in the context of the `UseCase` concept and listed as follows.

Rule 1. A USL model has one `InitialNode`:

```
self.uslnode->selectByType(InitialNode)
->size()=1
```

Rule 2. A USL model has at least one `FinalNode`:

```
self.uslnode->selectByType(FinalNode)
->size() >= 1
```

Rule 3. A USL model has at least one `FlowStep`:

```
self.uslnode->selectByKind(FlowStep)
```

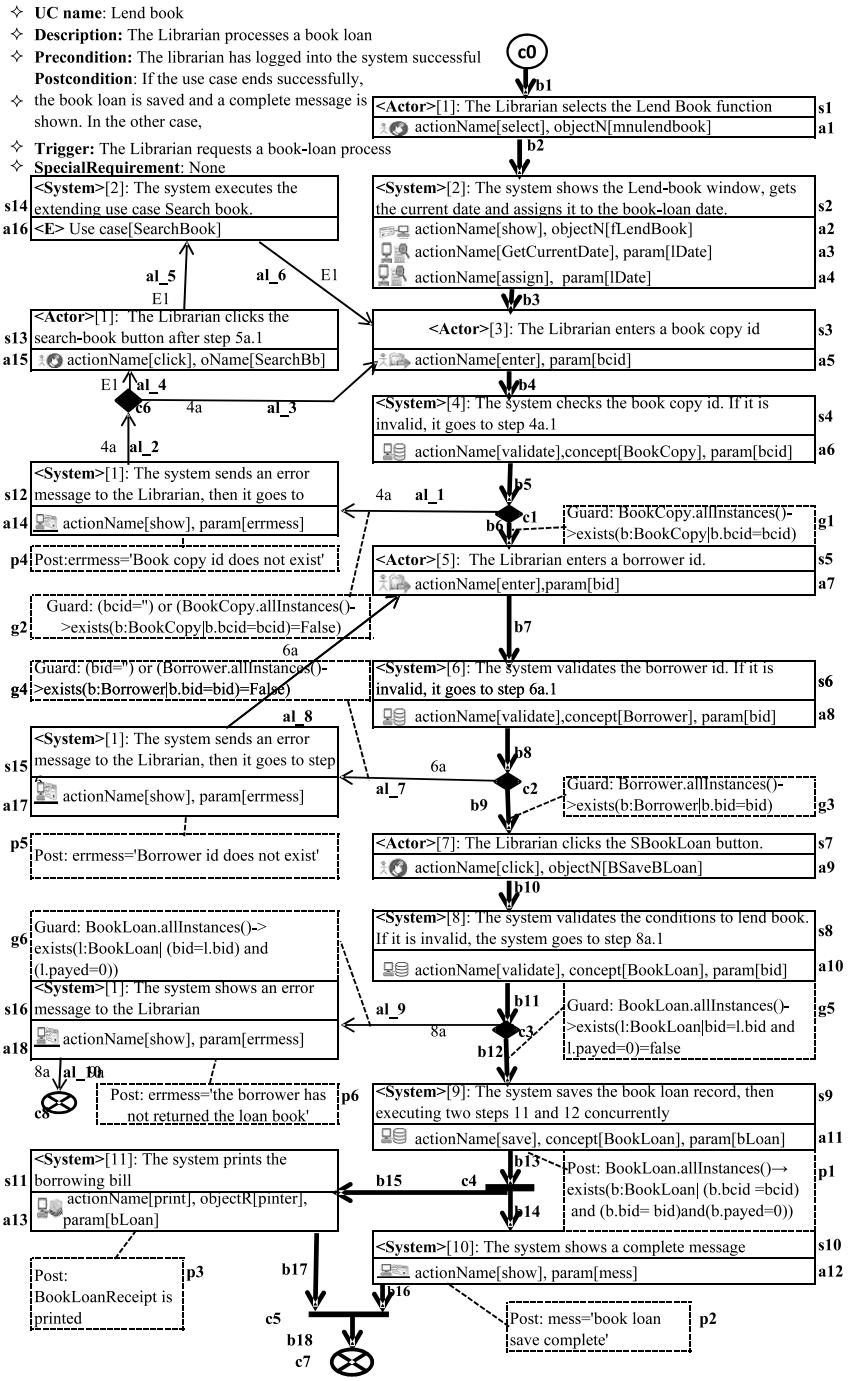


Figure 4: Representing the *Lend Book* use case as a USL model.

```
2 ->size()>=1
```

Rule 4. An `InitialNode` has one out-going `BasicFlowEdge` and does not have any in-coming `FlowEdges`:

```
1 (self.flowedge->select(t:FlowEdge|t.source.
      oclIsTypeOf (InitialNode))->size()=1)and(
      self.flowedge ->select (b:FlowEdge|(b.
      source.oclIsTypeOf(InitialNode)) and (b.
      oclIsTypeOf(BasicFlowEdge)))->size()=1)
      and (self.flowedge->select(t:FlowEdge| t
      .target.oclIsTypeOf(InitialNode))->size()
      =0)
```

Rule 5. A `FinalNode` has one in-coming `FlowEdge` and does not have any out-going `FlowEdge`:

```
1 self.uslnode->selectByType(FinalNode)->
      forAll (f:FinalNode|(self.flowedge->
      select(e:FlowEdge|e.target=f) ->size()
      =1) and (self.flowedge->select (e:
      FlowEdge|e.source=f)->size()=0))
```

Rule 6. A `DecisionNode` has one in-coming `FlowEdge` and at least two out-going `FlowEdges`:

```
(d:DecisionNode|(self.flowedge->select(e:
    FlowEdge|e.target=d)->size()=1)and (
    self.flowedge->select(e:FlowEdge|e.
    source=d)->size()>=2) )
```

Rule 7. A `ForkNode` has at least one in-coming `FlowEdge` and at least two out-going `FlowEdges`:

```
self.uslnode->selectByType(ForkNode)->forAll(
    f:ForkNode|(self.flowedge->select(e:
    FlowEdge|e.target=f)->size()>=1) and (
    self.flowedge->select(e:FlowEdge|e.source
    =f)->size()>=2) )
```

Rule 8. A `JoinNode` has at least two in-coming `FlowEdges` and one out-going `FlowEdge`.

```
self.uslnode->selectByType(JoinNode)->forAll
    (j:JoinNode|(self.flowedge->select(e:
    FlowEdge|e.target=j)->size()>=2)and(self.
    flowedge->select (e:FlowEdge|e.source=j)
    ->size()=1)).
```

Rule 9. A `SystemStep` or `ActorStep` has at least one in-coming `FlowEdge` and one out-going `FlowEdge`:

```
self.uslnode->selectByKind(FlowStep)->forAll(
    f:FlowStep|(self.flowedge->select(e:
    FlowEdge|e.target=f)->size()>=1)and(self.
    flowedge->select(e:FlowEdge|e.source=f)->
    size()=1))
```

Rule 10. A USL model is valid if the `FlowEdges` that connect the `USLNodes` of the model are valid, i.e., the type and label are correctly defined:

```
self.uslnode->forAll(n:USLNode|
if (n.oclIsTypeOf(InitialNode))then
 self.flowedge->select(b:FlowEdge|
 (b.source.oclIsTypeOf(USL::InitialNode)) and
     (b.oclIsTypeOf(USL::BasicFlowEdge)))->
     size()=1
else
 if (self.flowedge->selectByType(
     BasicFlowEdge)->select(b:BasicFlowEdge|b
     .target=n))->size()>=1 then
  if (n.oclIsTypeOf(DecisionNode))then
   self.flowedge->selectByType(BasicFlowEdge)
       ->select (b:BasicFlowEdge|b.source=n)
       ->size()=1
  else
   if (n.oclIsTypeOf(ForkNode)) then
    self.flowedge->select(f:FlowEdge|f.source
        =n)->forAll(b:FlowEdge|b.oclIsTypeOf(
        BasicFlowEdge))
   else
    if (n.oclIsTypeOf(JoinNode)) then
     (self.flowedge->select(f:FlowEdge|f.
         source=n)->forAll(b:FlowEdge|b.
         oclIsTypeOf(BasicFlowEdge)))and(self
         .flowedge->select(f:FlowEdge|f.
         target=n)->forAll (b:FlowEdge|b.
         oclIsTypeOf(BasicFlowEdge)))
    else
     if(n.oclIsKindOf(FlowStep)) then
      self.flowedge->select(f:FlowEdge|(f.
          source=n) and (f.oclIsTypeOf(
          BasicFlowEdge)))-> size() = 1
     else true
    endif
   endif
  endif
else ((self.flowedge->selectByType(
     BasicFlowEdge)->select(b:BasicFlowEdge|b.
     source=n))->size()=0) and if(n.
     oclIsTypeOf(FinalNode))then
   true else self.flowedge ->selectByType (
       AlternateFlowEdge) ->exists (f:
       AlternateFlowEdge|f.label=self.
       flowedge->selectByType(
       AlternateFlowEdge)->select(a:
       AlternateFlowEdge|a.target=n)->first()
       .label)endif)
endif
endif)
```

Rule 11. The `number` property of each `FlowStep` in a **Basic flow** is unique:

```
self.uslnode->selectByKind(FlowStep)->select(
    n:FlowStep|self.flowedge->selectByType(
    BasicFlowEdge)->exists(t:BasicFlowEdge|(t
    .source=n)or(t.target=n)))->forAll(n1:
    FlowStep, n2:FlowStep|n1.number=n2.number
     implies n1=n2).
```

Rule 12. The `number` property of each `FlowStep` in an **Alternate flow** is unique:

```
self.uslnode->selectByKind(FlowStep)->select
    (n:FlowStep|self.flowedge->selectByType(
    AlternateFlowEdge)->exists(t:
    AlternateFlowEdge|(t.source=n)or(t.target
    =n)))->forAll(n1:FlowStep,n2:FlowStep|(n1
    .number=n2.number)and(self.flowedge->
    selectByType(AlternateFlowEdge)->select(
    t1:AlternateFlowEdge|t1.target=n1)->first
    ().label=self.flowedge->selectByType(
    AlternateFlowEdge)->select(t2:
    AlternateFlowEdge|t2.target=n2)->first().
    label) implies n1=n2).
```

Example 4.1.5. Let us focus on the USL model as shown in Fig. 4:

- If we remove c0 from or add a new `InitialNode` to this model then it will violate Rule 1.
- If we remove both c7, c8 from the model then it will violate Rule 2.
- If the model only has `ControlNodes` then it will violate Rule 3.
- If `FlowEdge` b1 is not a `BasicFlowEdge` but an `AlternateFlowEdge` then the model will violate Rule 4.
- If we connect b17 to c8 and remove c7 then the model will violate Rule 5.
- If we add an `AlternateFlowEdge` to connect s4 to c6 then the model will violate Rules 6 and 9.
- If we remove b14, s11, b16, p3 then the model will violate Rules 7 and 8.

- If either `FlowEdge` b6 is not a `BasicFlowEdge` but an `AlternateFlowEdge` or the value of the `label` property of `AlternateFlowEdge` al_1 is not "4a" then the model will violate Rule 10.
- If the value of the `number` property of s2 is not 2 but 1 then the model will violate Rule 11.
- If the value of the `number` property of s14 is not 2 but 1 then the model will violate Rule 12.

4.2 The USL concrete syntax

In order to help the user to easily create USL models, we propose a concrete syntax for USL with the graphical notations as shown in Table 3. We have implemented this syntax in a visual editor for USL modelling. A detailed explanation of this tool will be presented in Sect. 6.

4.3 Formal semantics of USL

We formally define a `USL model` as follows. Here, we consider a USL model as a graph consisting of nodes and edges. A node represents either a step or a control action performed by the system. Further, we will take into account the fact that the underlying use case references the domain concepts captured in a UML class diagram.

Definition 1. A `USL Model` of a use case is the tuple $D = \langle D_C, A, E, C\rangle$ such that:

- D_C is a class diagram to represent the underlying domain;
- A is the set of `USLNodes`;
- E is the set of `FlowEdges`;
- $C = G \cup C_{preUC} \cup C_{postUC} \cup C_{preA} \cup C_{postA}$ is the set of `Constraints`,

where:

- $A = A_{cNode} \cup A_f$;
- $A_{cNode} = N_I \cup N_F \cup N_d \cup N_j \cup N_f$, where $N_I = \{a \mid a \in A, \text{InitialNode}(a)\}$ $N_F = \{a \mid a \in A, \text{FinalNode}(a)\}$, $N_d = \{a \mid a \in A, \text{DecisionNode}(a)\}$, $N_j = \{a \mid a \in A, \text{JoinNode}(a)\}$, $N_f = \{a \mid a \in A, \text{ForkNode}(a)\}$;
- $|N_I| = 1$; $|N_F| \geq 1$;
- $A_f = A_a \cup A_s$, where $A_f = \{a \mid a \in A, \text{FlowStep}(a)\}$, $A_a = \{a \mid a \in A, \text{ActorStep}(a)\}$, $A_s = \{a \mid a \in A, \text{SystemStep}(a)\}$;
- $|A_s| \geq 1$; $\forall s \in A_f.|\texttt{actions}(s)| \geq 1$;
- $E = E_b \cup E_a$ and $E_b \cap E_a = \emptyset$, where $E_b = \{e \mid e \in E, \text{BasicFlowEdge}(e)\}$, $E_a = \{e \mid e \in E, \text{AlternateFlowEdge}(e)\}$.

Table 3: The graphical notations of USL

Concepts	Presentation	Notation
DescriptionInfor	A borderless text box that properties are listed in the text box	
InitialNode	An unfilled circle	
FinalNode	A circle with a crosshairs symbol	
DecisionNode	A filled diamond with one in-coming arrowed line and at least two out-going arrowed lines	
ForkNode	A solid line segment with one in-coming arrowed line and at least two out-going arrowed lines	
JoinNode	A solid line segment with at least two in-coming arrowed lines and one out-going arrowed line	
BasicEdge	A thick arrowed line	
AlternateEdge	A labelled, thin arrowed line (the label is the name of the flow)	label
ActorStep	A labeled, 2-part rectangle. The first part contains the label **<Actor>** and two properties `numberStep` and `description` of the `ActorStep`. The second part contains the `ActorActions` of the `ActorStep`	<Actor>
SystemStep	A labelled, 2-part rectangle. The first part contains the label **<System>** and two properties `numberStep` and `description` of the `SystemStep`. The second part contains the `SystemActions` of the `SystemStep`	<System>
Action	Information of a `Action` are presented by textual form in the second part of `FlowSteps`	

Example 4.1.5. The USL model as shown in Fig. 4 contains the following elements: $N_I = \{c0\}$; $N_F = \{c7, c8\}$; $A_{cNode} = \{c0, \ldots, c8\}$; $A_a = \{s1, s3, s5, s7, s13\}$; $A_s = \{s2, s4, s6, s8, \ldots, s12, s14, s15, s16\}$; $E_b = \{b1, \ldots, b17\}$; $E_a = \{al\_1, \ldots, al\_10\}$; $G = \{g1, \ldots, g6\}$; $C_{preUC} = \emptyset$; $C_{postUC} = \emptyset$; $C_{preA} = \emptyset$; and $C_{postA} = \{p1, \ldots, p6\}$. D_C corresponds to the conceptual model shown in the part (b) of Fig. 1. There are sixteen constraints for guard conditions and pre- and postconditions, e.g., the postcondition $p1$ of `Action` $a11$ is expressed by the following OCL contraint:

```
BookLoan.allInstances()->exists(b:BookLoan|
(b.bcid = bcid) and (b.bid = bid) and
(b.payed=0)).
```

We use LTS [20] to formally define the operational semantics of USL. Conceptually, the execution of a USL model is modelled by an LTS, whose transitions are caused by the execution of use case actions, and whose states are defined by variable assignments during the execution. We define the LTS of a USL model recursively from the basic USL concepts. The semantics of these concepts are defined as summarized in Table 4. Definition 2 formalizes the notion of the LTS of the USL model.

Definition 2. Given a USL model $D = \langle D_C, A, E, C\rangle$, an `LTS` that results from the execution of D is the tuple $\langle \Sigma(\mathcal{V}), \mathbb{P}(\mathcal{G} \times \mathcal{A} \times \mathcal{P}), \mathcal{T}, \alpha_{init}, \mathcal{F}\rangle$ such that:

Table 4: LTS-based semantics of the basic USL concepts

USL concepts	Notation	LTS-based semantics
Step	$a_1 \rightarrow \cdots \rightarrow a_n$	$\alpha \xrightarrow{g_1\|a_1\|r_1} \alpha_1 \cdots \xrightarrow{g_n\|a_n\|r_n} \alpha_n$ where $g_i = \texttt{preA}(a_i)$, $r_i = \texttt{postA}(a_i)$ $(\forall i = 2, \ldots, n)$.
Flow edge	$n_1 \rightarrow n_2$	$\alpha \xrightarrow{g_1\|a_1\|r_1} \alpha_1 \xrightarrow{g_2\|a_2\|r_2} \alpha_2$ where $a_2 = \texttt{firstAct}(n_2)$, $g_2 = \texttt{guardE}(n1, n2) \wedge \texttt{preA}(a_2)$ $r_2 = \texttt{postA}(a_2)$.
Decision node	$n_d \rightarrow \diamond \xrightarrow{c} n_1, \ldots, n_m$	$\alpha \xrightarrow{g_d\|a_d\|r_d} \alpha_d \xrightarrow{g_c\|c\|\texttt{true}} \alpha_c \xrightarrow{g_a\|a\|r_a} \alpha'$ where $a_d = \texttt{lastAct}(n_d)$; $g_c = \texttt{guardE}(n_d, c)$, $a = \texttt{firstAct}(n)$, $g_a = \texttt{guardE}(c, n) \wedge \texttt{preA}(a)$ s.t $n \in \{n_1, \ldots, n_m\}$.
Fork node	$n_f \rightarrow n_1, \ldots, n_m$ (c)	$\alpha \xrightarrow{g_f\|a_f\|r_f} \alpha_f \xrightarrow{g_c\|c\|\texttt{true}} \alpha_c \xrightarrow{g_1\|a_1\|r_1} \alpha'_1, \ldots, \xrightarrow{g_m\|a_m\|r_m} \alpha'_m$ where $a_f = \texttt{lastAct}(n_f)$; $g_c = \texttt{guardE}(n_f, c)$, $a_i = \texttt{firstAct}(n_i)$, $g_i = \texttt{preA}(a_i)$, $r_i = \texttt{postA}(a_i)$ $(\forall i = 1, \ldots, m)$.
Join node	$n_1, \ldots, n_m \rightarrow n_j$ (c)	$\alpha_1, \ldots, \alpha_m \xrightarrow{g_1\|a_1\|r_1, \ldots, g_m\|a_m\|r_m} \alpha_c \xrightarrow{g_w\|c\|\texttt{true}} \alpha_j \xrightarrow{g_j\|a_j\|r_j} \alpha$ where $a_i = \texttt{lastAct}(n_i)$, $(\forall i = 1, \ldots, m)$; $g_w = \left(\bigwedge_{(e \in D.E, \texttt{target}(e)=c)} \texttt{isCompleted}(e) \wedge \texttt{guardE}(e)\right)$; $a_j = \texttt{firstAct}(n_j)$, $g_j = \texttt{guardE}(c, n_j) \wedge \texttt{preA}(a_j)$, $r_j = \texttt{postA}(a_j)$.
Initial node	$c \rightarrow n$	$\alpha \xrightarrow{r_u\|c\|\texttt{true}} \alpha_i \xrightarrow{g\|a\|r} \alpha'$ where $\alpha = \alpha_{init}$, $r_u = \texttt{preC}(D)$; $a = \texttt{firstAct}(n)$, $g = \texttt{guardE}(c, n) \wedge \texttt{preA}(a)$, $r = \texttt{postA}(a)$.
Flow final node	$n \rightarrow \otimes$ (c)	$\alpha \xrightarrow{g\|a\|r} \alpha_f \xrightarrow{g_c\|c\|r_f} \alpha'$ where $\alpha' \in \mathcal{F}$, $r_f = \texttt{postC}(D, c)$, $g_c = \texttt{guardE(n, c)}$; $a = \texttt{lastAct}(n)$,
USL model with include action	$n_1 \rightarrow n \rightarrow n_2$, $n \rightarrow D_I$ $\equiv$ $n_1 \rightarrow n_{I_1} \rightarrow \cdots \rightarrow n_{I_m} \rightarrow n_2$	where $n \in A_s$, $\|\texttt{actions}(n)\| = 1$, $\texttt{SystemInclude}(a)$ $(a \in \texttt{actions}(n))$, $t = (\alpha, (g_a\|a\|r_a), \alpha')$; $n_{I_1}, \ldots, n_{I_m} \in D_I.A$, $g_a = \texttt{guardE}(n1, n) \wedge \texttt{preA}(a) \wedge \texttt{preC}(D_I)$, $r_a = \texttt{postC}(D_I)$
USL model with extend action	$n_1 \rightarrow n \rightarrow n_2$, $n \rightarrow D_X$ $\equiv$ $n_1 \rightarrow n_{X_1} \rightarrow \cdots \rightarrow n_{X_m} \rightarrow n_2$	where $n \in A_s$, $\|\texttt{actions}(n)\| = 1$, $\texttt{SystemExtend}(a)$ $(a \in \texttt{actions}(n))$, $t = (\alpha, (g_a\|a\|r_a), \alpha')$; $n_{X_1}, \ldots, n_{X_m} \in D_X.A$, $g_a = \texttt{preA}(a) \wedge \texttt{preC}(D_X)$, $g_a = \texttt{guardE}(n1, n) \wedge \texttt{preA}(a) \wedge \texttt{preC}(D_X)$, $r_a = \texttt{postC}(D_X)$
Legend	a a action in a step; s a step; D_u use case; α a state	

- $\mathcal{V}$ is a finite set of variables whose types include the basic types and the classes of the D_C;
- $\Sigma(\mathcal{V})$ is the set of states (α), each of which is a set of value assignments to a subset of variables in $\mathcal{V}$;
- $\mathcal{P} \subseteq C_{postA} \cup C_{postUC}$ is the set of constraints as the postconditions of D;
- $\mathcal{A} = A_{cNode} \cup A_{act}$ is the set of actions;
- $\mathcal{G} \subseteq G \cup C_{preUC} \cup C_{preA}$ is the set of guard conditions of the transitions;
- $\mathcal{T} \subseteq \Sigma(\mathcal{V}) \times \mathbb{P}(\mathcal{G} \times \mathcal{A} \times \mathcal{P}) \times \Sigma(\mathcal{V})$ is the transition relation defined as follows: A transition $t = (\alpha, (g, a, d), \alpha') \in \mathcal{T}$, written as $\alpha \xrightarrow{g|a|r} \alpha'$, where $a \in \mathcal{A}$ is the action that causes t, $g = \texttt{defGuard}(a) \in \mathcal{G}$ is the guard condition to execute a, $r \in \mathcal{P}$ is the postcondition of a, and $\alpha, \alpha' \in \Sigma(\mathcal{V})$ are the pre- and post-states of t (resp.) such that α' satisfies r;
- $\alpha_{init} \in \Sigma(\mathcal{V})$ is the initial state;
- $\mathcal{F} \subset \Sigma(\mathcal{V})$ is the set of final states,

where:

- $A_{act} = \bigcup_{s \in A_f} \texttt{actions}(s)$;
- `defGuard` is defined as follows (summarized from Table 4).

$$= \begin{cases} \texttt{preC}(D), \texttt{ifInitialNode}(a) \\ \texttt{guardE}(e)(e \in D.E, \texttt{target}(e) = a), \texttt{ifDecisionNode}(a) \\ \vee\ \texttt{ForkNode}(a) \vee\ \texttt{FinalNode}(a) \\ \bigwedge_{(e \in D.E, \texttt{target}(e)=a)} \texttt{isCompleted}(e) \wedge \texttt{guardE}(e), \\ \text{if } \texttt{JoinNode}(a) \\ \texttt{preC}(D_I) \wedge \texttt{preA}(a) \wedge \texttt{guardE}(e)(e \in D.E, \texttt{target}(e) = a), \\ \text{if } \texttt{SystemInclude}(a) \\ \texttt{preC}(D_X) \wedge \texttt{preA}(a) \wedge \texttt{guardE}(e)(e \in D.E, \texttt{target}(e) = a), \\ \text{if } \texttt{SystemExtend}(a) \\ \texttt{preA}(a) \wedge \texttt{guardE}(e)(s \in A_f, \texttt{target}(e) = s), \\ \texttt{if}((a \in A_{act}) \wedge \ \ (a = \texttt{firstAct}(s)) \\ \texttt{preA}(a)(s \in A_f, a \in actions(s)), \texttt{if}otherwise \end{cases}$$

Brrower

bid	name	bday
123	Joney	6/2/90
124	Mary	2/3/91

BookCopy

bcid	bkid	state
001	N01	'normal'
002	N01	'normal'

BookLoan

lbid	bcid	bid	lid	ldate	payed
1	002	123	110	2/3/17	0

Librarian

lid	name	bday	pword
110	Davi	5/8/80	12334
111	Bob	9/3/91	12344

Figure 5: A snapshot of the *Lend Book* use case.

Example 4.3.1. We assume that the snapshot shown in Fig. 5 is captured when the USL model as shown in Fig. 4 is executed at Step a8. We have the following value assignments: $(bcid, \text{“001”}) \equiv bcid = \text{“001”}$, $(lid, \text{“110”}) \equiv lId = \text{“100”}$, $(ldate, \text{“25/8/17”}) \equiv ldate = \text{“25/8/17”}$, $(bid, \text{“1234”}) \equiv bcid = \text{“1234”}$. The objects of the snapshot are as follow: BookCopy:“001”, BookCopy:“002”, Borrower:“123”, Borrower:“124”, Librarian:“100”, Librarian:“111”, BookLoan:“1”. Then, we have $\alpha_{a8} = \{(bcid, \text{“001”}), (ldate, \text{“001”}), (lid, \text{“110”}), (bid, \text{“124”}), (bLoan, (\text{“2”}, \text{“001”}, \text{“124”}, \text{“110”}, \text{“25/8/17”}, 0))$, BookCopy:“001”, BookCopy:“002”, Borrower:“123”, Borrower:“124”, Librarian:“100”, Librarian:“111”, BookLoan:“1”}.

Certain use case actions are concurrent actions, whose executions cause concurrent transitions between states. The next two definitions define precisely what this means.

Definition 3. Given a current state α of an LTS L of a USL model D and a transition $t = \alpha \xrightarrow{g|a|r} \alpha' \in L.\mathcal{T}$, we define the following terms:

- $\texttt{preT}(t) = \alpha$, $\texttt{postT}(t) = \alpha'$, $\texttt{guard}(t) = g$, $\texttt{postC}(t) = r$, and $\texttt{act}(t) = a$.
- $\texttt{eval}(g)$ is the evaluation of Constraint g.
- $\texttt{reachable}(\alpha) = \{t \mid \texttt{preT}(t) = \alpha\}$ is the set of transitions that start from α.
- $\texttt{firable}(\alpha) = \{t \in \texttt{reachable}(\alpha), \texttt{eval}(\texttt{guard}(t)) = \texttt{true}\}$ is the set of transitions that can be fired from α.

Example 4.3.2. When the USL model as shown in Fig. 4 executes at action $a11$, we have $\alpha_{a11} = \{(bcid, \text{“001”}), (lDate, \text{“001”}), (lid, \text{“110”}), (bid, \text{“124”}), (bLoan, (\text{“2”}, \text{“001”}, \text{“124”}, \text{“110”}, \text{“25/8/17”}, 0))$, BookCopy:“001”, BookCopy:“002”,Borrower:“123”, Borrower:“124”, Librarian:“100”, Librarian:“111”, BookLoan:“1”, BookLoan:“2”}.

Transition $t_{a11,c4} = \alpha_{a11} \xrightarrow{true|c4|true} \alpha_{c4}$. $\texttt{reachable}(\alpha_{a11}) = \{t_{a11,c4}\}$ and $\texttt{firable}(\alpha_{a11}) = \{t_{a11,c4}\}$.

Definition 4. Given a current state α of an LTS L of a USL model D, a concurrent transition $\tau \in L.\mathcal{T}$ is a set of transitions $t_1, t_2, \ldots, t_n \in \texttt{firable}(\alpha)$.

Example 4.3.3. When the USL model as shown in Fig. 4 executes at Step $c4$, we have two transitions $t_{c4,a12} = \alpha_{c4} \xrightarrow{true|a12|p2} \alpha_{a12}$ and $t_{c4,a13} = \alpha_{c4} \xrightarrow{true|a13|p3} \alpha_{a13}$, $\texttt{reachable}(\alpha_{c4}) = \{t_{c4,a12}, t_{c4,a13}\}$ and $\texttt{firable}(\alpha_{c4}) = \{t_{c4,a12}, t_{c4,a13}\}$. Hence, $\{t_{c4,12}, t_{c4,13}\}$ is a concurrent transition and α_{a12}, α_{a13} satisfy p2, p3, respectively.

Within our approach the LTS of a USL model may contain both concurrent and non-concurrent transitions. We next define the semantics of a use case scenario.

Definition 5. Given a use case scenario of a USL model D that consists of the following sequence of actions $(a_0, \ldots, a_{n-1})$. The execution of this scenario is realized as a path in the LTS L of D: $p = \alpha_0 \xrightarrow{t_0} \alpha_1 \xrightarrow{t_1} \cdots \xrightarrow{t_{n-1}} \alpha_n$, where $t_i = \alpha_i \xrightarrow{g_i|a_i|r_i} \alpha_{i+1}$ $(\forall i = 0, \ldots, n-1)$, $\alpha_0 = L.\alpha_{init}$, $\alpha_n \in L.\mathcal{F}$, and $t_i \in L.\mathcal{T}$.

Example 4.3.4. When the USL model as shown in Fig.4 executes at Step α_{a11} as mentioned above and eval(g1), eval(g3), and eval(g5) are true, then the use case scenario is as follows:

$$p = \alpha_{init} \xrightarrow{true|a1|true} \alpha_{a1} \xrightarrow{true|a2|true} \alpha_{a2} \xrightarrow{true|a3|true} \alpha_{a3} \xrightarrow{true|a4|true} \alpha_{a4} \xrightarrow{true|a5|true} \alpha_{a5} \xrightarrow{true|a6|true} \alpha_{a6} \xrightarrow{true|c1|true} \alpha_{c1} \xrightarrow{g1|a7|true} \alpha_{a7} \xrightarrow{true|a8|true} \alpha_{a8} \xrightarrow{true|c2|true} \alpha_{c2} \xrightarrow{g3|a9|true} \alpha_{a9} \xrightarrow{true|a10|true} \alpha_{a10} \xrightarrow{g5|a11|true} \alpha_{a11} \xrightarrow{true|c4|true} \alpha_{c4} \xrightarrow{\{true|a12|p2, true|a13|p3\}} \alpha_{a12-a13} \xrightarrow{true|c5|true} \alpha_{c5} \xrightarrow{true|c7|true} \alpha_{c7}\ (\alpha_{c7} \in \mathcal{F}).$$

5 Transforming USL models to other software artifacts

This section explains how USL models can be transformed to software artifacts including test cases, structural and behavioral models, and textual template-based use case descriptions (TUCDs). We particularly focus on the last transformation (to obtain TUCDs) and show how the transformation could be realized.

5.1 Generating test cases

A test scenario is used to create a set of test cases [27]. A test case results from combining a test scenario with some test data. According to the use case-driven testing approach [27], a use case scenario identifies one test scenario (a use case description consists of one or more use case scenarios). The constraints of a use case scenario help identifying the test data of the corresponding test scenario.

The model-based testing (MBT) method [28] presents a specific technique for automatically generating test cases from a use case model. Specifically, the control flows of a use case model are used to generate the use case scenarios. For example, Linzhang [29] first presents a technique to represent the control flows using UML activity diagram. He then proposes an algorithm to traverse all the possible basic paths of the activity diagram to generate the test scenarios.

Two other works [30, 31] focus on the problem of automatically generating test data from the test scenario constraints, written in OCL. They develop OCL constraint solvers for this task.

Since our USL captures the necessary information elements of the use case description, we argue that USL models can also be used as an input to generate test cases. More specifically, USL has meta-concepts for representing the different control nodes of the UML activity diagram. Further, the `Constraint` meta-concept of USL captures the different types of constraints that are needed to generate test data.

5.2 Generating structural and behavioural models

In the requirement analysis activity, the behaviours described in a use case description are analysed in order to create other structural and behavioural models. The target models are often represented using UML diagrams, including activity diagram, class diagram, collaboration diagram, and sequence diagram.

D. Savić *et al.* [16] and M. Smialek *et al.* [17] propose a specific method for the above. In particular, they first use different types of actions to precisely model the use case behaviours. They then present a model transformation technique that automatically transforms the behaviours and other relevant model elements into a class diagram. Examples of these elements that are discussed in [28] include sender and receiver objects, messages, and parameters.

Our USL specification was inspired by this work. Specifically, we use `Action` meta-concept to represent use case behaviours and the relevant model elements discussed above. Regarding to behavioural modelling, a USL model can be used as input to generate activity and sequence diagrams. The reason is because USL represents all the control nodes of UML activity diagram. For example, a specific technique for generating sequence diagram is presented in [12].

5.3 Generating TUCDs

According to [32, 33, 34], textual template-based use case descriptions (TUCDs) [1, 22, 8] enable the customer to positively participate in requirement analysis, to identify and resolve conflicts in the requirement drafts, and to ensure that it is consistent with their intention. Table 1 shown earlier is an example of such a template.

In order to automatically generate a TUCD from a USL model, we develop a transformation USL2TUCD using the model-to-text transformation language Acceleo [18]. The transformation USL2TUCD is shown in Listing 1. We illustrate this transformation using the USL model of the use case named `Withdrawal` (shown in Fig. 9). The output TUCD is a text file named `Withdrawal.txt` that is shown in Fig. 10.

Briefly, the USL2TUCD transformation uses five queries to extract information from the input USL model (`uc`). The first query is `getBasicFlow(uc)` at line 19. It is used is to find all the `BasicFlowSteps` in `uc`. The second query is `getDecisionNode (uc)` at line 25. It is used to get all the `DecisionNodes` in `uc`. The third query is `getPreAlternateFlowLabel(uc,d)` at line 27. It is used to get the label of the in-coming `AlternateFlowEdge` of some `DecisionNode d` in `uc`. It returns empty if no such `AlternateFlowEdges` exist. The fourth query is `getAFEdges(uc,d)` at lines 29 and 37. This query is used to get the out-going `AlternateFlowEdges` from a `DecisionNode d` in `uc`. The fifth query is `getAlternateFlow(uc, l)` at lines 30 and 39. This query is used to find the `FlowSteps` in the `AlternateFlow` in `uc` that is labeled `l`.

The definitions of all five queries are written in another transformation named `libraryUCD`. The transformation is shown in Listing 2.

Listing 1: The USL2TUCD transformation

```
1  [module GenUCDescription('http://eclipse
       .USLModel/USL')]
2  [import org::eclipse::acceleo::module::
       sample::service::libraryUCD]
3  [template public generateElement(uc:
       UseCase)]
4  [comment @main/]
5  [file (uc.descriptioninfor->r at(0).
       useCaseName.concat('.txt'), false, '
       UTF-8')]
6  [let d:DescriptionInfor =uc.
       descriptioninfor-> at(0)]
7  ----------------------------------
8  UC name: [d.useCaseName/]
9  Description: [d.description /]
10 Actor: [for(a:String|d.actor)] a, [/for]

11 Level abstract: [d.levelAbstract /]
12 Precondition: [d.preCondition /]
13 Postcondition: [d.postCondition /]
14 SpecialRequirement: [d.
       specialRequirement/]
15 [/let]
16 ----------------------------------
17 BasicFlow
18 ----------------------------------
19 [let Bsteps:OrderedSet(FlowStep)=
       getBasicFlow(uc)]
20 [for(s:FlowStep|Bsteps)]
21   [s.number/].  [s.description/]
22 [/for] [/let]
```

```
----------------------------------
AlternateFlow
[let dList:OrderedSet(DecisionNode)=
    getDecisionNode(uc)]
[for(d:DecisionNode|dList)]
 [let preALabel:String=
     getPreAlternatFlowLabel(uc, d)]
 [if(preALabel='')]
  [for(af:AlternateFlowEdge|getAFEdges(
      uc, d))]
   [let Asteps:OrderedSet(FlowStep)=
       getAlternateFlow(uc, af.label)]
   [af.label/]. [af.description]
   [for(s:FlowStep|Asteps)]
     [s.number/]. [s.description/]
   [/for] [/let]
  [/for]
 [else]
  [for(af:AlternateFlowEdge|getAFEdges(
      uc, d))]
   [if(af.label <>preALabel)]
    [let Asteps:OrderedSet(FlowStep)=
        getAlternateFlow(uc, af.label)]
    [af.label/]. [af.description/]
    [for(s:FlowStep|Asteps)]
      [s.number/]. [s.description/]
    [/for] [/let]
   [/if]
  [/for]
 [/if] [/let]
[/for] [/let]
----------------------------------
[/file]
[/template]
```

Listing 2: The `libraryUCD` transformation

```
[comment encoding = UTF-8 /]
[module libraryUCD('http://eclipse.
    USLModel/USL')]
[query public getBasicFlow(uc:UseCase):
    OrderedSet(FlowStep)=uc.uslnode->
    select (n:USLNode|uc.flowedge->
    selectByType (BasicFlowEdge)->exists(
    b:BasicFlowEdge|(n=b.source) or (n=b.
    target)))->selectByKind(FlowStep)/]

[query public getAlternateFlow(uc:
    UseCase,l:String): OrderedSet(
    FlowStep)=uc.uslnode->select(n:
    USLNode|uc.flowedge->selectByType(
    AlternateFlowEdge)->select(a:
    AlternateFlowEdge|a.label=l)->exists(
    f:AlternateFlowEdge|(f.target=n)or(f.
    source=n)))->selectByKind(FlowStep)/]

[query public getAFEdges(uc:UseCase, d:
    DecisionNode): OrderedSet(
    AlternateFlowEdge) =uc.flowedge->
    select(f:FlowEdge|(f.source=d)and(f.
    oclIsTypeOf(AlternateFlowEdge)))->
    selectByType (AlternateFlowEdge) /]

[query public getDecisionNode(uc:UseCase
    ): OrderedSet(DecisionNode)=uc.
    uslnode->selectByType(DecisionNode)
    /]

[query public getPreAlternatFlowLabel (
    uc:UseCase,d:DecisionNode):String=
 if uc.flowedge->selectByType(
     AlternateFlowEdge)->select(f:
     AlternateFlowEdge| f.target=d)->size
     ()>0 then
  uc.flowedge->selectByType(
      AlternateFlowEdge)->select(f:
      AlternateFlowEdge| f.target=d)->at
      (0).label
 else
   ''
 endif /]
```

6 Tool support and evaluation

In this section, we first describe a USL tool that we have developed for visually creating USL models. After that, we explain two case studies for USL. We conclude this section with an evaluation of USL.

6.1 Tool support

We developed a support tool for our approach as illustrated in Fig. 6. This USL tool provides three main functions. The first function (displayed on the left of the figure) is called the "Loading function". It is responsible for loading the use cases and domain concepts of a system from a UML use case diagram and a class diagram. The second function (shown on the right of the figure) is called "USL Editor". It is used to create the USL models for the loaded use cases. This editor has a user-friendly GUI. The third function is called "Generating Artifacts". It automatically generates other software artifacts.

In our tool, the "Loading function" was developed using a Java project. The "USL Editor" was implemented using an EMF project and an GMF project within the Eclipse tool [26]. Specifically, the EMF project is to build the abstract syntax of USL and the GMF project is to build the concrete syntax and to implement the OCL constraint rules on the metamodel. The "Generating artifacts" function was written using model transformation languages, such as M2T and M2M [18]. To illustrate, Fig. 8 shows a USL model for the use case *Session*, that is created by the "USL Editor". Figure 10 shows a TUCD text file that is automatically generated by a transformation that was specified earlier in Listings 1 and 2. This transformation was written in the Acceleo M2T language.

Note that when working with a generation relationship between use cases, the modeler needs to create USL models only for the specific use cases rather than for the abstract ones.

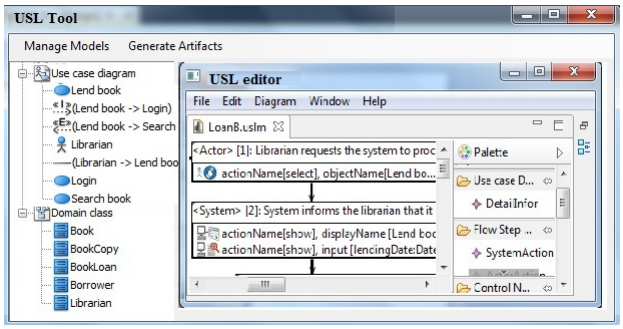


Figure 6: The USL tool.

6.2 Case study

In order to demonstrate the applicability of our method, we chose another system case study named *ATM*, which is described in Bjork [35]. The system includes three actors, seven specific use cases, one abstract use case, and two use case relationships. Figure 7 shows the use cases of the *ATM* system. Figure 8 and Fig. 9 show two USL models corresponding to these two use cases: *Session* and *Withdrawal*. Figure 10 and Fig. 11 show two TUCD text files that are generated from these two USL models, by applying the function "Generating Artifact". These files are the use case descriptions of the two corresponding use cases.

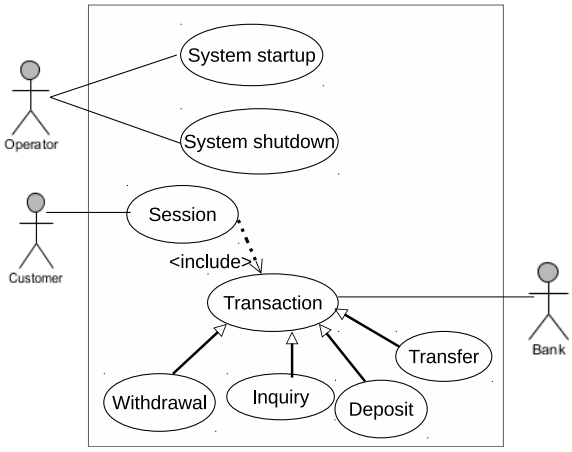


Figure 7: The use case diagram of the `ATM` system.

6.3 Language evaluation

This section presents our evaluation of USL's expressiveness, compared to five languages: RUCM [3], UC-B [10], MBD-L[1] [4], SiLabReq [16] and RSL [17]. We use the following four sub-criteria of expressiveness:

C1. Template-based representation of use case descriptions
C2. Control flow-based representation of use case behaviour
C3. Action specification
C4. Use case constraint representation

[1] 'MBD' stands for the author's names, 'L' for language.

Table 5 lists the evaluation results for the above criteria. In the table, we use three letters 'F', 'I', 'N' to denote the specification method that is used for each language: 'F' denotes *formal specification method*, 'I' denotes *informal specification method* and 'N' denotes that the specification method is not discussed.

Table 5: Expressiveness comparison between use case specification languages

Use case information	RUCM [3]	UC-B [10]	MBD-L [4]	SelabReq [16]	RSL [17]	USL
(c1) Overview elements	I	N	F	N	N	F
(c1) Flows of use case	I	I	F	N	N	F
(c1) Use case scenarios	N	N	N	F	F	N
(c2) Control flows	I	N	F	N	F	F
(c2) Concurrent actions	N	N	N	N	N	F
(c3) Action types	I	N	F	F	F	F
(c4) Use case scenario's pre- and postcondition	I	F	F	N	I	F
(c4) Guard conditions	I	F	F	N	I	F
(c4) Action's pre- and postcondition	N	F	N	N	N	F

We will discuss in detail the results shown in the table in the first five subsections that follow. In the last subsection, we discuss the possibility of applying USL in practice.

6.3.1 Template-based representation of use case descriptions

As discussed in Sect. 4, USL enables us to capture all the information elements of the use case description template shown in Table 1. In particular, the elements of overview information are described by the properties of the `DescriptionInfo` object in the model. The steps in a basic flow are represented by `FlowSteps` (including `ActorStep` and `SystemStep`) and are connected by `BasicFlowEdges` and `ControlNodes`. Similarly, steps in an alternate flow are represented by `FlowSteps` (including `ActorStep` and `SystemStep`) and are connected by `ControlNodes` and `AlternateFlowEdges`. USL represents this template precisely using the corresponding USL meta-concepts. Briefly, we draw the following conclusions from Table 5:

- USL is more expressive and more precise than three other languages, namely UC-B, SilabReq, and RSL.
- USL is more precise than RUCM.
- USL is as expressive and precise as MBD-L.

Specifically, our USL is more expressive than UC-B, SilabReq, and RSL because of the following reasons. First,

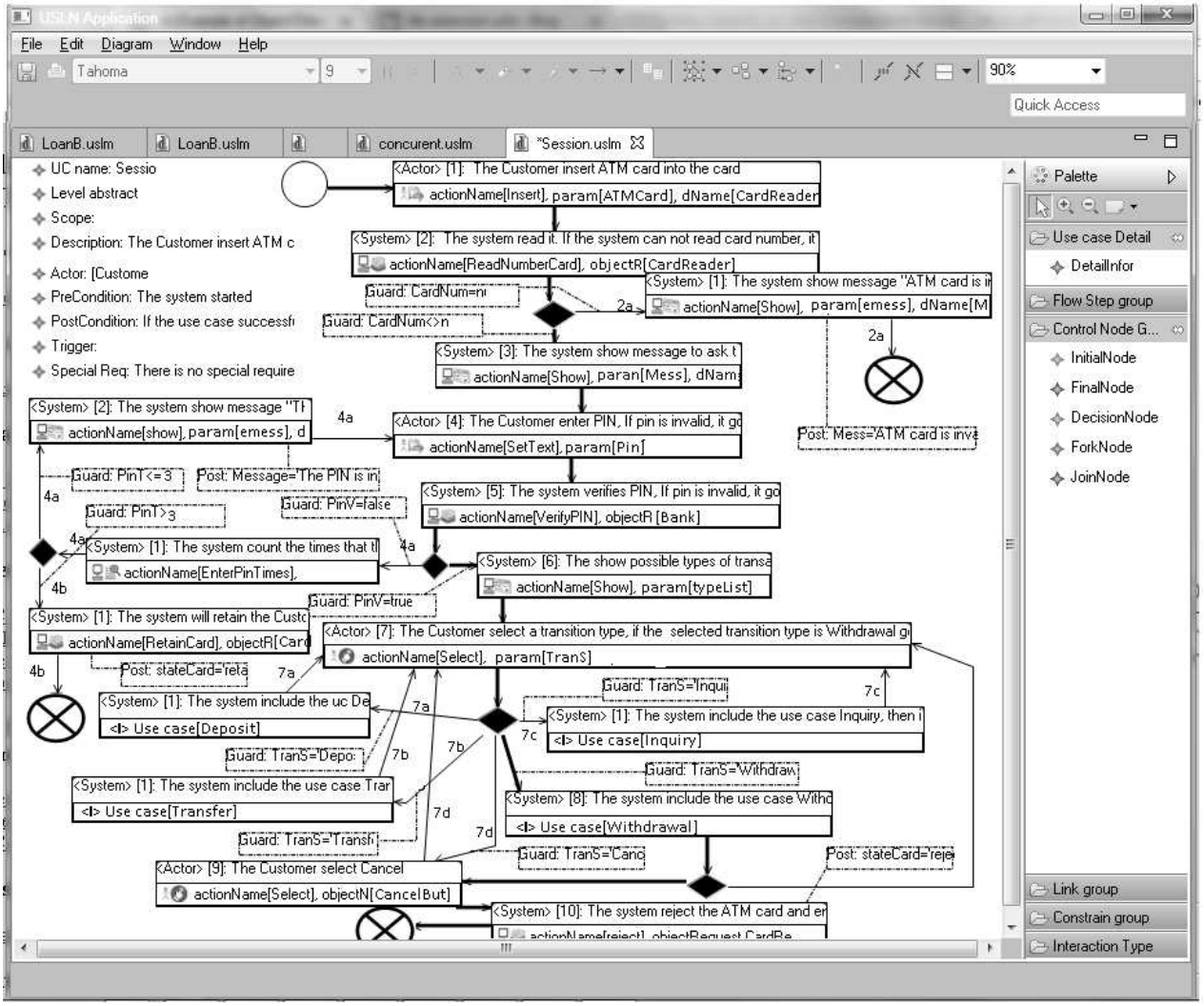


Figure 8: Modelling use case Session in the USL Editor tool.

the use case information elements that are captured in USL are more formal than what are represented in UC-B. UC-B provides a GUI for informally describing use case scenarios. Second, UC-B only represents the steps of a use case scenario and the trigger of a use case. SilabReq and RSL only capture flows corresponding to use case scenarios. With USL, we can express more use case information, such as the pre- and postcondition of an action.

On the other hand, USL captures information elements as expressively as RUCM. The RUCM method proposes a Restricted Use Case Modeling (RUCM) language, using a set of keywords and restricted description rules. Specifications in USL are more formal than those in RUCM, because RUCM's specifications are expressed in natural language.

In comparison with MBD-L, USL lacks concepts for specifying sub-flows. However, as discussed in Sect. 2, use cases containing sub-flows can be smoothed so that they are suitable for modelling in USL. On the other hand, MBD-L is only specified with the abstract syntax. Unlike USL, it does not contain a concrete syntax and a formal semantic.

6.3.2 Control flow representation for use case behaviour

Our USL language is built on UML activity diagram. A USL model includes `USLNodes` (corresponding to `Nodes` in UML activity diagram) and `FlowEdges` (corresponding to `Edges` in UML activity diagram) to specify control flows which pass through steps in the use case's flows. USL captures the different control flow types of UML activity diagram (such as sequence, branch, loop, and concurrence flow). In addition, USL also specifies steps with a limited number of iterations. For example, in the use case `Session` in SubSect. 6.2, Step 4 executes a maximum of three times. Briefly, we draw the following conclusions from Table 5:

- USL can represent concurrent steps, while the other languages do not.
- USL *directly* represents control flows using `USLNodes` and `FlowEdges`, while the other languages do not model the flows directly.

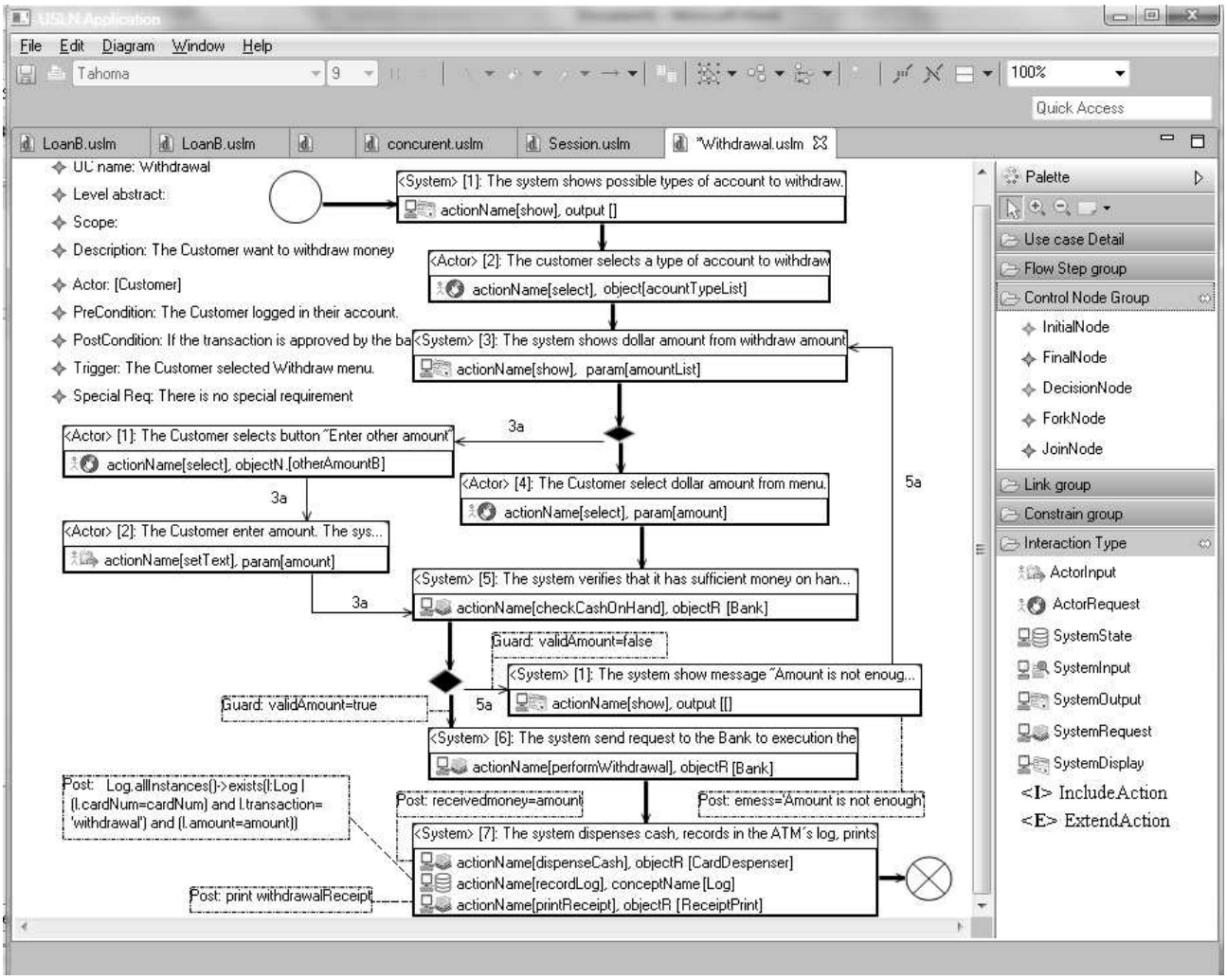
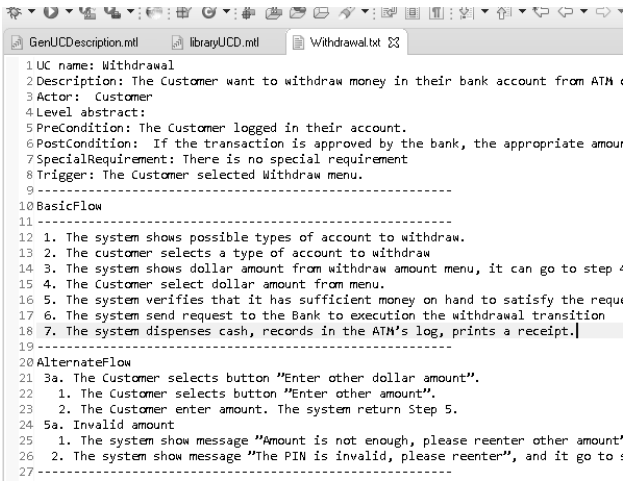


Figure 9: Modelling use case withdrawal in the USL Editor tool.



```
UC name: Withdrawal
Description: The Customer want to withdraw money in their bank account from ATM
Actor:  Customer
Level abstract:
PreCondition: The Customer logged in their account.
PostCondition:  If the transaction is approved by the bank, the appropriate amour
SpecialRequirement: There is no special requirement
Trigger: The Customer selected Withdraw menu.
-------------------------------------------------------
BasicFlow
-------------------------------------------------------
 1. The system shows possible types of account to withdraw.
 2. The customer selects a type of account to withdraw
 3. The system shows dollar amount from withdraw amount menu, it can go to step
 4. The Customer select dollar amount from menu.
 5. The system verifies that it has sufficient money on hand to satisfy the reque
 6. The system send request to the Bank to execution the withdrawal transition
 7. The system dispenses cash, records in the ATM's log, prints a receipt.
-------------------------------------------------------
AlternateFlow
 3a. The Customer selects button "Enter other dollar amount".
   1. The Customer selects button "Enter other amount".
   2. The Customer enter amount. The system return Step 5.
 5a. Invalid amount
   1. The system show message "Amount is not enough, please reenter other amount"
   2. The system show message "The PIN is invalid, please reenter", and it go to
-------------------------------------------------------
```

Figure 10: The TUCD generated from the use case *Withdrawal*.

In comparison with all other works [10, 3, 4, 16, 17], our method can additionally specify concurrent steps. Moreover, these works do not directly specify control flows. They only capture rejoin points or refer to other steps.

6.3.3 Action specification

As discussed in Sect. 4, USL precisely specifies use case behaviors using nine action types. These action types are represented by meta-concepts in the USL metamodel. The action type of each behavior enables us to identify sender objects, receiver objects, messages, parameters of actions and object types. Briefly, we draw the following conclusions from Table 5:

- *Action type coverage*:
  - USL represents all the action types that are supported in other languages.
  - USL complements several action types, compared to four related languages, MBD-L, SiLabReq, RSL, and RUCM. USL employs two new action types `IncludeAction` and `ExtendAction` to represent use case relationships.

```
libraryUCD.mtl   Withdrawal.txt   *Session.txt
1 UC name: Session
2 Description: The Customer insert ATM card into Card reader to exec
3 Actor:  Customer
4 Level abstract:
5 PreCondition: The system started up
6 PostCondition:  If the use case successfully ends, the book loan i
7 -------------------------------------------------------
8 BasicFlow
9 -------------------------------------------------------
10 1. The Customer insert ATM card into the card reader.
11 2. The system read it. If the system can not read card number, it
12 3. The system shows message to ask the PIN.
13 4. The Customer enters PIN
14 5. The system verifies PIN, If pin is invalid, it go to 5a.1
15 6. The show possible types of transaction.
16 7. The Customer selects a transition type, if the selected transit
17 Transfer go to 7b.1, is Inquiry go to 7c.1 and if the customer pre
18 8. The system includes the use case Withdrawal.
19 9. The Customer selects Cancel Key. If the Custom select other tra
20 10. The system rejects the ATM card and end use case
21 -------------------------------------------------------
22 AlternateFlow
23  2a. The ATM cards is invalid
24    1. The system show message "ATM card is invalid".
25    2. The system reject the ATM card, and end the use case.
26  5a. Invalid PIN
27    1. The system counts the times that the Customer entered invali
28    2. The system show message "The PIN is invalid, please reenter"
29  5b. The Custom entered too many invalid PINs
30    1. The system will retain the Customer's card, and end the use
31  7a. The Customer select Deposit transaction
32    1. The system include the use case Deposit, then it go to step
33  7b. The Customer select Transfer transaction
34    1. The system include the use case Transfer, then it go to step
35  7c. The Customer select Inquiry transaction
36    1. The system include the use case Inquiry, then it go to step
37  7d. The customer may abort the session by pressing the Cancel key
38    1. The Custom select cancel key.
39    2. The system reject the ATM card and end use case
40 -------------------------------------------------------
```

Figure 11: The TUCD generated from the use case *Session*.

- *Precise specification*:
  - USL uses the USL meta-concepts to represent actions.
  - The actions in USL are precisely specified using pre- and postconditions. Some languages, e.g., MBD-L, SiLabReq, and RSL also support this feature. Others, namely UC-B and RUCM, do not support it.

We use more action types to classify behaviors and we capture the behavior's information more precisely. More specifically, by using different concepts in USL to specify action types our approach captures behaviors more formally than UC-B. In UC-B, behaviors are not precisely specified and are divided into different action types. Similarly, behaviors are better captured in USL than in RUCM, because the latter only uses keywords and restricted rules in natural language to divide behaviors into action types. Moreover, RUCM does not support the action type named `SystemDisplay`, that is captured in USL.

In comparison with MBD-L, SiLabReq, and RSL, actions in USL are better classified with nine action types. MBD-L uses only four categories of action types: Request, DataValidate, Expletive and Response. Similarly, SeLabReq divides actions into four groups: *Actor prepares Data (APDExecuteSO)*, *Actor calls System (ACSExecuteSO)*, *System executes SystemOperation (SExecuteSO)*, and *System replies and returns Result (SRExecutionSO)*. The classification method of RSL is less specific than USL's, because it does not support the type of system action that sends a request to a primary actor. This system action type is specified in USL by `SystemRequest`.

6.3.4 Constraint representation

USL employs OCL to define constraints in use case. Specifically, a use case's precondition is specified by a `Constraint` associated with the `InitialNode`. A use case scenario's postcondition is specified by a `Constraint` associated with a `FinalNode` of scenario. Similarly, guard conditions on flows and actions' pre- and postconditions are captured by `Constraint` associated with `FlowEdges` and actions, respectively. Briefly, we draw the following conclusions from Table 5:

- USL supports a more complete set of constraints than four other languages, namely RUCM, MBD-L, SiLabReq, and RSL.
- Constraint representation in USL (using OCL) is more precise than two other languages, RUCM and RSL (these languages use natural language to write constraints).

USL specifies more constraint types than four other language: RUCM, MBD-L, SiLabReq, and RSL. Unlike USL, these languages do not support actions' pre- and postcondition. Moreover, USL is better than RUCM and RSL in terms of precision, because several languages, such as MBD-L, SiLabReq, and RSL, also support this feature. The other languages, UC-B and RUCM, do not support it.

It is worth mentioning that constraints specified in USL are quite similar to constraints in UC-B. In the latter, constraints are specified using Event-B's mathematical language. However, this language is rather inconvenient and difficult for non-technical stakeholders to understand.

6.3.5 Applying USL in practice

It is possible to apply USL in practice for two main reasons. First, as discussed in Sect. 5, use cases are precisely specified and represented in USL as models, which conform to a metamodel. This enables them to be automatically transformed into other software artifacts, such as textual use case descriptions, structural and behavioral models and test cases. These generated models are necessary artifacts in software development.

Second, the USL tool realizes our USL approach as an Eclipse modeling project (DSL toolkit) [26]. This tool enables the modeler to visually create USL models and to integrate these models into the existing UML use case models and class model (the latter captures the domain concepts of a system). Moreover, our DSL toolkit provides the meta-metamodel language MOF to build USL. It also enables the definition of model transformation languages in order to realize the transformations discussed in Sect. 5.

However, USL is not without limitations. The graphical concrete syntax of the language might be inconvenient for modelers who prefer writing use cases in the textual form. In order to accommodate for this, the USL tool would be extended with a textual editor, similar to one used in the RSL approach [17]. This textual editor would enable a

modeler to specify use cases by entering descriptive sentences about actions in steps, constraints, and relations between steps. The tool would then process these to create the corresponding USL model.

7 Related work

We position our work in the intersection between use case-driven development [1] and model-driven development [18]. Within this context, a use case model is usually represented as a combination of a UML use case diagram and a textual description written in natural language. Such a use case specification tends to be ambiguous, unclear, and inconsistent. In order to precisely specify use cases several approaches as in [10], [4], [16], [17], [3] have been proposed.

T. Yue *et al*. [3] proposed a use case modeling language called Restricted Use Case Modeling (RUCM), which is composed of a use case description template, a set of keyword, and a set of well-defined restrictions for a restricted natural language to specify use cases. However, the RUCM is semi-formal textual language and it does not mention some important information such as concurrent actions, the pre- and postcondition of actions. Hence, in other work that use RUCM to express use case specifications to automatically generate other artifacts, they have to use NLP(Natural Language Processing) technique to extract information. For example, C. Wang *et al*. [30] uses use case specifications expressed in RUCM in order to generate test cases. After use NLP technique to extract test scenarios and constraints described in natural language, they use OCL to precisely specify constraints and use these precise specifications to automatically generate test data.

R. Murali *et al*. [10] proposed using a mathematical language w.r.t. Event-B in order to formalize the pre- and postcondition of triggers and actions within use case flows. However, other descriptions of a use case are still informal. Their proposition only focus automatically generates a corresponding Event-B model that is then amenable to the Rodin verification tools that enable system-level properties to be verified.

M. Misbhauddin *et al*. [4] extended the metamodel of UML use case models in order to capture both the structural and behavioural aspects of use cases. To specify a use case, they developed a prototype tool called `UCDest`. However, concurrent actions, pre- and postcondition of actions have not been mentioned. Moreover, action types are defined inadequately.

D. Savić *et al*. [16] and M. Smialek *et al*. [17] proposed the DSLs named `SilabReq` and `RSL` in order to capture use cases as the functional requirements models. The DSLs only focus on flows describing use case scenarios while other description information of use case is omitted. In addition, the RSL does not define distinguish actions inserting an extending use case and an included use case, both are defined <invoke> action. Furthermore, the DSLs do not mention concurrent actions, pre and postcondition of actions.

In comparison with all the work above, We provide for USL a formal semantic which use LTS to express , while other works lack a formal semantics.

Our previous work in [36, 9] proposed a metamodel to specify use cases. In that work we also tried to define a precise semantics for use cases based on graph transformation. Our work here continues it by enhancing the use case metamodel as well as proposing a new LTS-based technique in order to characterize the operational semantics of use case.

Furthermore, all above mentioned approaches still lack a method specifying use cases satisfying all relevant information of use cases including flows, steps, system actions, actor actions, control flows, relationships, and constraints on the use case and its flows.

The USL language, introduced in this work, aims to cover all relevant information of a use case including both structural and behavioural aspect. Comparing to the current works in literature, USL could obtain the following advantages: (1) to specify concurrent actions in flows; (2) to capture and represent nine action types in which there are the system action including another use case and the system action extending another use case that have not been mentioned in other research; (3) to present not only constraints on the use case and its flows but pre- and postcondition of each action in flows; (4) to present control flows of steps within the use case. In addition, in this paper we also defined operational semantics of USL to specify dynamic information when use case scenarios execute. In that way, from USL models we could obtain software artifacts by transformations.

8 Conclusion

This paper proposed a DSL named USL to specify use cases. A USL model can cover the relevant information of a use case description including flows, steps, system actions, actor actions, relationships, control flows, and constraints. We built the abstract using a metamodel together with wellformedness rules and the graphical concrete syntax of USL. Moreover, we defined precise semantic for the USL by mapping USL models to LTSs. We also developed a USL Editor to create the USL models visually. In addition, we explained how USL models can be transformed to some software artifacts and developed a model transformation program to automatically generate textual template-based use case descriptions. Moreover, we evaluated USL's expressiveness.

In the future work, we will focus on realizing transformations from USL models in order to generate test cases as well as other software artifacts automatically. In addition, we will enrich the abstract syntax and enhance the concrete syntax of USL in order to support better for modelers.

Acknowledgement

We wish to thank the anonymous reviewers for their useful comments.

References

[1] I. Jacobson, Object-Oriented Software Engineering: A Use Case Driven Approach, Addison Wesley Longman Publishing Co., Inc., 2004.

[2] OMG, "UML 2.5," May 2005.

[3] T. Yue, L.C. Briand, and Y. Labiche, "Facilitating the Transition from Use Case Models to Analysis Models: Approach and Experiments," ACM Trans. Softw. Eng. Methodol., vol.22, no.1, pp.5:1–5:38, March 2013.

[4] M. Misbhauddin and M. Alshayeb, "Extending the UML Use Case Metamodel with Behavioral Information to Facilitate Model Analysis and Interchange," Software & Systems Modeling, vol.14, no.2, May.

[5] P. Kruchten, The Rational Unified Process: An Introduction, 3 ed., Addison-Wesley Professional, 2004.

[6] D. Liu, K. Subramaniam, B.H. Far, and A. Eberlein, "Automating Transition from Use-Cases to Class Model," Proc. Canadian Conf. Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No.03CH37436)(CCECE), 2003.

[7] P. Haumer, "Use case-based software development," in Scenarios, Stories, Use Cases: Through the Systems Development Life-Cycle, ed. I. Alexander and N. Maiden, ch. 12, pp.237–264, Wiley, 2004.

[8] S. Tiwari and A. Gupta, "A Systematic Literature Review of Use Case Specifications Research," Inf. Softw. Technol., vol.67, no.C.

[9] D.H. Dang, "Triple Graph Grammars and OCL for Validating System Behavior," Proc. 4th Int. Conf. Graph Transformations (ICGT), LNCS 5214, pp.481–483, Springer, 2008.

[10] R. Murali, A. Ireland, and G. Grov, "UC-B: Use Case Modelling with Event-B," Proc. 5th Int. Conf. Abstract State Machines, Alloy, B, TLA, VDM, and Z(ABZ), ed. M. Butler, K.D. Schewe, A. Mashkoor, and M. Biro, LNCS 9675, Springer, 2016.

[11] W. Grieskamp and M. Lepper, "Using Use Cases in Executable Z," Proc. 3th Int. Conf. Formal Engineering Methods (ICFEM), pp.111–119, IEEE, 2000.

[12] J.S. Thakur and A. Gupta, "Automatic Generation of Sequence Diagram from Use Case Specification," Proc. 7th India Conf. Software Engineering (ISEC), pp.20:1–20:6, ACM, 2014.

[13] L. Li, "Translating Use Cases to Sequence Diagrams," Proc. 15th Int. Conf. Automated Software Engineering (ASE), pp.293–298, IEEE Computer Society, 2000.

[14] J.M. Almendros-Jiménez and L. Iribarne, "Describing Use Cases with Activity Charts," Proc. Int. Conf. Metainformatics (MIS 2Generation of System Test Cases from Use Case Specifications,"004), LNCS 3511, pp.141–159, Springer-Verlag, 2005.

[15] S. Tiwari and A. Gupta, "An Approach of Generating Test Requirements for Agile Software Development," Proc. 8th on India Conf. Software Engineering (ISEC), ACM, 2015.

[16] D. Savić, S. Vlajić, S. Lazarević, I. Antović, V. Stanojević, M. Milić, and A.R. da Silva, "Use Case Specification Using the SILABREQ Domain Specific Language," Computing and Informatics, vol.34, no.4, pp.877–910, Feb. 2016.

[17] M. Smialek and W. Nowakowski, From Requirements to Java in a Snap: Model-Driven Requirements Engineering in Practice, Springer.

[18] M. Brambilla, J. Cabot, and M. Wimmer, Model-Driven Software Engineering in Practice, 1st ed., Morgan & Claypool Publishers, 2012.

[19] OMG, "OCL 2.0," May 2006.

[20] R.M. Keller, "Formal verification of parallel programs," Commun. ACM, vol.19, no.7, pp.371–384, July 1976.

[21] M.H. Chu, D.H. Dang, N.B. Nguyen, M.D. Le, and T.H. Nguyen, "USL: Towards Precise Specification of Use Cases for Model-Driven Development," Proc. 8th Int. Conf. Information and Communication Technology (SoICT), pp.401–408, 2017.

[22] A. Cockburn, Writing Effective Use Cases, 1 edition ed., Addison-Wesley Professional, Boston, Oct. 2000.

[23] I. Jacobson, I. Spence, and K. Bittner, USE-CASE 2.0 The Guide to Succeeding with Use Cases, Ivar Jacobson International SA., 2011.

[24] M. Giese and R. Heldal, "From Informal to Formal Specifications in UML," Proc. Int. Conf. The Unified Modeling Language: Modelling Languages and Applications (UML), ed. T. Baar, A. Strohmeier, A.M.D. Moreira, and S.J. Mellor, LNCS 3273, 2004.

[25] P. Schmitt, I. Tonin, C. Wonnemann, E. Jenn, S. Leriche, and J.J. Hunt, "A Case Study of Specification and Verification Using JML in an Avionics Application,"

[26] R.C. Gronback, Eclipse Modeling Project: A Domain-Specific Language (DSL) Toolkit, 1 edition ed., Addison-Wesley Professional, Boston, March 2009.

[27] J. Heumann, "Generating Test Cases From Use Cases," tech. rep., Rational Software, 2001.

[28] s. results and B. Legeard, Practical Model-Based Testing: A Tools Approach, 1 edition ed., Morgan Kaufmann, Amsterdam ; Boston, Dec. 2006.

[29] W. Linzhang, Y. Jiesong, Y. Xiaofeng, H. Jun, L. Xuandong, and Z. Guoliang, "Generating Test Cases from UML Activity Diagram Based on Gray-Box Method," Proc. 11th Asia-Pacific Conf. Software Engineering (APSEC), IEEE Computer Society, 2004.

[30] C. Wang, F. Pastore, A. Goknil, L. Briand, and Z. Iqbal, "Automatic Generation of System Test Cases from Use Case Specifications," Proc. Int. Symposium Conf. Software Testing and Analysis (ISSTA), pp.385–396, ACM, 2015.

[31] "Generating Test Data from OCL Constraints with Search Techniques," vol.39.

[32] B. Regnell, M. Andersson, and J. Bergstrand, "A Hierarchical Use Case Model with Graphical Representation," Proc. IEEE Symposium and Workshop on Engineering of Computer-Based Systems (ECBS), pp.270–277, IEEE Computer Society, 1996.

[33] G. Kotonya and I. Sommerville, Requirements Engineering: Processes and Techniques, 1st ed., Wiley Publishing, 1998.

[34] M. Genero Bocco, A. Durán Toro, and B. Bernárdez Jiménez, "Empirical Evaluation and Review of a Metrics-Based Approach for Use Case Verification," Journal of Research and Practice in Information Technology, vol.36, no.4, pp.247–258, 2004.

[35] Russell C. Bjork, "An Example of Object-Oriented Design: An ATM Simulation." `http://www.math-cs.gordon.edu/courses/cs211/ATMExample/`. Accessed: 2018-01-01.

[36] D.H. Dang, A.H. Truong, and M. Gogolla, "Checking the Conformance between Models Based on Scenario Synchronization," Journal of Universal Computer Science, vol.16, no.17, pp.2293–2312, 2010.

https://doi.org/10.31449/inf.v42i3.2301 Informatica **42** (2018) 345–356 **345**

Effective Deep Multi-source Multi-task Learning Frameworks for Smile Detection, Emotion Recognition and Gender Classification

Dinh Viet Sang and Le Tran Bao Cuong
Hanoi University of Science and Technology, 1 Dai Co Viet, Hai Ba Trung, Hanoi, Vietnam
E-mail: sangdv@soict.hust.edu.vn, ltbclqd2805@gmail.com

Keywords: multi-task learning, convolutional neural network, smile detection, emotion recognition, gender classification

Received: March 29, 2018

Automatic human facial recognition has been an active reasearch topic with various potential applications. In this paper, we propose effective multi-task deep learning frameworks which can jointly learn representations for three tasks: smile detection, emotion recognition and gender classification. In addition, our frameworks can be learned from multiple sources of data with different kinds of task-specific class labels. The extensive experiments show that our frameworks achieve superior accuracy over recent state-of-the-art methods in all of three tasks on popular benchmarks. We also show that the joint learning helps the tasks with less data considerably benefit from other tasks with richer data.

Povzetek: Razvita je izvirna metoda globokih nevronskih mrež za tri hkratne naloge: prepoznavanje smeha, čustev in spola.

1 Introduction

In recent years, we have witnessed a rapid boom of artificial intelligence (AI) in various fields such as computer vision, speech recognition and natural language processing. A wide range of AI products have boosted labor productivity, improved the quality of human life, and saved human and social resources. Many artificial intelligence applications have reached or even surpassed human levels in some cases.

Automatic human facial recognition has become an active research area that plays a key role in analyzing emotions and human behaviors. In this work, we study different human facial recognition tasks including smile detection, emotion recognition and gender recognition. All of three tasks use facial images as input. In smile detection task, we have to detect if the people appearing in a given image are smiling or not. We then classify their emotions into seven classes: angry, disgust, fear, happy, sad, surprise and neutral in emotion recognition task. Finally, we determine who are males and who are females in gender classification task.

In general, these tasks are often solved as separate problems. This may lead to many difficulties in learning models, especially, when the training data is not large enough. On the other hand, the data of different facial analysis tasks often shares many common characteristics of human faces. Therefore, joint learning from multiple sources of face data can boost the performance of each individual task.

In this paper, we introduce effective deep convolutional neural networks (CNNs) to simultaneously learn common features for smile detection, emotion recognition and gender classification. Each task takes input data from its corresponding source, but all the tasks share a big part of the networks with many hidden layers. At the end of each network, these tasks are separated into three branches with different task-specific losses. We combine all the losses to form a common network objective function, which allows us to train the networks end-to-end via the back propagation algorithm.

The main contributions of this paper are as follows:

1. We propose effective architectures of CNNs that can learn joint representations from different sources of data to simultaneously perform smile detection, emotion recognition and gender classification.

2. We conduct extensive experiments and achieve new state-of-the-art accuracies in different tasks on popular benchmarks.

The rest of the paper is organized as follows. In section 2, we briefly review related work. In section 3, we present our proposed multi-task deep learning frameworks and describe how to train the networks from multiple data sources. Finally, in section 4, we show the experimental results on popular datasets and compare our proposed frameworks with recent state-of-the-art methods.

2 Related work

2.1 Deep convolutional neural networks

In recent years, deep learning has been proven to be effective in many fields, and particularly, in computer vision. Deep CNNs are one of the most popular models in the family of deep neural networks. LeNet [21], and AlexNet [20]

are known to be the earliest CNN architectures with not many hidden layers.

Latest CNNs such as VGG [33], Inception [35], ResNet [13] and DenseNet [16] tend to be deeper and deeper. In ResNet, residual blocks can be stacked on top of each other with over 1000 layers. Meanwhile, some other CNN architectures like WideResNet [41] or ResNeXt [40] tend to be wider. All these effective CNNs have demonstrated their impressive performances in one of the biggest and the most prestigious competitions in computer vision - the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

2.2 Smile detection

Traditional methods often detect smile based on a strong binary classifier with low-level face descriptors. Shan et al. [32] propose a simple method that uses the intensity differences between pixels in the gray-scale facial images and then combines them with AdaBoost classifier [39] for smile detection. In order to represent faces, Liu et al. [23] use histograms of oriented gradients (HOG) [10], meanwhile, An et al. [4] use local binary pattern (LBP) [3], local phase quantization (LPQ) [25] and HOG. Both of them [23, 4] then apply SVM classifier [9] to detect smiles. Jain et al. [18] propose to use Multi-scale Gaussian Derivatives (MGD) and SVM classifier as well for smile detection.

Some recent methods focus on applying deep neural networks to smile detection. Chen at al. [6] use deep CNNs to extract high-level features from facial images and then use SVM or AdaBoost classifiers to detect smiles as a classification task. Zhang et al. [42] introduce two efficient CNN models called CNN-Basic and CNN 2-Loss. The CNN-2Loss is a improved variant of the CNN-Basic, that tries to learn features by using two supervisory signals. The first one is recognition signal that is responsible for the classification task. The second one is expression verification signal, which is effective to reduce the variation of features which are extracted from the images of the same expression class. [30] proposes an effective VGG-like network, called BKNet, to detect smiles. BKNet achieves better results than many other state-of-the-art methods in smile detection.

2.3 Emotion recognition

Classical approaches to facial expression recognition are often based on Facial Action Coding System (FACS) [11]. FACS includes a list of Action Units (AUs) that describe various facial muscle movements causing changes in facial appearance. Cootes et al. [38] propose a model based on an approach called the Active Appearance Model [8] that creates over 500 facial landmarks. Next, the authors apply PCA algorithm to the set of landmarks and derive Action Units (AUs). Finally, a single layered neural network is used to classify facial expressions.

In Kaggle facial expression recognition competition [1], the winning team [36] proposes an effective CNN, which uses the multi-class SVM loss instead of the usual cross-entropy loss. In [31], Sang et al. propose the so-called BKNet architecture for emotion recognition and achieve better performance compared to previous methods.

2.4 Gender classification

Conventional methods for gender classification often take image intensities as input features. [26] combines the 3D structure of the head with image intensities. [15] uses image intensities combined with SVM classifier. [5] tries to use AdaBoost instead of SVM classifier. [12] introduces a neural network trained on a small set of facial images. [37] uses the Webers Local texture Descriptor [7] for gender classification. More recently, Levi et al. [22] present an effective CNN architecture that yields fairly good performance in gender classification.

2.5 Multi-task learning

Multi-task learning aims to solve multiple classification tasks at the same time by learning them jointly, while exploiting the commonalities and differences across the tasks. Recently, Kaiser et al. [19] propose a big model to learn simultaneously many tasks in nature language processing and computer vision and achieve promising results. Rothe et al. [28] propose a multi-task learning model to jointly learn age and gender classification from images. Zhang et al. [2] propose a cascaded architecture with three stages of carefully designed deep convolutional networks to jointly detect faces and predict landmark locations. Ranjan et al. [27] introduce a multi-task learning framework called hyperface for face detection, landmark localization, pose estimation, and gender recognition. Nevertheless, the hyperface is only trained from a unique source of data with full annotations for all tasks.

3 Our proposed frameworks

3.1 Overall architecture

In this work, we propose effective deep CNNs that can learn joint representations from multiple data sources to solve different tasks at the same time. The merged dataset (Fig. 1) is fed into a block called “CNN Shared Network", which can be designed by using an arbitrary CNN architecture such as VGG [33], ResNet [13] and so on. The motivation of the CNN Shared Network is to help the networks learn the shared features from multiple datasets across different tasks. It is thought that the features learned in the shared block can generalize better and make more accurate predictions than a single-task model. Moreover, thanks to joint representation learning, the tasks with less data can largely benefit from other tasks with more data.

After the shared block, each network is separated into three branches associated with three different tasks. Each

branch learns task-specific features and has its own loss function corresponding to each task.

3.2 Multi-task BKNet

Our first multi-task deep learning framework called Multi-task BKNet has been previously described in [29] (Fig. 3), which is based on the BKNet architecture [30, 31]. We construct the CNN shared network by eliminating three last fully-connected layers of BKNet (Fig. 2).

CNN Shared Network. In this part, we use four convolutional (conv) blocks. The first conv block includes two conv layers with 32 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. The second conv block includes two conv layers with 64 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. The third conv block includes two conv layers with 128 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. Finally, the last conv block includes three conv layers with 256 neurons 3×3 with the stride 1, followed by a max pooling layer 2×2 with the stride 2. Each conv layer is followed by a Batch normalization layer [17] and a ReLU (Rectified Linear Unit) activation function [24]. The Batch normalization layer reduces the internal covariant shift, and, hence, allows us to use higher learning rate when applying the SGD algorithm to accelerate the training process.

Branch Network. After the CNN shared network, we split the network into three branches corresponding to separate tasks, *i.e.*, smile detection, emotion recognition and gender classification. While the CNN shared network can learn joint representations across three tasks from multiple datasets, each branch tries to learn individual features corresponding to each specific task.

Each branch consists of two fully connected layers with 256 neurons and a final fully connected layer with C neurons, where C is the number of classes in each task ($C = 2$ for smile detection and gender classification branch, and $C = 7$ for emotion recognition branch). Note that, after the last fully connected layer, we can either use an additional softmax layer as a classifier or not, depending on what kind of loss function is being used. These kinds of loss function are described in detail in the next section. Similar with the CNN shared network, each fully connected layer in all branches (except the last one) is followed by a Batch Normalization layer and ReLU. Dropout [34] is also utilized in all fully connected layers to reduce overfitting.

3.3 Multi-task ResNet

ResNet [13] is known as one of the most efficient CNN architectures so far. In order to enhance the information flow between layers, ResNet uses shortcut connections between layers. The original variant of ResNet is proposed by He et al. in [13] with different numbers of hidden layers: ResNet-18, ResNet-34 or ResNet-50, ResNet-101 and ResNet-152. He et al. then introduce an improved variant of ResNet (called ResNet_v2) in [14] which shows that the pre-activation order “conv - batch normalization - ReLU" is consistently better then post-activation order “batch normalization - ReLU - conv".

Inspire by the design concept of ResNet_v2, we propose a multi-task ResNet framework to jointly learn three tasks: smile detection, emotion recognition and gender classification. Since the amount of facial data is not large, we choose ResNet-50 (with bottleneck layer) as the base architecture to design our multi-task ResNet framework. In the original ResNet_v2-50 architecture, there are 4 residual blocks, each of which consists of some sub-sampling blocks and identity blocks. The architectures of identity blocks and sub-sampling blocks are shown in Fig. 4a and Fig. 4b. For both these two kinds of blocks, we use the bottleneck architecture with *base depth* m that consists of three conv layers: a 1×1 conv layer with m filters followed by a 3×3 conv layer with m filters and a 1×1 conv layers with $4m$ filters. The identity blocks and sub-sampling blocks are distinguished by the stride value in the second conv layer and the shortcut connection. In sub-sampling blocks, we use a conv layer with stride 2 instead of stride 1 as in identity blocks. The first residual block of ResNet-50 contains only 3 identity blocks and has no sub-sampling block. The next three residual blocks of ResNet-50 have a sub-sampling block at the top, followed by 3, 5 and 2 identity blocks, respectively.

Based on the aforementioned ResNet_v2-50 architecture, we propose two versions of multi-task ResNet framework. In the first version, which is abbreviated as Multi-task ResNet ver1, we use all of 4 residual blocks to build the CNN shared network to learn joint representations for three tasks. Like in multi-task BKNet, for each task in branch network, we use two fully connected layers with 256 neurons combined with a softmax classifier. Fig. 5a illustrates the architecture of Multi-task ResNet ver1.

In the second version, which is abbreviated as Multi-task ResNet ver2, we only use first three residual blocks to build the CNN shared network. For each task in the branch network, we use a separate residual block combined with global average pooling layer and a softmax classifier. Fig. 5b illustrates the architecture of Multi-task ResNet ver2.

3.4 Multi-source multi-task training

In this paper, we propose effective deep networks that can learn to perform multi tasks from different data sources. All data sources are mixed together and form a large common training set (Fig. 1). Generally, each sample in the mixing training set is only related to some of the tasks.

Suppose that:

- T is the number of tasks ($T = 3$ in this paper);
- L_t is the individual loss corresponding to the tth task, $t = 1, 2, ..., T$.

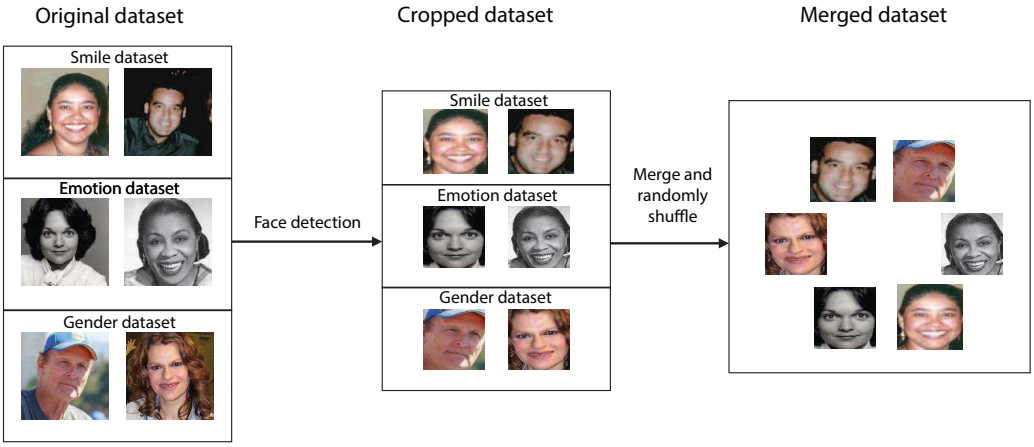


Figure 1: Merged dataset

Figure 2: The CNN shared network in Multi-task BKNet is just the top part (marked by red lines) of the BKNet architecture [30], excluding the last three fully-connected layers.

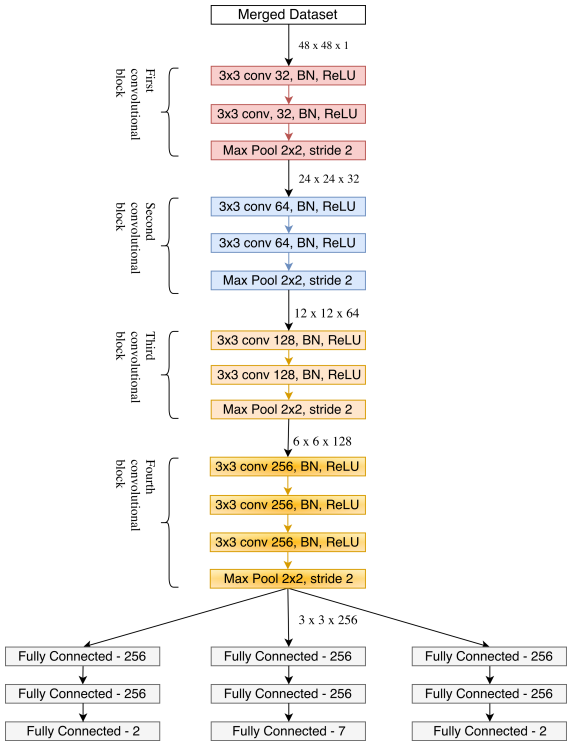
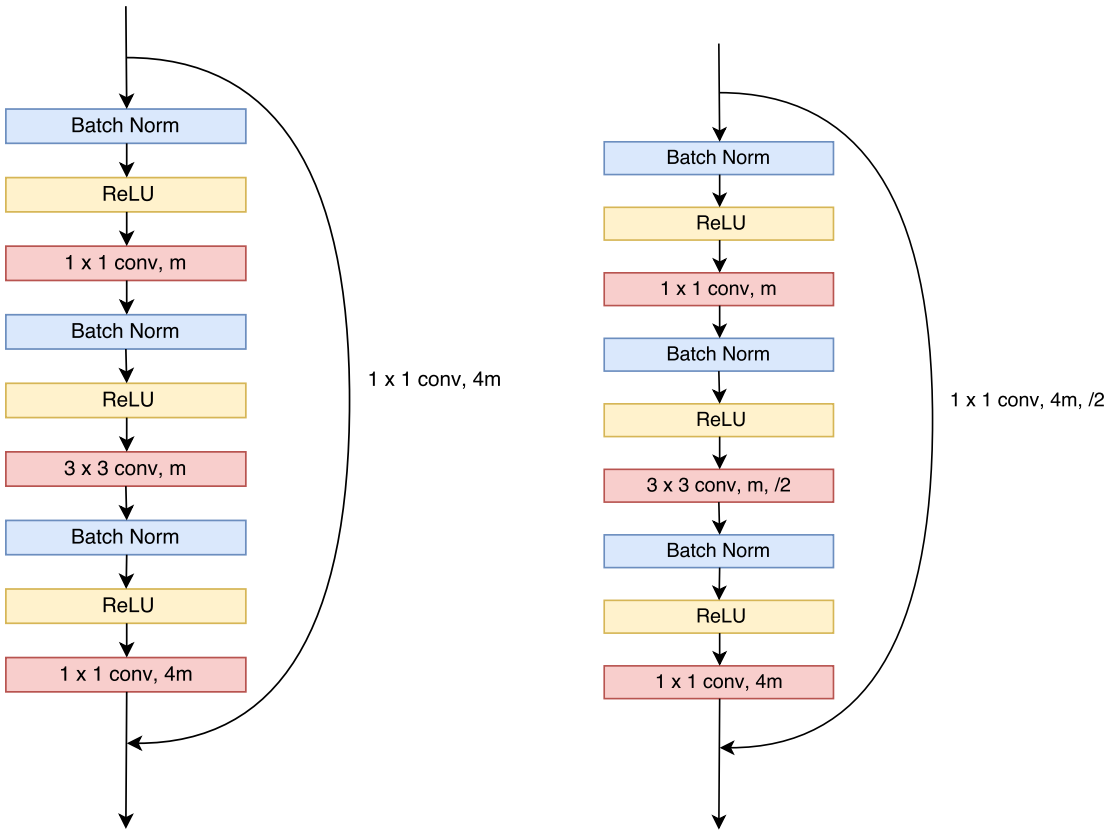


Figure 3: Our proposed Multi-task BKNet

- N is the number of samples from all training datasets;
- C_t is the number of classes corresponding to the tth task ($C_1 = C_3 = 2$ for smile detection and gender classification task, $C_2 = 7$ for emotion recognition task);
- $\mathbf{s}_i^t$ is the vector of class scores corresponding to i-th sample in t^{th} task;
- l_i^t is the correct class label of i-th sample in t^{th} task;
- $\mathbf{y}_i^t$ is the one-hot encoding of the correct class label of i-th sample in t^{th} task ($y_i^t(l_i^t) = 1$);
- $\widehat{\mathbf{y}}_i^t$ is the probability distribution over the classes of i-th sample in t^{th} task, which can be obtained by applying the softmax function to $\mathbf{s}_i^t$.
- $\alpha_i^t \in \{0, 1\}$ is the sample type indicator ($\alpha_i^t = 1$ if the ith sample is related to the tth task, and $\alpha_i^t = 0$ otherwise).

Note that, if the ith sample is not related to t^{th} task, then the true label does not exist, and we can ignore l_i^t and $\mathbf{y}_i^t$. To ensure the mathematical correctness in this case, we can set them to arbitrary values, for instance, $l_i^t = 0$ and $\mathbf{y}_i^t$ is a



(a) Identity block (b) Subsampling block

Figure 4: The architectures of identity blocks and sub-sampling blocks in our Multi-task ResNet framework.

zero vector.

In this paper, we try two kinds of loss: soft-max cross entropy or multi-class SVM loss.

The cross-entropy loss requires to use a softmax layer after the last fully-connected layer of each branch. The cross-entropy loss L_t corresponding to t^{th} task is defined as follows:

$$L_t = -\frac{1}{N}\sum_{i=1}^{N}\left(\alpha_i^t \sum_{j=1}^{C_t} \mathbf{y}_i^t(j) log(\widehat{\mathbf{y}}_i^t(j))\right), \quad (1)$$

where $\mathbf{y}_i^t(j) \in \{0,1\}$ indicates whether j is the correct label of i-th sample; $\widehat{\mathbf{y}}_i^t(j) \in [0,1]$ expresses the probability that j is the correct label of i-th sample.

The multi-class SVM loss function is used when the last fully connected layer in each task-specific branch accompanies with no activation function. The multi-class SVM loss function corresponding to the tth task can be defined as follows:

$$L_t = \frac{1}{N}\sum_{i=1}^{N}\left(\alpha_i^t \sum_{\substack{j=1 \\ j\neq l_i^t}}^{C_t} max(0, \mathbf{s}_i^t(j) - \mathbf{s}_i^t(l_i^t) + 1)^2\right), \quad (2)$$

where $\mathbf{s}_i^t(j)$ indicates the score of class j in the i-th sample; $\mathbf{s}_i^t(l_i^t)$ defines the score of true label l_i^t in the i-th sample.

The total loss of the network is computed as the weighted sum of the three individual losses. In addition, we also add L2 weight decay term associated with all network weights $\mathbf{W}$ to the total network loss to reduce overfitting. The overall loss can be defined as follows:

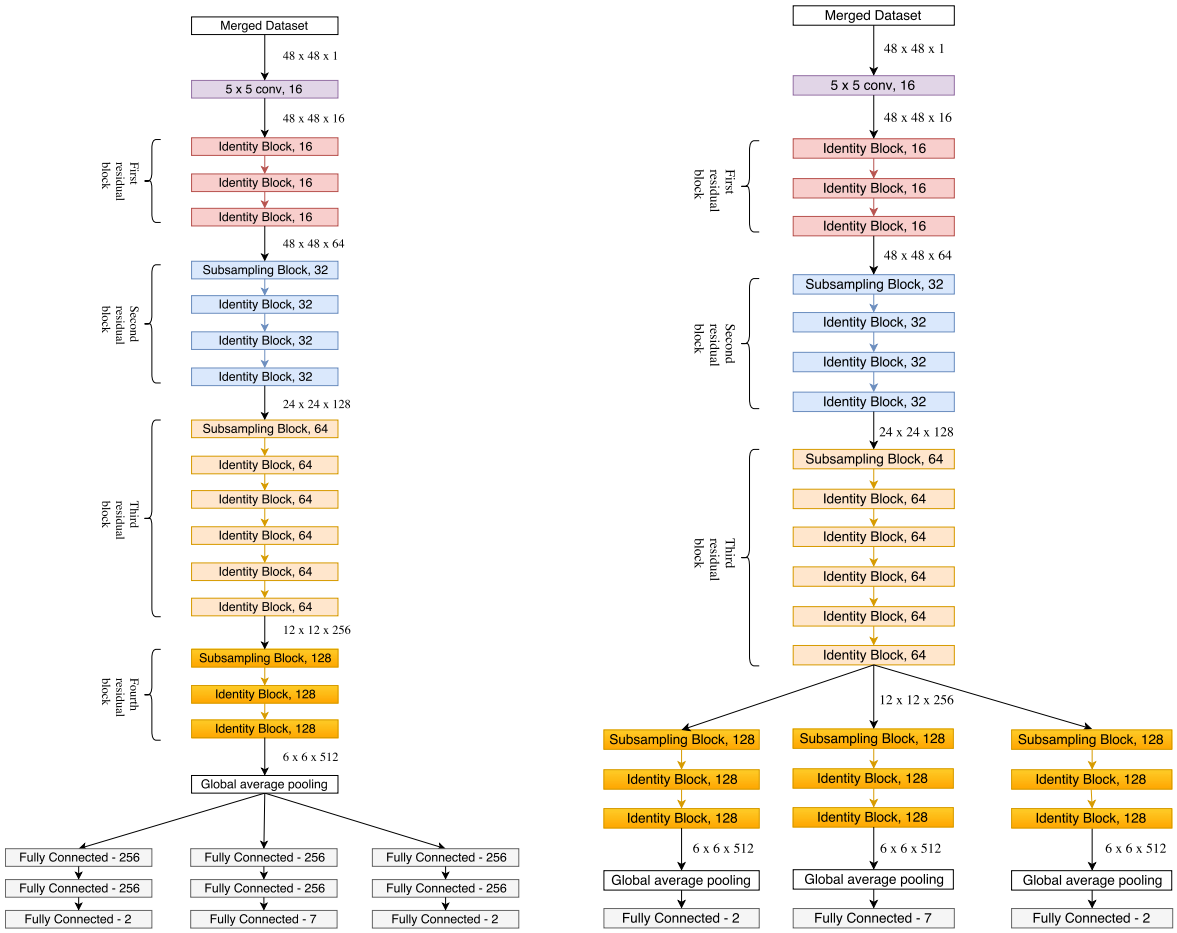
$$L_{total} = \sum_{1}^{T} \mu_t L_t + \lambda \|\mathbf{W}\|_2^2, \quad (3)$$

where μ_t is the importance level of the tth task in the overall loss; λ is the weight decay coefficient.

We train the network end-to-end via the standard back propagation algorithm.

3.5 Data pre-processing

All the images from the datasets that we use later are portraits. Nevertheless, our networks works with facial regions only. Thus, we have to perform data pre-processing to crop faces from the original images in the datasets. Here we use Multi-task Cascaded Convolutional Neural Networks (MTCNN) [2] to detect faces in each image. Fig. 6 shows some examples of using MTCNN for cropping faces.



(a) First version with fully connected layers in the branch network

(b) Second version with residual blocks in the branch network

Figure 5: Our proposed Multi-task ResNet framework. The notation *"Identity block, m"* means the identity block with base depth m.

After that, the cropped images are converted to grayscale and resized to 48×48 ones.

Figure 6: MTCNN for face detection. The top row is original images. The bottom row are cropped faces using MTCNN.

3.6 Data augmentation

Due to small amount of samples in the dataset, we use data augmentation techniques to generate more new data for the training phase. These techniques help us to reduce overfitting and, hence, to learn more robust networks.

We used three following popular ways for data augmentation:

- Randomly crop: We add margins to each image in the datasets and then crop a random area of that image with the same size as the original image;

- Randomly flip an image from left to right;

- Randomly rotate an image by a random angle from $-15°$ to $15°$. The space around the rotated image is then filled with black color.

In practice, we find that applying augmentation techniques greatly improves the performance of the model.

4 Experiments and evaluation

4.1 Datasets

4.1.1 GENKI-4K dataset

GENKI-4K is a well-known dataset used in smile detection task. This dataset includes 4000 labelled images of human face from different ages, and races. Among these pictures, 2162 images were labeled as smile and 1838 images were

labeled as non-smile. The images in this dataset are taken from the internet with different real-world contexts (unlike other face datasets, often taken in the same scene), which makes the detection more challenging. However, some images in the dataset are unclear (not sure whether smile or not). In some previous works, some unclear images are eliminated during the training and testing phases. It is obviously that keeping wrong samples in the dataset intuitively makes the model more likely to be confused during the training phase. In the testing phase, the wrong samples might considerably reduce the overall accuracy, when the model makes true predictions but the data says no. Despite that fact, in this work we still retain all the images in the original dataset in both phases. Fig. 7 shows some examples from GENKI-4K dataset.

Figure 7: Some samples in the GENKI-4K dataset. The top two rows are examples of smile faces and the bottom two rows are examples of non-smile faces.

4.1.2 FERC-2013 dataset

FERC-2013 dataset is provided on the Kaggle facial expression competition. The dataset consists of 35,887 gray images of 48x48 resolution. Kaggle has divided into 28,709 training images, 3589 public test images and 3589 private test images. Each image contains a human face that is not posed (in the wild). Each image is labeled by one of seven emotions: angry, disgust, fear, happy, sad, surprise and neutral. Some images of the FERC-2013 dataset are showed in Fig. 8.

4.1.3 IMDB and Wiki dataset

In this work, we use IMDB and Wiki datasets as data sources for gender classification task.

The IMDB dataset is a large face dataset that includes data from celebrities. The authors take the list of the most popular 100,000 actors as listed on the IMDB website and (automatically) crawl from their profiles date of

Figure 8: Some samples in the FERC-2013 dataset.

birth, name, gender and all images related to that person. The IMDB dataset contains about 470.000 images. In this paper, we only use 170.000 images from IMBD. The Wiki dataset also includes data from celebrities, which are crawled data from Wikipedia. The Wiki dataset contains about 62.000 images and in this work we will use about 34.000 images from this dataset. Fig. 9 shows some samples from IMDB and Wiki datasets.

Figure 9: Some samples in the IMDB and Wiki datasets.

4.2 Implementation detail

In the experiments, we use GENKI-4K dataset for smile detection, FERC-2013 for emotion recognition. We separately use one of the two IMDB and Wiki datasets for gender classification task.

Our experiments are conducted using Python programing-language on computers with the following specifications: Intel Xeon E5-2650 v2 Eight-Core Processor 2.6GHz 8.0GT/s 20MB, Ubuntu Operating

System 14.04 64 bit, 32GB RAM, GPU NVIDIA TITAN X 12GB.

Preparing data: Firstly, we merge three datasets (GENKI-4K, FERC-2013, gender dataset IMDB/Wiki) to make a large dataset. We then create a marker vector to define sample type indicators α_i^t. We always keep the number of training data for each task equally to help the learning process stability. For example, if we train our model with two datasets: dataset A with 3000 samples, dataset B with 30000 samples, we will duplicate dataset A 10 times to make a big dataset with total 60000 samples.

In our work, we divide each dataset into training set and testing set. With GENKI-4K dataset, we use 3000 samples for training and 1000 samples for testing. With FERC-2013 dataset we use data split as provided by Kaggle. With Wiki dataset, we use 30000 samples for training and about 4200 samples for testing. With IMDB dataset, we use 150000 samples for training and about 20000 samples for testing.

Training phase: With Multi-task BKNet architecture, our model is trained end-to-end by using SGD algorithm with momentum 0.9. We set the batch size equal to 128. We initialize all weights using a Gaussian distribution with zero mean and standard deviation 0.01. The L2 weight decay is $\lambda = 0.01$. All the tasks have the same importance level $\mu_1 = \mu_2 = \mu_3 = 1$. The dropout rate for all fully connected layers is set to 0.5. Moreover, we apply an exponential decay function to decay the learning rate through time. The learning rate at step k is calculated as follows:

$$curLr = initLr * decayRate^{m/decayStep}, \quad (4)$$

where $curLr$ is the learning rate at step m; $initLr$ is the initialization learning rate at the beginning of training phase; $decayStep$ is the number of steps when the learning rate decayed.

In our experiment, we set $initLr = 0.01$, $decayRate = 0.8$ and $decayStep = 10000$. We train our Multi-task BKNet model in 250 epochs.

Similar to Multi-task BKNet, we train our Multi-task ResNet end-to-end by using SGD algorithm with momentum 0.9. We set the batch size equal to 128. We initialize all weights using variance scaling initializer (He initializer). The L2 weight decay is 10^{-4}. All the tasks have the same important level $\mu_1 = \mu_2 = \mu_3 = 1$. We train the Multi-task ResNet ver1 in 100 epochs and train the Multi-task ResNet ver2 in 80 epochs. The initial learning rate is 0.05 and then decreased by 10 times whenever the training loss stops improving.

Testing phase: In the testing phase, our model is evaluated by k-fold cross-validation algorithm. This method splits our original data into k parts of the same size. The model evaluation is performed through loops, each loop selects $k - 1$ parts of data as training data and the rest is used for testing model. For the convenience of doing comparison between different methods, we use 4-fold cross-validation algorithm as previous works. We will report the average accuracy and the standard deviation after 4 iterations. Moreover, we test our model with two different loss functions mentioned above.

Furthermore, we combine different checkpoints obtained during the training phases to infer test samples. In the paper, we keep 10 last checkpoints corresponding to 10 last training epochs for inference.

4.3 Experimental results

4.3.1 Multi-task BKNet

In this work, we set up two experiment cases. Firstly, we train our model with GENKI-4K, FERC-2013 and Wiki dataset. Secondly, we train our model with GENKI-4K, FERC-2013 and IMDB dataset. Table 1 shows our experiment setup.

We report our results and compare with previous methods in Table 2. As we can see, using cross-entropy loss function gives better result than using SVM loss function in all cases.

In smile detection task, the best accuracy we achieve is $96.23 \pm 0.58\%$ when we train our model with GENKI-4K, FERC-2013 and IMDB dataset. In all experiment cases, we achieve better results than previous state-of-the-art methods. Especially, the Multi-task BKNet clearly outperforms the single-task BKNet [30]. This fact proves that the smile detection task largely benefits from other tasks thanks to sharing the commonalities between data.

In emotion recognition task, the best accuracy we achieve is $71.03 \pm 0.11\%$ for public test and $72.18 \pm 0.23\%$ for private test. This result considerably outperforms all of previous methods.

In gender classification task, to the best of our knowledge, there are no previous results on the Wiki and IMDB datasets for gender classification. In this paper, we apply the single-task BKNet model [30] and achieve the accuracy $95.82 \pm 0.44\%$ and $91.17 \pm 0.27\%$ on the Wiki and IMDB datasets, respectively. The best accuracy we get on Wiki is $96.33 \pm 0.16\%$ when we train our Multi-task BKNet model on Wiki. The best accuracy we get on IMDB is $92.20 \pm 0.11\%$ when we train our model on IMDB. We also report the test accuracy on IMDB when we train the model on Wiki, and the test accuracy on Wiki when we train the model on IMDB.

In all tasks, the Multi-task BKNet yields comparative results and even better than the single-task BKNet in many cases. Furthermore, it should be emphasized that the Multi-task network can effectively solve all the three tasks by using only a common network instead of three separate ones, which would requires approximately three times more memory storage and computational complexity.

4.3.2 Multi-task ResNet

Based on the experimental results of Multi-task BKNet, we will choose the best config B4 in Table 1 to evaluate our

Table 1: Experiment setup

Name	Datasets	Loss function	Use ensemble?
Config A1	GENKI-4K, FERC-2013, IMDB	SVM loss	No
Config A2	GENKI-4K, FERC-2013, IMDB	Cross-entropy loss	No
Config A3	GENKI-4K, FERC-2013, IMDB	SVM loss	Yes
Config A4	GENKI-4K, FERC-2013, IMDB	Cross-entropy loss	Yes
Config B1	GENKI-4K, FERC-2013, Wiki	SVM loss	No
Config B2	GENKI-4K, FERC-2013, Wiki	Cross-entropy loss	No
Config B3	GENKI-4K, FERC-2013, Wiki	SVM loss	Yes
Config B4	GENKI-4K, FERC-2013, Wiki	Cross-entropy loss	Yes

Table 2: Accuracy comparison on four datasets

Method	GENKI-4K	FERC-2013		Wiki	IMDB
		Public test	Private test		
Chen et al [6]	91.8 ± 0.95	-	-	-	-
CNN Basic [42]	93.6 ± 0.47	-	-	-	-
CNN 2-Loss [42]	94.6 ± 0.29	-	-	-	-
Single-task BKNet + Softmax [30]	95.08 ± 0.29	-	-	95.82 ± 0.44*	91.16 ± 0.27*
CNN (team Maxim Milakov - rank 3 Kaggle)	-	68.2	68.8	-	-
CNN (team Unsupervised - rank 2 Kaggle)	-	69.1	69.3	-	-
CNN+SVM Loss (*team RBM*) [36]	-	69.4	71.2	-	-
Single-task BKNet + SVM loss [31]	-	71.0	71.9	-	-
Our Multi-task BKNet (Config A1)	95.25 ± 0.43	68.10 ± 0.14	69.10 ± 0.57	93.33 ± 0.19	89.60 ± 0.22
Our Multi-task BKNet (Config A2)	95.56 ± 0.66	68.47 ± 0.33	69.40 ± 0.21	93.67 ± 0.26	90.50 ± 0.24
Our Multi-task BKNet (Config A3)	95.60 ± 0.41	70.43 ± 0.19	71.90 ± 0.36	93.70 ± 0.37	91.33 ± 0.42
Our Multi-task BKNet (Config A4)	**96.23±0.58**	70.15 ± 0.19	71.62 ± 0.39	94.00 ± 0.24	**92.20±0.11**
Our Multi-task BKNet (Config B1)	95.25 ± 0.44	68.60 ± 0.27	69.28 ± 0.41	95.25 ± 0.15	88.18 ± 0.26
Our Multi-task BKNet (Config B2)	95.13 ± 0.20	69.12 ± 0.18	69.40 ± 0.22	95.75 ± 0.18	88.68 ± 0.15
Our Multi-task BKNet (Config B3)	95.52 ± 0.37	70.63 ± 0.11	71.78 ± 0.08	95.95 ± 0.15	88.83 ± 0.18
Our Multi-task BKNet (Config B4)	95.70 ± 0.25	**71.03±0.11**	**72.18±0.23**	**96.33±0.16**	89.34 ± 0.15
Our Multi-task ResNet ver1 (Config B4)	95.55 ± 0.28	70.09 ± 0.13	71.55 ± 0.19	96.03 ± 0.22	89.01 ± 0.18
Our Multi-task ResNet ver2 (Config B4)	95.30 ± 0.34	69.33 ± 0.31	71.27 ± 0.11	95.99 ± 0.14	88.88 ± 0.07

Multi-task ResNet frameworks.

The results of our Multi-task ResNet are also shown in Table 2. As one can see, our first version yields better results than the second version in all three tasks.

In smile detection task, the first version of multi-task ResNet achieves $95.55 \pm 0.28\%$ accuracy, while the second version achieves $95.30 \pm 0.34\%$ accuracy. With the same config B4, our Multi-task BKNet model achieves $95.70 \pm 0.25\%$ accuracy, which is slightly better then Multi-task ResNet.

In emotion recognition task, the accuracy of the first version of Multi-task ResNet is $70.09 \pm 0.13\%$ for public test set and $71.55 \pm 0.19\%$ for private test set. The accuracy of the second version is a little bit lower with $69.33 \pm 0.31\%$ and $71.27 \pm 0.11\%$ for public test set and private test set, respectively. In this task, both versions of Multi-task ResNet seem to clearly lose Multi-task BKNet, which obtains higher approximately 1% accuracy in each test set.

In gender classification task, both our variants of multi-task ResNet yield pretty good results, which compete with the results of of the multi-task BKNet model. The first variant achieves the accuracy of $96.03 \pm 0.22\%$ and $89.01 \pm 0.18\%$ for Wiki dataset and IMDB dataset, respectively. The second variant achieves the accuracy of $95.99 \pm 0.14\%$ for Wiki dataset and $88.88 \pm 0.07\%$ for IMDB dataset.

The experiment results show that the Multi-task ResNet is slightly worse than the Multi-task BKNet in all tasks. The reason could be due to that ResNet with a pretty deep architecture and fairly large number of parameters tends to be over-complex w.r.t the mixing training data across the three tasks and leads to overfitting. Meanwhile, BKNet is quite smaller than ResNet, and is capable to fit the data

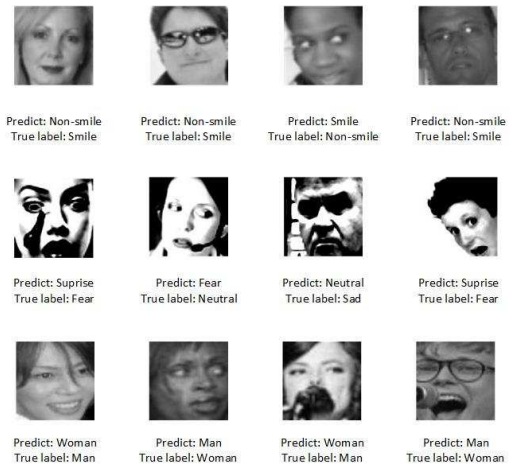


Figure 10: Some samples that our Multi-task BKNet gives wrong predictions.

better.

4.3.3 Speed performance comparison between different frameworks

In Table 3 and Table 4, we show the inference time and training time of three frameworks: Multi-task BKNet, Multi-task ResNet ver1 and Multi-task ResNet ver2 with Config B4 (from Table 1).

As one can see, the Multi-task ResNet ver2 acquires the fastest convergence. Despite a little longer in training time, Multi-task BKNet is significantly faster in inference in comparison with both versions of Multi-task ResNet. The fast inference with high accuracy make the Multi-task BKNet well suitable for real-time applications.

Table 3: Comparison of inference time between different frameworks

Framework	Inference time per image (sec)
Multi-task BKNet	0.02
Multi-task ResNet ver1	0.065
Multi-task ResNet ver2	0.071

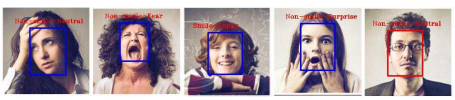


Figure 11: Some results of our Multi-task BKNet framework. The blue box corresponds to females and the red box corresponds to males.

5 Conclusion

In this paper, we propose effective multi-souce multi-task deep learning frameworks to jointly learn three facial analysis tasks including smile detection, emotion recognition and gender classification. The extensive experiments in well-known GENKI-4K, FERC-2013, Wiki, IMDB datasets show that our frameworks achieve superior accuracy over recent state-of-the-art methods in all tasks. We also show that the smile detection task with few data largely benefit from the two other tasks with richer data.

In the future, we would like to exploit some new auxiliary losses to regulate the model learning process in order to improve the performance accuracy of neural networks in various computer vision tasks.

6 Acknowledgments

This research is funded by Hanoi University of Science and Technology under grant number T2016-LN-08.

References

[1] Challenges in respresentation learning: Facial expression recognition challenge, 2013.

[2] Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. https://doi.org/10.1109/lsp.2016.2603342.

[3] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. *Computer vision-eccv 2004*, pages 469–481, 2004.

[4] L. An, S. Yang, and B. Bhanu. Efficient smile detection by extreme learning machine. *Neurocomputing*, 149:354–363, 2015. https://doi.org/10.1016/j.neucom.2014.04.072.

[5] S. Baluja, H. A. Rowley, et al. Boosting sex identification performance. *International Journal of computer vision*, 71(1):111–119, 2007. https://doi.org/10.1007/s11263-006-8910-9.

[6] J. Chen, Q. Ou, Z. Chi, and H. Fu. Smile detection in the wild with deep convolutional neural networks. *Machine vision and applications*, 28(1-2):173–183, 2017. https://doi.org/10.1007/s00138-016-0817-z.

[7] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao. Wld: A robust local image descriptor. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1705–1720, 2010. https://doi.org/10.1109/tpami.2009.155.

Table 4: Comparison of training time between different frameworks

Framework	Number of epochs	Training time per epoch (min)	Total training time (min)
Multi-task BKNet	250	3.42	854
Multi-task ResNet ver1	100	8.12	817
Multi-task ResNet ver2	80	8.67	693

[8] T. F. Cootes, C. J. Taylor, et al. Statistical models of appearance for computer vision, 2004.

[9] C. Cortes and V. Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[10] O. Déniz, G. Bueno, J. Salido, and F. De la Torre. Face recognition using histograms of oriented gradients. *Pattern Recognition Letters*, 32(12):1598–1603, 2011. https://doi.org/10.1016/j.patrec.2011.01.004.

[11] P. Ekman and E. L. Rosenberg. *What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS)*. Oxford University Press, USA, 1997. https://doi.org/10.1093/acprof:oso/9780195179644.001.0001.

[12] B. A. Golomb, D. T. Lawrence, and T. J. Sejnowski. Sexnet: A neural network identifies sex from human faces. In *NIPS*, volume 1, page 2, 1990.

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. https://doi.org/10.1109/cvpr.2016.90.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer, 2016. https://doi.org/10.1007/978-3-319-46493-0_38.

[15] X. He and P. Niyogi. Locality preserving projections. In *Advances in neural information processing systems*, pages 153–160, 2004.

[16] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017. https://doi.org/10.1109/cvpr.2017.243.

[17] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[18] V. Jain and J. L. Crowley. Smile detection using multi-scale gaussian derivatives. In *12th WSEAS International Conference on Signal Processing, Robotics and Automation*, 2013.

[19] L. Kaiser, A. N. Gomez, N. Shazeer, A. Vaswani, N. Parmar, L. Jones, and J. Uszkoreit. One model to learn them all. *arXiv preprint arXiv:1706.05137*, 2017.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. https://doi.org/10.1109/5.726791.

[22] G. Levi and T. Hassner. Age and gender classification using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 34–42, 2015. https://doi.org/10.1109/cvprw.2015.7301352.

[23] M. Liu, S. Li, S. Shan, and X. Chen. Enhancing expression recognition in the wild with unlabeled reference data. In *Asian Conference on Computer Vision*, pages 577–588. Springer, 2012. https://doi.org/10.1007/978-3-642-37444-9_45.

[24] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[25] V. Ojansivu and J. Heikkilä. Blur insensitive texture classification using local phase quantization. In *International conference on image and signal processing*, pages 236–243. Springer, 2008. https://doi.org/10.1007/978-3-540-69905-7_27.

[26] A. J. O'toole, T. Vetter, N. F. Troje, and H. H. Bülthoff. Sex classification is better with three-dimensional head structure than with image intensity information. *Perception*, 26(1):75–84, 1997. https://doi.org/10.1068/p260075.

[27] R. Ranjan, V. M. Patel, and R. Chellappa. HyperFace: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1,

2017. https://doi.org/10.1109/tpami.2017.2781233.

[28] R. Rothe, R. Timofte, and L. Van Gool. Dex: Deep expectation of apparent age from a single image. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 10–15, 2015. https://doi.org/10.1109/iccvw.2015.41.

[29] D. V. Sang, L. T. B. Cuong, and V. V. Thieu. Multi-task learning for smile detection, emotion recognition and gender classification. In *Proceedings of the Eighth International Symposium on Information and Communication Technology, Nha Trang City, Viet Nam, December 7-8, 2017*, pages 340–347, 2017. https://doi.org/10.1145/3155133.3155207.

[30] D. V. Sang, L. T. B. Cuong, and D. P. Thuan. Facial smile detection using convolutional neural networks. In *The 9th International Conference on Knowledge and Systems Engineering (KSE 2017)*, pages 138–143, 2017. https://doi.org/10.1109/kse.2017.8119448.

[31] D. V. Sang, N. V. Dat, and D. P. Thuan. Facial expression recognition using deep convolutional neural networks. In *The 9th International Conference on Knowledge and Systems Engineering (KSE 2017)*, pages 144–149, 2017. https://doi.org/10.1109/kse.2017.8119447.

[32] C. Shan. Smile detection by boosting pixel differences. *IEEE transactions on image processing*, 21(1):431–436, 2012. https://doi.org/10.1109/tip.2011.2161587.

[33] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958, 2014.

[35] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. https://doi.org/10.1109/cvpr.2015.7298594.

[36] Y. Tang. Deep learning using support vector machines. *CoRR, abs/1306.0239*, 2, 2013.

[37] I. Ullah, M. Hussain, G. Muhammad, H. Aboalsamh, G. Bebis, and A. M. Mirza. Gender recognition from face images with local wld descriptor. In *Systems, Signals and Image Processing (IWSSIP), 2012 19th International Conference on*, pages 417–420. IEEE, 2012.

[38] H. Van Kuilenburg, M. Wiering, and M. Den Uyl. A model based method for automatic facial expression recognition. In *Proceedings of the 16th European Conference on Machine Learning (ECML'05)*, pages 194–205. Springer, 2005. https://doi.org/10.1007/11564096_22.

[39] P. Viola and M. Jones. Fast and robust classification using asymmetric adaboost and a detector cascade. In *Advances in neural information processing systems*, pages 1311–1318, 2002.

[40] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5987–5995. IEEE, 2017. https://doi.org/10.1109/cvpr.2017.634.

[41] S. Zagoruyko and N. Komodakis. Wide residual networks. In *Procedings of the British Machine Vision Conference 2016*. British Machine Vision Association, 2016. https://doi.org/10.5244/c.30.87.

[42] K. Zhang, Y. Huang, H. Wu, and L. Wang. Facial smile detection based on deep learning features. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 534–538. IEEE, 2015. https://doi.org/10.1109/acpr.2015.7486560.

Alignment-free Sequence Searching over Whole Genomes Using 3D Random Plot of Query DNA Sequences

Da-Young Lee, Hae-Sung Tak, Han-Ho Kim and Hwan-Gue Cho
Dept. of Electrical and Computer Engineering, Pusan National University, South Korea
E-mail: {schematique, tok33, quant1216, hgcho}@pusan.ac.kr

Keywords: sequence search, sequence visualization, whole genome, 3D random plot

Received: March 29, 2018

Most genomic data studies are based on sequence comparisons and searches, and comparison models based on alignment algorithms are most commonly used. This method is very accurate, but it is useful when the query is short in kilobytes, because it requires the quadratic time and space complexity, $O(n^2)$ where n is the length of target and query sequences. With the development of Next Generation Sequencing techniques, researches on whole genome sequence data of megabyte size are being actively studied, and new comparison and search methods for large-scale sequence data are needed. We propose a new alignment-free sequence comparison and search method to overcome the limitations of the alignment-based model. In this graphical model, the sequence searching problem in DNA strings can be reduced to find some parts of geometric object within a relatively small-scale geometric space. When comparing similarity by modifying sequences of similar length, we can confirm that the comparison model is appropriate by accurately reflecting the degree of similarity. When searching the query sequence comparison model based on 200MB sized whole genome sequence, using the compressed coordinate information, it was able to search the 10MB sequences in 22s, which is a very reduced time compared to alignment. Although it is not possible to find the exact position of the base pair unit as in the alignment result, it is a model that can be used as a preprocessing process to quickly search a whole genome sequence of several hundred megabytes-size.

Povzetek: Na podlagi 3D vizualizacije celotnega zaporedja genoma so avtorji pokazali, da je na dolžini poizvedbe možno prilagodljivo hitro iskanje.

1 Introduction

Genomic data studies are done through sequence comparisons, mostly using a model based on an alignment algorithm. For example, Basic Local Alignment Search Tool (BLAST)[1] is the most common method to search for sequences in a database. It divides the query sequence into three characters, finds the matching region, and gradually widens the region to select candidates for alignment. Although it is very useful when searching for a short query in the whole database, since it is based on alignment, it is difficult to obtain an immediate processing result in the case of a large sequence such as a megabyte-scale chromosome owing to a large increase in computational cost. When utilizing the actual BLAST service, it is recommended to reduce the database search scope when the query size is of the order of megabytes, and it is often time consuming to search and provide results by mail, rather than providing it immediately.

In addition, since gene recombination is different from sequence alignment based on conservation of contiguity between homologous segments, in order to overcome this problem, alignment-free comparison method such like word-frequency statistics, a method of calculating distance in space defined by frequency vectors, is also actively underway[2]. Such research is also widely used as a prefilter for processing queries of alignment-based models.

In this paper, we propose a geometric-based heuristic technique that enables the rapid comparison and search of sequences in personal computers. In this regard, AMSS[3] is a model that provides shape-based similarity comparison, assuming that the time series data is a vector sequence. Instead of focusing on individual points of time series data, the model focuses on vectors and compares similarities between data using cosine similarity. This method is advantageous in that it is effective for amplitude and time shifting. In this study, we also aimed to reduce the time and space complexity by converting the genetic sequence into a geometric object such as a random plot and performing comparison and search, taking into account that the genetic sequence data is ordered sequence data. Instead of considering a single separate base, as in the alignment algorithm, the method compares the vector generated based on the sequence of the predetermined unit only once, and it is possible to significantly reduce the time required for comparison operation by visualizing a sequence search result and presenting the information more intuitively. In addition, the high-speed heuristic search technique can be applied to large amounts of data, and it is possible to specify the necessary precise alignment analysis.

Compared to [14], we present an improved similarity computation algorithm that considers input sequences with different lengths. We show the effectiveness of the proposed method with experiments on searching for short query sequences on a long sequence.

2 Related work

2.1 Genome Sequence Visualization

Most genetic data have a huge volume, and it is difficult to find meaningful patterns in such data owing to the irregular configuration of the four bases. The visualization of sequence information and sequence analysis information can help in forming an intuitive understanding of the genomic data and enable the efficient representation of the results. Genome visualization research focuses on two aspects. The first is the visualize of a large amount of genetic information in a short time and a limited space, and the second is the representation of complex information as intuitively as possible.

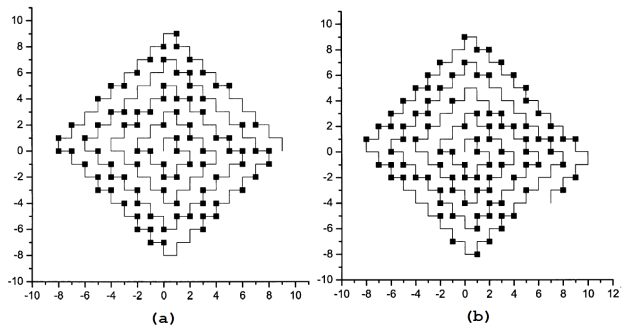


Figure 1: The compact graphical representation [4] of the first exon of human β-globin gene(a) and gorilla β-globin gene. The visualization of search result for query sequence of 10M size in human chromosome 1.

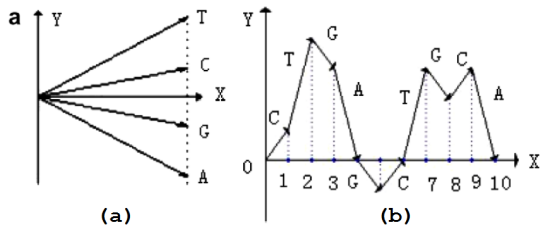


Figure 2: The vector design of 'H-L curve'[5] (a) and graphical representation for the DNA sequence $s =$ `'ATGGCATGCA'` (b).

The 'Worm Curve'[4, 6] represents genome information in a limited space, and it assigns a binary code to each base. It is plotted on a Cartesian coordinate system, and its most significant biggest advantage is that the curve can represent all the information in a relatively small space, despite how little the point intersects with each other. Studies have been actively conducted using a variety of curves to intuitively represent complex information. For example, the 'Dual-Base Curve' (DB-Curve)[7] has been designed to visualize the features of a genome sequence at a glance. In this curve, the two different bases are configured as a combination, and a two-dimensional vector is assigned, where the y component is assigned as a constant (+1) and the x components are assigned separately. In this visualized, since the curve is continuous in the positive direction of the y axis, there is no point at which it crosses with itself. Obtaining a ratio of the x-coordinates of the end points can confirm the relative existing ratio of the two bases to obtain the statistical information of the sequence in an intuitive manner.

In contrast, the 'H-L curve'[5] is a method of assigning a two-dimensional vector for the four bases with a constant x component, and this curve avoids intersection with itself because different y-components are assigned. Since the progress of a DNA sequence matches one-to-one with the 'H-L Curve,' it has the advantage that the main difference of each sequence with other sequences can be checked quickly.

In addition to visualizing curves, there is a 'Four-Color Map'[8], which assigns colors to each base and fills areas proportional to the frequency of occurrence with the corresponding color, and 'Circos'[9, 10], which visualizes the whole genome in a circular track form. 'Circos' represents a chromosome as a piece of a circular track, and connects the interactive chromosome tracks with a curve, thereby effectively expressing the internal relation of the whole genome. Although most relational connection visualization methods express only one-to-one associations, 'Circos' can express many-to-many associations as well by using circular tracks.

2.2 Visualization Tool for Genome Sequence

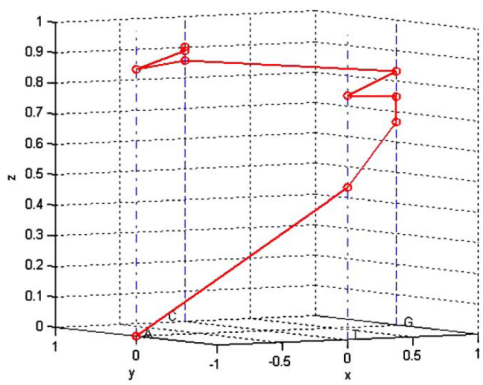


Figure 3: 3D graphical representation of DNA sequence using Z-axis as time axis[11]. The graphical representation for the sequence `'ATGGTGCACC'`.

To compensate for the drawbacks of the sequence alignment method in terms of processing speed, a heuristic method based on visualization is utilized. By converting

a large amount of text information composed of only four kinds of bases, the meaning of which is difficult to intuitively grasp, to geometry information, heuristic methods are able to identify the type of data through visual examination to easily find patterns that cannot be revealed using computational methods[12]. Furthermore, geometric rules found in the visible results often have a meaningful relationship with genomic analysis in the field. Heuristic methods are especially useful when utilized for quickly calculating similarity or dissimilarity.

For example, large-scale genomic sequence information is converted into information on a polygon domain, and the problem of finding similarity is solved by replacing the comparison of similarity of sequences with the comparison of image similarity[13]. By setting a direction for each base, the sequence is converted to a random plot in which the polygon area is simplified with the k-convex hull, and the homology of two random plots is compared. Studies [14, 15] have considered the extended space up to three dimensions in the vector assignment for each base. Consequently, a random plot can be visualized on three dimensions, and the similarity can be compared by simplifying it to be close to the actual random plot.

Since direct comparison is difficult for a walk-plot object in three dimensions, a random plot is populated in a certain space around the polygon area, and the orthogonal projection of this space on each plane (X-Y, Y-Z, and X-Z) is used to compare the degree of similarity using the overlap area ratio. However, the comparison method based on the overlapping area has a drawback in that it does not take into account the random plot present in the local area. To overcome this drawback without simplifying the random plot, the shape of the line is maintained while the shortest distance between any points of two random plots is calculated for comparing the degree of similarity between two sequences[16].

Previously, an alignment method called 'Four Line' involving graphical-domain sequence alignment, rather than string alignment, was proposed[17]. By assigning the four bases to different points on the Y-axis and connecting the matched points in the sequence to be subjected to alignment in the X-axis to make a visualization of the zigzag curve, the visualization result of the two sequences are compared to conduct alignment.

In order to overcome the disadvantages such as loss of information and self-intersection of existing two-dimensional visualization methods, there is a study in which a DNA sequence is three-dimensionally utilized as a time axis[11]. Regardless of the information of the base to the z-axis will always increases, and by assigning vectors x, y axis is increased or decreased for each base. Not only it limited to visualization, to derive the geometrical center of the curve, this time the center of this curve is important information indicating the distribution of each base. In this study, a similarity comparison model was devised by assigning vectors to each other in different ways and using the Euclidean distance and angle correlation of the distance to the start and end points of the vector through eight transform. As a result, they could construct the similarity matrix, it shown that the similar species such as human and gorilla have high similarity.

In this manner, visualization results can be used not only for the intuitive delivery of sequence information but also as an analysis target to improve the processing speed and to obtain meaningful results. In this study, by focusing on this point, we convert a whole genome sequence to a walk-plot object in three-dimensional space, extract a vector, and compare and search for the sequence with improved speed. Furthermore, by visualizing a search query sequence together with the random plot of the whole genome sequence, the position and distribution of the obtained similar sequence can be transferred in an intuitive form.

Table 1: Functional Performance of Previous Research

Research	Plotting space dimension	Supports large-scale sequence	Global similarity compute	Local similarity compute
BLAST [1]	N/A	△	O	O
Compact 2D [4]	2D	O	O	X
H-L Curve [5]	2D	△	X	X
Bo Liao [11]	3D	△	O	X
3D Random [15]	3D	O	O	X
Proposed	3D	O	O	O

3 New method using 3D random plot

3.1 Sequence Searching method with 3D Random Plot Structure

An overview of our algorithm framework is shown in Figure 4. Generally, all types of biological sequence comparison exploit the sequence alignment based on a dynamic programming approach. One popular alignment algorithm is the Needlemann–Wunsch algorithm, which is widely used in molecular biology. There are many variations in sequence alignment, such as global alignment, local alignment, and semi-global alignment. Though the alignment approach has many advantages, it has a critical drawback in that it involves high complexity in terms of execution-time complexity and space complexity. The complexity of the basic alignment algorithm is $O(m \cdot n)$ if the lengths of two input sequences are n and m. If $\Theta(n) = \Theta(m)$, the complexity is quadratic: $O(n^2)$. When the size of the input sequence is greater than 100 megabytes, this alignment is impractical, because it requires a main memory greater than the order of gigabytes. To overcome these problems, researchers developed heuristic alignment techniques such as BLAST-like tools. Another problem in the alignment algorithm is that it is not easy to define the score/penalty matrix to meet the many different constraints in biological sequence comparison.

The basic idea of our approach is that we compute the similarity of two sequences in 'geometric random plot'

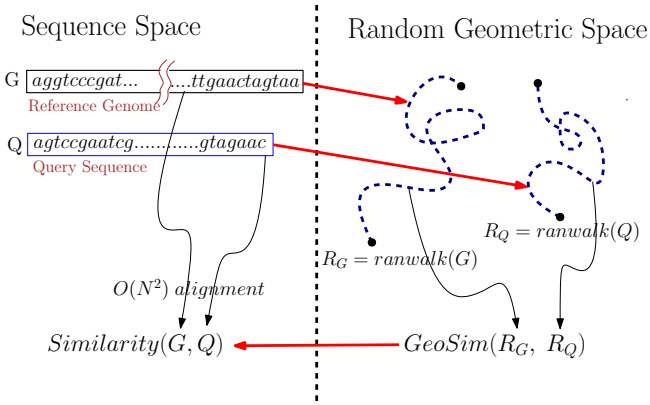


Figure 4: Space transform from sequence to 3D geometric shape.

space, rather than 'string sequence' space. As shown in Figure 4, we first transform the input sequences into random plot in 3D space. Then, we compare or search for a target sequence in 3D geometric object.

This transformed random plot can be visualized on an appropriately sized grid, and a sequence of megabytes in size can be represented by a list of pixels much smaller than the actual number of bp.

Thus, we can say that our geometric transformation is a type of approximation with visualization. The advantage of our transformation is that the global structure can be shown by hiding the biological noise embedded in the sequence. The main merit of our approach is that it is useful and efficient in comparing very long sequences. Assume that we are asked to find the location of a sequence that is a few megabytes in length in a whole genome longer than 100 megabytes.

3.2 Vector Allocation for random Plot

Sequence data are string information composed of {a,g,t,c}; therefore, they must be converted into graphical information for visualization. Previous 2-D visualization methods have visualized genome sequences by assigning a separate base in the positive and negative directions of each axis (x and y). This method has a disadvantage in that a large amount of information is lost when a base having a vector in opposite directions is continuously repeated. Furthermore, if the same pattern is continuously repeated, it is impossible to visualize a large volume of data in a limited space. To overcome this disadvantage, [15] used a 3D vector. A vector is assigned to each base, but a combination of two bases constitutes a random plot. When the two bases are coupled together with the vector in the opposite direction, the representation is made three-dimensional with a z-axis to minimize the lost information. In this study, by using a 3D vector allocation model[15], we calculate the vector character of the sequence data and obtain sequence search positions to visualize the results.

Table 2: Vector allocation method for each 2-mer base in a genome sequence in three-dimensional geometric space

2-mer	Vector	2-mer	Vector
AA	(2, 0, 0)	AG(GA)	(1, 1, 0)
AC(CA)	(1, -1, 0)	AT(TA)	(0, 0, -2)
CC	(0, -2, 0)	CG(GC)	(0, 0, +2)
CT(TC)	(-1, -1, 0)	GG	(0, 2, 0)
GT(TG)	(-1, 1, 0)	TT	(-2, 0, 0)

Table 2 summarizes the vector allocation method for each 2-mer. In Table 2, the base pairs AT and GC are represented on the z axis. The other base pairs are represented as the sum of two unit vectors for each base, as given by the WS-curve method.

After the vector transition for DNA genome data information, those vectors are visualized in three-dimensional space. The method of visualization is the same as that of two-dimensional visualization, where the sum of vector values is computed according to the order of sequences and the results are connected with a line to provide the final visualization result. For the random plot R, the starting point is $R(0) = (X_0, Y_0, Z_0)$ $(X_0 = Y_0 = Z_0 = 0)$. $Unit^{3d}(i)$ is the converted value of the ith 2-mer of the unit vector. The ith point $R(i) = (X_i, Y_i, Z_i)$ of the random plot is computed as follows:

$$R(i) = R(i-1) + Unit^{3d}(i) = \sum_{k=1}^{i} Unit^{3d}(k) \quad (1)$$

Figure 5 shows the direction of the random plot for each 2-mer read. Since the first 2-mer read 'AA' is on the x-axis (+2), it can be confirmed from figure (a) that the positive x-axis moves from the origin O. Since the next 2-mer read is 'AT', a movement in the z-axis by (-2) can be confirmed.

This vector transformation rule are determined empirically in order to discriminate different sequences effectively. As Figure 6, similar sequences are likely to produce similar walk plots.

In this way, the transformed random plot is visualized in an appropriate sized three-dimensional grid. The default grid size $500 \times 500 \times 500$ is what we empirically figured out at which this trade off between speed and correctness of comparison is well balanced for the sequences used in the experiments.

In case of the short genome sequence, it can be represented in a $500 \times 500 \times 500$ grid easily. But the large size sequence needs space normalization to visualize the random plot in limited space. When the vectors of the random plot are calculated, the points that are farthest from the origin $O(0, 0, 0)$ to the X, Y, and Z axes are max_x, max_y, max_z, and the view size of visualization is V, the normalized ith point $R(i) = (X_i, Y_i, Z_i)$ can be expressed as:

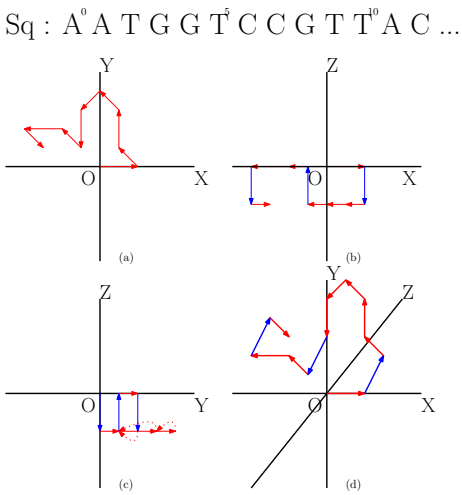


Figure 5: Movement of the random plot for each 2-mer read. (a), (b), (c) and (d) show plots in the form of walks in the X-Y, X-Z, and Y-Z planes in three-dimensional space. From $O(0,0,0)$, the random plot proceeds in accordance with the base assigned to 2-mer. The red random plot represents movement on the X-Y plane, and the blue random plot represents movement on the Z axis.

$$Regular(R(i)) = (X_i \cdot \frac{V}{max_x}, Y_i \cdot \frac{V}{max_y}, Z_i \cdot \frac{V}{max_z}) \quad (2)$$

This visualization model is so useful to compare the huge whole genome. Figure 6 shows advantage of this works[15]. We have constructed the 3D random plots from two whole genomes such as Human Chromosome 1 and Chimpanzee Chromosome 1. In Figure 6, red random plot represents the Human and green one represents the Chimpanzee. Red random plots are up in the positive direction of the X and Y-axis than the green one. This visualization method directly make us to confirm that two genomes are quite similar and the Human chromosome has more 'G' and 'A' base compared to Chimpanzee.

3.3 Vector Extraction from Random Plot

For G, a genome sequence consisting of 4 DNA bases { a, g, t, c }, $ranwalk(G)$ represents a three-dimensional geometric object constructed by our proposed algorithm. Therefore, $ranwalk(G_i)$ consists of a list of linked pixels as follows:

Definition 1.

$$ranwalk(G) =< P_1, P_2, \ldots, P_l >$$

The position of a $ranwalk$ pixel is denoted $P_i = (x_i, y_i, z_i)$ satisfying $|x_i - x_{i+1}| \leq 1$, $|y_i - y_{i+1}| \leq 1$ and $|z_i - z_{i+1}| \leq 1$, which means two pixels P_i and P_{i+1}

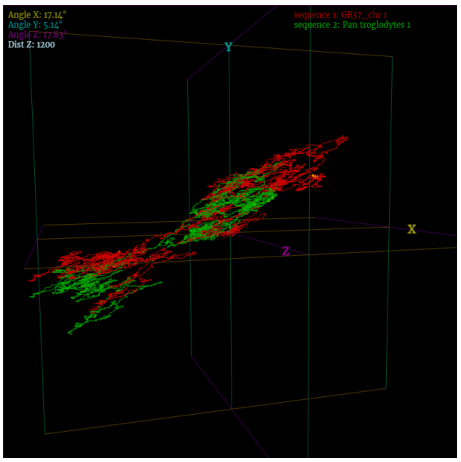


Figure 6: Visualization result of Human and Chimpanzee chromosome 1. Red plot is constructed from Human chromosome 1 and the green random plot is constructed from the whole genome of Chimpanzee (Pan troglodytes) chromosome 1.

are adjacent to each other, sharing a common face. We say P_i and P_{i+1} are 'adjacent' if they are within a distance of 1.

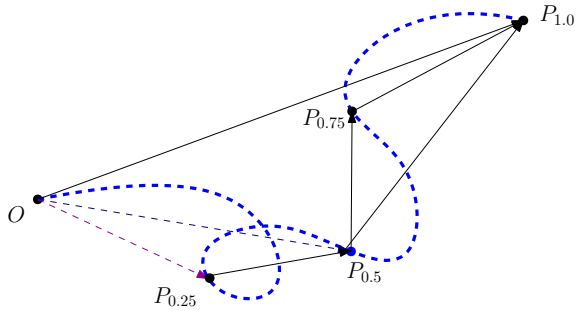


Figure 7: A geometric random plot (blue dotted line) and corresponding vectors.

Now, we explain how to compute the distance between two $ranplot$ pixels obtained from two genomes G_a and G_b to be compared. Assume that we constructed two geometric objects, $R_a = ranplot(G_a)$ and $R_b = ranplot(G_b)$. The proposed distance measure, *random plot distance* ($Rdist$), is a vector with two components $\Delta Span$ and $\Delta Degree$. The proposed $Rdist()$ measure has another parameter, depth k. The distance between two random plot R_a and R_b at depth k is defined recursively as follows. In this definition, R_a1 is the first half of R_a, and R_a2 is the last half of R_a. R_{b1} and R_{b2} are defined in a similar manner. Thus, $R_a = R_{a1} \odot R_{a_2}$, where $\odot$ denotes the geometric concatenation operation.

Definition 2.

$$Rdist(R_a, R_b, k) = Rdist(R_{a1}, R_{b1}, k+1) + Rdist(R_{a2}, R_{b2}, k+1)$$

Now, we explain how to compute $Rdist(R_a, R_b, k = 1)$ at the basic depth = 1 level. In Figure 7, the thick blue

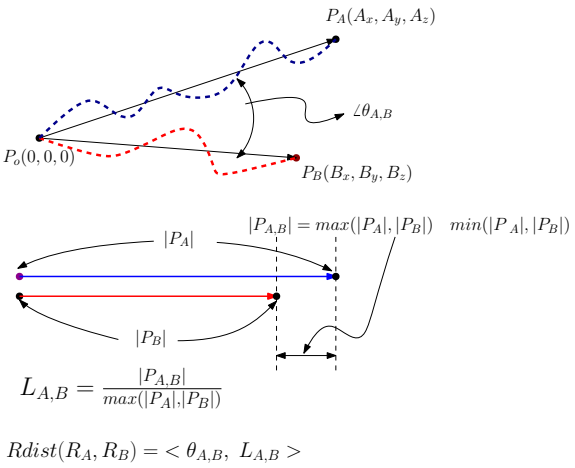


Figure 8: Two comparison parameters $\{\theta_{AB}, L_{AB}\ \}$.

dotted curve represents the random plot for a genome sequence. Symbols $P_0(O)$ and P_1 denote the first and last pixel of a random plot, respectively. P_t denotes the first t-percentile pixel. Thus, $P_{0.5}$ denotes the exact middle pixel in the list of pixels generated by our transformation algorithm.

For an interval in a random walk, we obtain a parameter, the length of the direction vector (P_0, P_1). If two random walks to be compared start with the origin $(0, 0, 0)$, then we can obtain the lengths of two direction vectors from R_a and R_b and compute the angle difference between two vectors Pa_1 and Pb_1.

Assume the start and end points of R_a are P_{a0}, P_{a1}, and those of R_b are P_{b0}, P_{b1}. If $k = 1$ is, the comparison target is $\overrightarrow{P_{a0}P_{a1}}$ and $\overrightarrow{P_{b0}P_{b1}}$. If $k = 2$, further down one step,divided into two vectors are compared both front and rear vector. Therefore, the comparison target are $\overrightarrow{P_{a0}P_{a0.5}}$ and $\overrightarrow{P_{b0}P_{b0.5}}$, $\overrightarrow{P_{a0.5}P_{a1.0}}$ and $\overrightarrow{P_{b0.5}P_{b1.0}}$. If $k = 3$, by applying the same method, it performs a comparison of eight times (2^k).

If the length of divided vector drops below the appropriate length D, the recursion is aborted. In this paper, the threshold D value is set to 100 times the unit size, where unit size is the number of bp per pixel when visualized. The D value was determined experimentally because at least the length of the vector was more than 100px, meaningful comparison was possible.

3.4 Computing Similarity and Search on Random Plot

$Rdist$ refers to the similarity distance between the two vectors. Figure 8 shows that two parameters of $\theta_{A,B}, L_{A,B}$ for $Rdist$. $\theta_{A,B}$ refers to the angle between the two vectors, and $L_{A,B}$ refers to the ratio between the length of two vectors differ and from those of the longer vector. If the two vectors have the same orientation, $\theta_{A,B} = 0$, two vectors, if the length is equal to $L_{A,B} =$ is $0(0 \leq \theta_{A,B} \leq 180, 0 \leq L_{A,B} \leq 1)$.

Algorithm 2 Comparison Algorithm

```
initialize beg ← 0
initialize end ← len(R_a)
initialize O ← {0, 0, 0}
initialize D ← threshold lenth of vector
procedure SIM(beg, end : index of vector list, R_a, R_b :
random plot of G_a, G_b, threshold θ_s, L_s)
    mid ← (end − beg)/2 + beg
    cnt ← 0
    if end − beg > D then
        cnt+ = Sim(beg, mid, R_a, R_b)
        cnt+ = Sim(mid + 1, end, R_a, R_b)
    else
        V_a ← R_a[end] − R_a[beg]
        V_b ← R_b[end] − R_b[beg]
        Len_a ← euclideanDist(O, V_a)
        Len_b ← euclideanDist(O, V_b)
        θ_{a,b} ← acos(dotProduct(V_a, V_b) / (Len_a × Len_b)) × 180
        _{a,b} ← abs(Len_a − Len_b) / max(Len_a, Len_b)
        if θ_{a,b} ≤ θ_s and _{a,b} ≤ L_s then
            return 1
        else
            return 0
        end if
    end if
    return cnt
end procedure
```

To compare and visualize the random plot in a limited space, compression is necessary, as described earlier formula 2. However, in the case of the reference sequence, to calculate the overall similarity of the two vectors, it maintains the two normalized values set. One is a normalized value that is used to process the query sequence, and the other is a normalized value of the calculated original reference sequence. When comparing the sequence to search when the use of normalized values of the query, and visualization uses the original normalized value. This is because it can not be an accurate comparison due to the size difference between the reference and the query, the normalized values differ.

After the normalization of the reference sequence and query sequence the normalized according to the normalization value of the query sequence, extend the depth to a predetermined level k to proceed comparison by dividing a random plot as unit size. Compare all the pieces of the vector unit size extracted from the two random plot by $Rdist()$. When processing the results meet the predetermined reference range, the higher the degree of similarity ($\theta_{A,B} \leq \theta_s$ and $L_{A,B} \leq L_s$). The ratio between the number of the unit vectors that meet the conditions and the total number of vector is similarity between two sequences.

3.5 Reference Sequence Slot

If the length of the query is long enough, the sequence information is compressed at an appropriate rate during visualization in a limited space. Therefore, it is possible to perform in the on-memory state by applying the same compression ratio when searching in the reference sequence. However, sequences with short lengths, such as the LTR sequence, are only kilo-bytes in size and remain uncompressed in the visualization process. In this case, vector information becomes large, and query search becomes impossible in on-memory state. In order to compensate for this, when the length of the reference sequence differs by more than 200 times, the reference sequence is divided into an appropriate number of slots to perform the search. A slot is like a window. By reducing the search range by multiple of the query length at a certain point in time, the method described above can be applied even in a case where a search is required at a low compression ratio in a large size sequence.

$$|Slot(Q,R)| = \frac{|ranwalk(R)| - c_0 \cdot |ranwalk(Q)|}{|ranwalk(Q)| \cdot (c_0 - 1)} \quad (3)$$

Equation 3 is the number of slots created when a query and reference sequence are given. Q and R are Query and Reference sequence respectively, and $len(ranwalk(X))$ represents the length of the whole vector information when X sequence is expressed as a random plot. c_0 is a control constant, which is the size of the space in which a vector should be searched when a certain size query vector is given. In this paper, c_0 is set to around 200.0. Since the query may exist at the point where the slot is divided, the boundaries of each slot are overlapped by the length of the query vector. Figure 9 shows that the vector of the reference sequence is divided into slots.

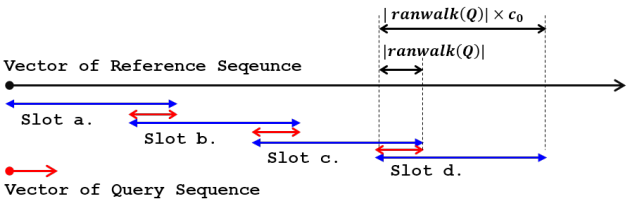


Figure 9: Slot division in reference sequence vector based on the vector length of the query sequence.

4 Experiments

4.1 Dataset Preparation

Actual biological sequence data were used for the searching experiment, and artificial data were used to validate the similarity comparison model. The biological sequences are Human chromosome 1 (246MB size) and the sequence of a 1M-10M size extracted from chromosome 1. Artificial sequence data are obtained by extracting a sequence of 1-10 MB length from the Human chromosome 1 sequence at a random location and inserting noise in a predetermined ratio. A number of bases with different sizes are deleted, inserted, and replaced by a ratio of 1% to 50%. The artificial data information such as ratio and the b.p. size and number of pixels and compression ratio is shown in Tables 3 and 4. 'A1-0' means that the artificial data of 1M size and 0% modified, namely it is just extracted from Human sequence, not modified. But 'A10-25' means that the artificial data of 10M size and 25% modified.

This modification rate is expressed as 'M' (M.Rate) in Table 3 and 4. 'M' (M.Rate) refers to the modified ratio of the number of B.P. on origin sequence. For verification of the similarity comparison model, this rate was set higher gradually as the experiment was repeated.

'Ratio' refers to the compression ratio of the number of B.P. and pixels of the actual sequence to be converted to a random plot. For example, in the Table 3, since A1-1 sequence has 1000.02K bases, and random plot size consists of 36K pixel, the compression ratio is 3.58%. 'Sim' means that the similarity result of origin sequence and modified sequence and 'Comp.t' represents the comparison time.

Table 3: Specification of artificial data of 1M, 2M size extracted from Human chromosome 1 and comparison result

Sq N.	M (%)	Length (K bp)	Plot (K px)	Ratio (%)	Sim. (%)	Cmp.t (s)
A1-0	0	1000.02	36.00	3.58	100.00	0
A1-1	1	999.93	35.79	3.58	99.59	0
A1-2	2	1000.01	36.17	3.62	99.45	0
A1-5	5	999.89	36.67	3.67	98.23	0
A1-8	8	999.97	37.74	3.77	96.06	0
A1-10	10	1000.49	38.05	3.80	91.73	0
A1-15	15	999.78	40.74	4.07	93.58	0.016
A1-20	20	1000.29	42.49	4.25	91.76	0
A1-25	25	999.92	44.2	4.42	86.14	0
A1-30	30	999.79	47.18	4.72	84.23	0.015
A1-40	40	1001.12	50.86	5.08	69.86	0.015
A1-50	50	999.47	58.36	5.84	63.53	0.016
A2-0	0	2000.04	67.09	3.35	100.00	0
A2-1	1	1999.96	66.89	3.34	98.03	0
A2-2	2	2000.15	67.27	3.36	95.85	0
A2-5	5	2000.26	68.99	3.45	94.65	0
A2-8	8	2000.2	70.4	3.52	90.5	0
A2-10	10	2000.14	69.64	3.48	91.2	0.016
A2-15	15	1999.94	70.84	3.54	85.71	0
A2-20	20	2000.18	77.56	3.88	83.62	0
A2-25	25	2000.66	79.97	4.00	72	0
A2-30	30	1999.85	89.15	4.46	73.37	0
A2-40	40	2001.5	88.54	4.42	63.34	0.016
A2-50	50	2000.62	104.11	5.20	54.91	0.016

Tables 5 and 6 are data for searching for LTR sequences that are frequently handled in real bioinformatics analysis. In the table 5, R-F-1 is the reference sequence and means chromosome 1 sequence of the Flatfish. In the corresponding table 6, Q-F-1 is the query sequence of R-F-1 and is the LTR sequence extracted from R-F-1. The biggest difference from the artificially generated data is that the LTR sequence is too short and thus has a low compression rate

Table 4: Specification of artificial data of 4M, 10M size

Sq N.	M (%)	Length (K bp)	Plot (K px)	Ratio (%)	Sim. (%)	Cmp.t (s)
A4-0	0	4000.09	42.62	1.07	100.00	0
A4-1	1	4000.18	42.69	1.07	99.3	0
A4-2	2	3999.71	42.15	1.05	98.93	0
A4-5	5	3999.51	44.13	1.10	98.18	0
A4-8	8	3999.36	44.08	1.10	96.03	0
A4-10	10	4000.1	45.95	1.15	96.27	0
A4-15	15	3999.75	45.69	1.14	94.63	0
A4-20	20	4000.23	49.33	1.23	91.33	0
A4-25	25	3999.7	49.78	1.24	90.93	0
A4-30	30	4001.21	53.79	1.34	84.36	0.016
A4-40	40	3999.59	57.16	1.43	76.82	0.015
A4-50	50	4000.14	64.1	1.60	66.87	0
A10-0	0	10000.05	65.26	0.65	100.00	0
A10-1	1	10000.03	65	0.65	98.08	0
A10-2	2	10000.13	64.81	0.65	97.29	0
A10-5	5	9999.47	66.32	0.66	96.76	0.015
A10-8	8	9999.74	68.75	0.69	95.12	0
A10-10	10	10000.71	67.93	0.68	94.9	0.015
A10-15	15	9999.97	75.13	0.75	91.18	0
A10-20	20	9998.82	74.38	0.74	90.24	0
A10-25	25	9999.4	78.34	0.78	87.68	0.016
A10-30	30	9999.24	82.29	0.82	82.49	0
A10-40	40	9999.82	87.51	0.88	78.48	0
A10-50	50	10001.48	94.45	0.94	66.47	0

in the visualized space. This is because visualization is possible in a limited space without compression. Since the reference sequences are based on the compression ratio of the query sequence, we can see that the random plot size of the reference sequence is very large relatively.

Table 5: Specification of biological data for reference

Sq N.	Chr.	Species	Length (M bp)	Plot (M px)	Ratio (%)
R-F-1	1	Flatfish	19.80	19.02	95.06
R-F-2	2	Flatfish	20.14	19.34	96.02
R-F-3	3	Flatfish	22.24	21.36	96.04
R-F-5	5	Flatfish	23.64	22.69	95.98
R-H-1	1	Human	246.89	236.44	95.77

Table 6: Specification of biological data for query

Sq N.	Chr.	Species	Length (K bp)	Plot (K px)	Ratio (%)
Q-F-1	1	LTR	0.41	0.41	100.00
Q-F-2	2	5'LTR	1.56	1.54	98.72
Q-F-3	3	Gypsy	4.84	4.78	98.76
Q-F-5	5	LTR	8.55	6.44	75.32
Q-H-1	1	HERV-K	9.26	8.06	87.04

4.2 Experiment:Comparison Between Modification ratio and Similarity based proposed Model

Table 3 and Figure 12 show the result of similarity analysis of origin extracted sequence and modified sequences. In Table 3, 'Sim' means that the similarity result of origin sequence and modified sequence and 'Comp.t' represents the comparison time. As the modification ratio increases, the degree of similarity decreases. Thus, it can be confirmed that the similarity comparison model proposed in this study accurately reflects the similarity of the sequences. In addition, except for sequence generation, the time required for comparison is 0.02 seconds, which means that it can be processed at a very high speed.

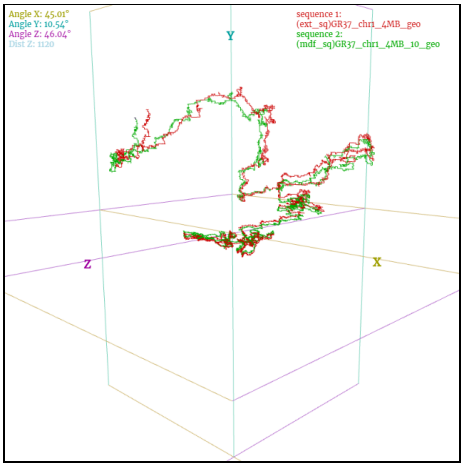


Figure 10: Red random plot represents one part of Human chromosome 1, the length of which is 4 MB, in terms of nucleotide bases. Green random plot represents the 10% distorted sequence of the red one, Human chromosome 1.

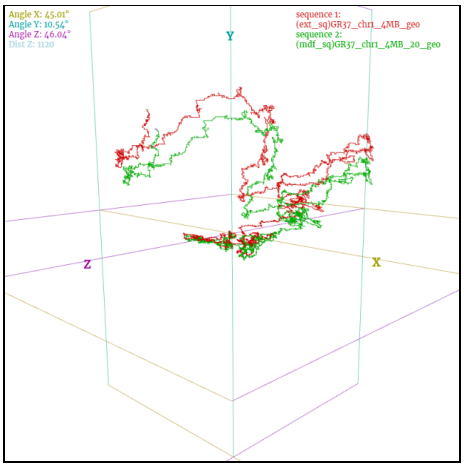


Figure 11: Red random plot represents one part of Human chromosome 1, the length of which is 4 MB, in terms of nucleotide bases. Green random plot represents the 30% distorted sequence of the red one, Human chromosome 1.

4.3 Experiment:Artificial Sequence Search over whole genome sequence

Table 7 is the result of sequence searching process for extracted original sequence from Human chromosome 1 and the modified sequences. 'Unit B. P. ' is the size of B.P. as

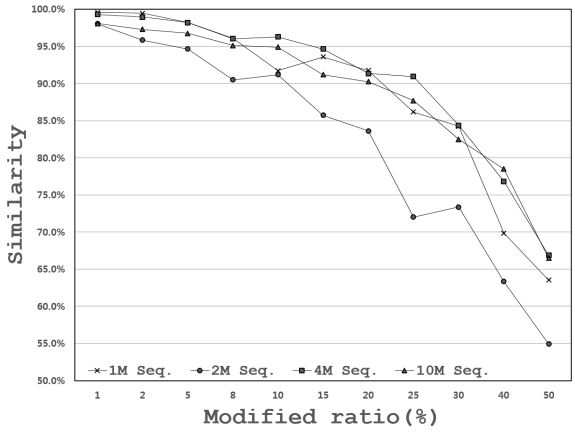


Figure 12: Similarity between origin sequence and modified sequences in each size 1-10MB.

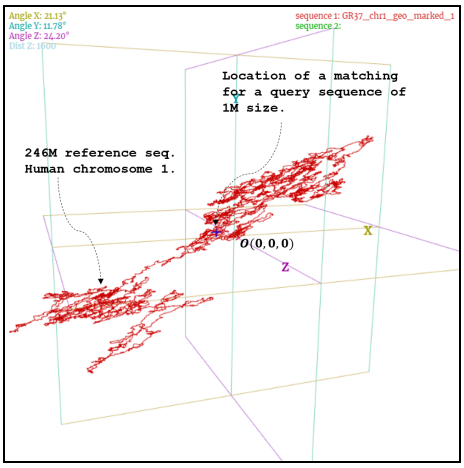


Figure 13: Searching result of query sequence (A1-0) in reference sequence (Human chromosome 1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

a unit of search,' Unit Vector' refers to the size of the vector to consider when comparing a time. 'Error Dist.' is the distance between the actual sequence position and the result of search position. 'Find.t' shows the amount of time spent on search. The original sequence (0% modified sequence) search, as well as about the modified sequence of up to 20% are also searched in a short time. The difference between the actual position and the search result is relatively accurate, as the query size is less than 200 B.P. when the query size is 1M, and only about 2000 B.P. when the query is 10M. Figure 13 and 14 are the visualization

Table 7: The result of sequence search for origin sequence and modified sequence in Human chromosome 1

Q sq.	Unit sz. (bp)	Vec.sz (px)	error Dist.	Sim. (%)	Find.t (s)
A1-0	28	11200	0	99.29	17.269
A1-5	27	10800	150	97.27	21.341
A1-10	26	10400	840	91.34	23.213
A1-20	23	9200	120	88.75	22.514
A4-0	92	36800	1160	92.81	6.537
A4-5	90	36000	160	98.41	6.896
A4-10	88	35200	1040	92.68	7.678
A4-20	80	32000	1040	86.3	9.132
A10-0	154	61600	1120	93.88	13.665
A10-5	150	60000	560	97.21	16.065
A10-10	148	59200	280	95.09	14.245
A10-20	134	53600	2020	81.95	22.241

result of search for the query sequence of 1MB, 10MB in the chromosome 1 of the Human. Red random plot is a visualization of Human chromosome 1, and blue point is the location where the query was searched. Through the visualization results, we can see that a query of 1MB size was found at a relatively early stage of the reference sequence, and a query of 10MB size was at the end of the sequence. This is consistent with the position in the actual sequence, and represents a search result in a more intuitive.

4.4 Experiment:Biological Sequence Search over whole genome sequence

Table 8 shows the results of searching a biological query sequence in a whole genome sequence. The search for the LTR sequence (Q-F-1) extracted from the flatfish chromosome 1 resulted in a similarity of 85.7% within 90 B.P. of the actual query position within about 0.4 seconds of search time. On the other hand, the HER-V sequence (Q-H-1) extracted from Human chromosome 1 took relatively longer time, longer than 40 seconds because the length of the query sequence was short and the length of the reference sequence was long. The difference between the actual position and the search result is about 2000 B.P., which is relatively accurate considering that the length of the reference sequence is more than 200M.

Figures 15,16,17 and 18 visualize the flatfish chromosome 1,2,3,5 sequences, respectively. The red one is a visualization of the whole genome of a flatfish, and the area marked in blue is where each query was searched. Figures 17 and 18 show that the marked positions are almost identical to the origin, reflecting that the Q-F-3 and Q-F-5 queries are actually located within 0.5 % of the flatfish whole genome sequence. On the other hand, Figures 15 and 16 reflect that the marked positions are relatively far away from the origin, that the positions of the Q-F-1 and Q-F-2 queries are actually located within 7% and 10% of the flatfish whole genome sequence. Figure 19 visualizes the Human chromosome 1 sequence and marks the result of searching the Q-H-1 query. It is well reflected that the Q-H-1 query is actually located in the early 63 % (about 155 MB.P.) of the Human sequence. Figure 20 is the result of original query sequence (Q-H-1) and enlarged subsequence of the reference sequence (R-H-1) at searched position. The similarity of the searched sequence in the reference (green plot) was 78%, and it can be confirmed that

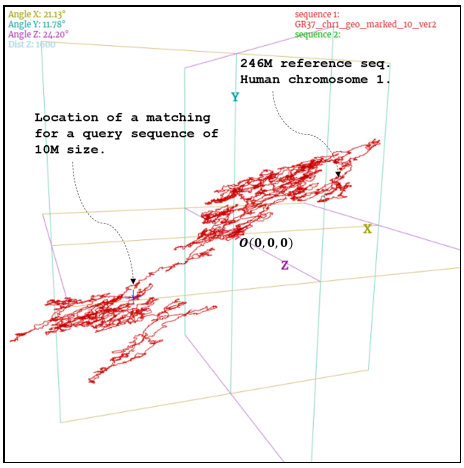


Figure 14: Searching result of query sequence (A10-0) in reference sequence (Human chromosome 1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

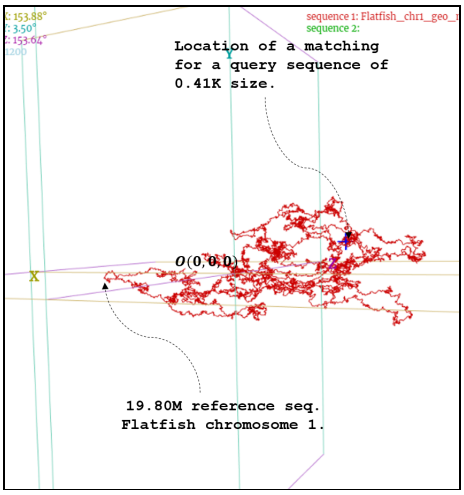


Figure 15: Searching result of query sequence (Q-F-1) in reference sequence (R-F-1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

the query is very similar to the query when matched with the query sequence.

Table 8: The result of sequence search for biological query sequence in flatfish and Human chromosome 1.

Q sq.	Unit sz. (bp)	Vec.sz (px)	error Dist.	Sim. (%)	Find.t (s)
Q-F-1	1	413	90	85.70	0.400
Q-F-2	1	1540	180	72.40	1.030
Q-F-3	1	4780	960	69.10	0.452
Q-F-5	1	6443	1230	75.20	2.038
Q-H-1	1	8063	2130	78.40	41.011

5 Conclusion

Most genome sequence analyses proceed through comparative analysis by finding similar sequence data. Therefore, there is a need for a technique to quickly compare and search for large amounts of sequence data. The alignment technique is a very accurate method to compare sequences, but its high time and space complexity is inadequate to handle large sequences. To overcome these disadvantage, we suggest a new method for comparison and finding for Mega size sequence. Converts the genome sequence as a random plot on the three-dimensional, followed by replacing the sequence comparison problem with geometric object comparison problem. As a result of experiments, similarity precessed by our comparison model accurately reflects the modified ratio between the modified sequence and the original sequence. Most analytical studies based on visualization derive only a single result because they derive a numerical value based on the final result of the visualization. The search and comparison method based on the sequence visualization proposed in this study has high value of utilization of information because all compressed partial visualization information is used for searching sequence. It is useful in that the partial similarity of the sequence can be measured. In addition, a query sequence of size 1-10M was searched in a whole genome sequence of 200M or more, and a relatively precise position was found for the original sequence as well as the modified sequence up to 20%. Also the search time 25 seconds or less, was confirmed handled in a very improved speed compared to the alignment algorithm.

On the other hand, when a sequence with a shorter kilobyte unit length is used as a query, such as an LTR sequence, the compression rate is lowered at the time of visualization, resulting in a lower compression rate of the reference sequence, which leads to a longer search time. However, considering the length of the reference, we can confirm that the position searched is relatively accurate.

The proposed alignment-free searching method is very fast and effective to find a long query sequence over the whole genomes whose size is more than multi-hundreds mega-bytes. It was able to compare and search the sequence at a much improved rate than the alignment-based model by modifying the sequence data into a three-dimensional random plot object and comparing the similarity with the compressed information. Searching algorithm based on alignment method is popular and works good biological sequence comparison but if the size of query and target reference is very large (more than 100 mega bases) the alignment base algorithm requires huge memory space and takes a long computation time. Though our algorithm can't locates the position of query sequence exactly by the DNA base unit, but we can use this procedure as one preprocessing step to find query sequence.

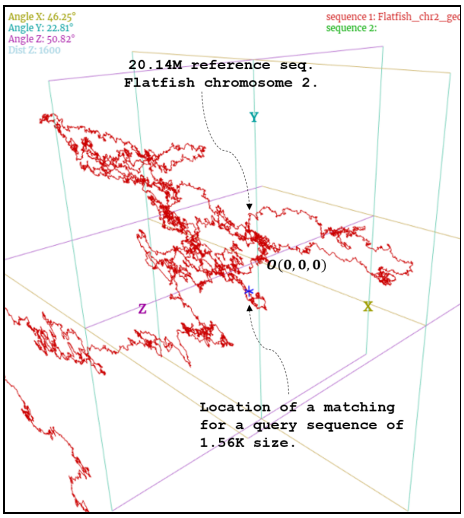


Figure 16: Searching result of query sequence (Q-F-2) in reference sequence (R-F-2). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

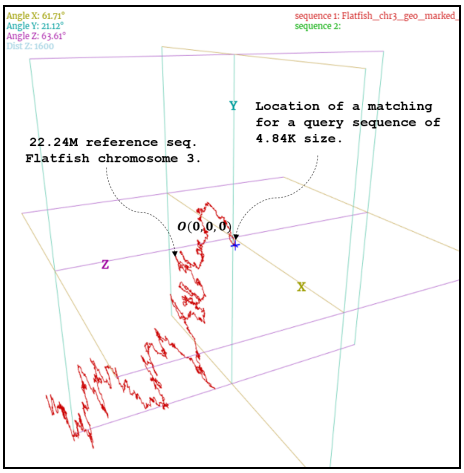


Figure 17: Searching result of query sequence (Q-F-3) in reference sequence (R-F-3). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

Acknowledgement

This research was supported by the Collaborative Genome Program of the Korea Institute of Marine Science and Technology Promotion (KIMST) funded by the Ministry of Oceans and Fisheries (MOF) (No. 20140428).

References

[1] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990. https://doi.org/10.1016/S0022-2836(05)80360-2.

[2] Susana Vinga and Jonas Almeida. Alignment-free sequence comparison—a review. *Bioinformatics*, 19(4):513–523, 2003. https://doi.org/10.1093/bioinformatics/btg005.

[3] Tetsuya Nakamura, Keishi Taki, Hiroki Nomiya, Kazuhiro Seki, and Kuniaki Uehara. A shape-based similarity measure for time series data with ensemble learning. *Pattern Analysis and Applications*, 16(4):535–548, 2013. https://doi.org/10.1007/s10044-011-0262-6.

[4] Milan Randić, Marjan Vračko, Jure Zupan, and Marjana Novič. Compact 2-d graphical representation of dna. *Chemical physics letters*, 373(5):558–562, 2003. https://doi.org/10.1016/S0009-2614(03)00639-0.

[5] Yongfan Li, Guohua Huang, Bo Liao, and Zanbo Liu. H-l curve: a novel 2d graphical representation of protein sequences. *MATCH-COMMUNICATIONS IN MATHEMATICAL AND IN COMPUTER CHEMISTRY*, 61(2):519–532, 2009. https://doi.org/10.1016/j.cplett.2008.07.046.

[6] Milan Randić. Graphical representations of dna as 2-d map. *Chemical Physics Letters*, 386(4):468–471, 2004. https://doi.org/10.1016/j.cplett.2004.01.088.

[7] Yonghui Wu, Alan Wee-Chung Liew, Hong Yan, and Mengsu Yang. Db-curve: a novel 2d method of dna sequence visualization and representation. *Chemical Physics Letters*, 367(1):170–176, 2003. https://doi.org/10.1016/S0009-2614(02)01684-6.

[8] Milan Randić, Nella Lerš, Dejan Plavšić, Subhash C Basak, and Alexandru T Balaban. Four-color map representation of dna or rna sequences and their numerical characterization. *Chemical physics letters*, 407(1):205–208, 2005. https://doi.org/10.1016/j.cplett.2005.03.086.

[9] Martin Krzywinski, Jacqueline Schein, Inanc Birol, Joseph Connors, Randy Gascoyne, Doug Horsman, Steven J Jones, and Marco A Marra. Circos: an information aesthetic for comparative genomics. *Genome research*, 19(9):1639–1645, 2009. https://doi.org/10.1101/gr.092759.109.

[10] Jiyuan An, John Lai, Atul Sajjanhar, Jyotsna Batra, Chenwei Wang, and Colleen C Nelson. J-circos: an interactive circos plotter. *Bioinformatics*, 31(9):1463–1465, 2015. https://doi.org/10.1161/CIRCULATIONAHA.115.015220.

[11] Bo Liao and Kequan Ding. A 3d graphical representation of dna sequences and its application. *Theoreti-*

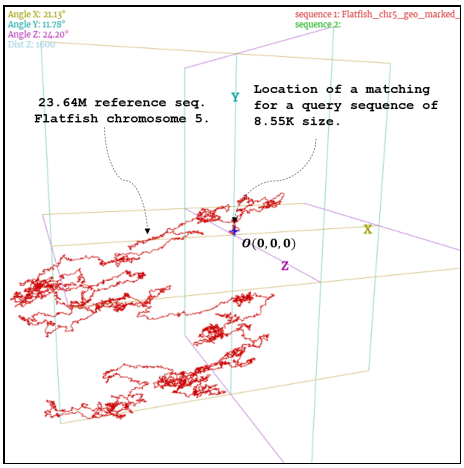


Figure 18: Searching result of query sequence (Q-F-5) in reference sequence (R-F-5). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

cal Computer Science, 358(1):56–64, 2006. https://doi.org/10.1016/j.tcs.2005.12.012.

[12] Alexey Pasechnik, Aleksandr Mylläri, Tapio Salakoski, A Mylläri, T Salakoski, and T Salakoski. Dynamical visualization of the dna sequence and its nucleotide content. *Proceedings of KRBIO*, 5:47–50, 2005.

[13] Min-Ah Kim, Eun-Jeong Lee, Hwan-Gue Cho, and Kie-Jung Park. A visualization technique for dna walk plot using k-convex hull. *Journal of WSCG*, 5(1-3):212–221, 1997.

[14] Daegeon Kwon. Whole genome data visualization and analysis using 3d random walk plot. Master's thesis, Pusan National University, 2015.

[15] Lee Da-Young, Kim Kyung-Rim, Kim Taeyong, and Cho Hwan-Gue. Comparison-specialized visualization model for whole genome sequences. *Journal of WSCG*, 24(2):43–52, 2016.

[16] Hwan-gue Cho Dayoung Lee, Daegeon Kwon. WebGL based Visualization System for Whole Genomes. In *Proceedings of KIISE*, pages 1414–1416. KOREA INFORMATION SCIENCE SOCIETY, 2016.

[17] Milan Randić, Jure Zupan, Dražen Vikić-Topić, and Dejan Plavšić. A novel unexpected use of a graphical representation of dna: Graphical alignment of dna sequences. *Chemical Physics Letters*, 431(4):375–379, 2006. https://doi.org/10.1016/j.cplett.2006.09.044.

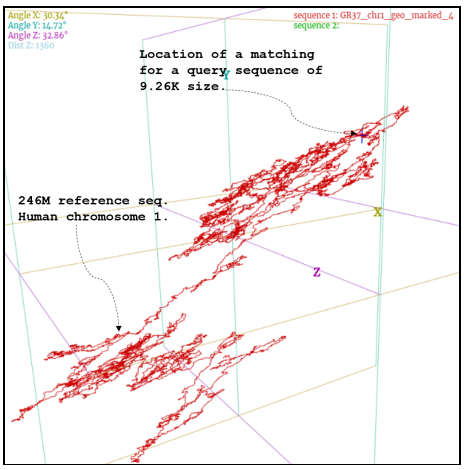


Figure 19: Searching result of query sequence (Q-H-1) in reference sequence (R-H-1). Red plot represents reference sequence and blue cross point represents the position of searched query sequence.

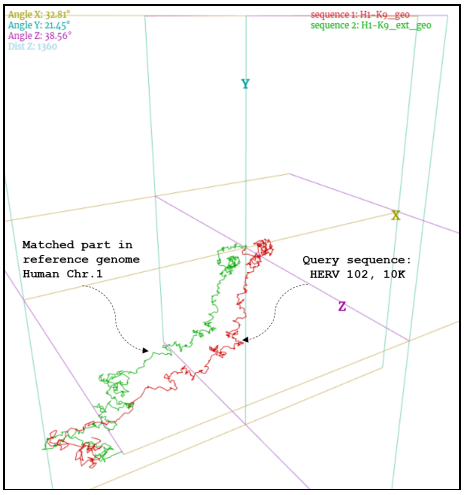


Figure 20: Matching result between the query sequence (Q-H-1) and the extended subsequence of reference sequence (R-H-1), which was depicted as a blue cross in Figure 19.

Cancelable Fingerprint Features Using Chaff Points Encapsulation

Mokhled S. Al-Tarawneh
Computer Engineering Department, Faculty of Engineering, Mutah University, B.O.Box (7), Mutah 61710, Jordan
E-mail: mokhled@mutah.edu.jo

Keywords: fingerprint, feature extraction, minutiae, cancelable, encapsulation, chaff points

Received: September 20, 2017

Recently, biometrics imaging is widely used in several security areas such as security monitoring, database access, border control and immigration, and for reliable personal verification, identification and recognition schemes. To determine or confirm the identity of an individual's based on their physiological and/or behavioral characteristics, biometric features must be used. The aim of this paper is to review cancelable biometric generation and protection schemes. An approach for generating chaff points for fingerprint template features encapsulation as fingerprint cancelability infrastructure has been presented. Results show that strong positive correlation of original minutiae scores go with high decapsulated minutiae scores. To test the given cancelable approach performance two indexes are used, FAR (false accept rate) and FRR (false reject rate).

Povzetek: Razvita je nova biometrična metoda za prepoznavanje prstnih odtisov, temelječa na računalniškem algoritmu.

1 Introduction

Biometrics increasingly forms the basis of identification and recognition across many sensitive applications[1]. Biometrics is statistical analysis of people's physical and behavioral characteristics, It is more convenient for users, reduces fraud and is more secure. Fingerprint is commonly used modality compared to traditional identification and verification methods, such as plastic identification card, or traditional passwords [2]. Fingerprint authentication has two phases, enrolment and authentication (or verification). Enrolment involves measuring an individual's biometric data to construct a template for storage. Authentication involves a measurement of the same data and comparison with the stored template [3]. The core of any biometric system is the extracted template, where the matcher algorithms in this systems depends on template matching in one to one (verification) and one to many (identification) modes. It has become critical to protect fingerprint templates in the widespread biometric community. One way for doing this is using cancelable techniques, which transforms original templates in a non-invertible way and uses those transformed templates to verify a person's identity. Securing a stored fingerprint template and image is of paramount importance because a compromised fingerprint cannot be easily revoked. That why fingerprint template should be protected, where an ideal biometric template protection scheme should possess the following four properties [2]. 1) Diversity: if a revoked template is replaced by a new model, it should not correspond with the former. This property ensures the privacy. 2) Revocability: It should be possible to revoke a compromised template and replace it with a new one based on the same biometric data. 3) Security: It must be computationally hard to obtain the original template from the protected template. This property prevents an adversary from creating a physical spoof of the biometric trait from a stolen template. 4) Performance: The biometric template protection scheme should not degrade the recognition performance, false acceptance rate (FAR) and false rejection rate (FRR) of the biometric system[4]. Due to some biometric vulnerabilities, lack of security because it is impossible to revoke biometric unlike password or token, and therefore if biometric is leaked out once and threat of forgery has occurred, the user cannot securely use his biometric anymore. The only remedy is to replace the template with another biometric feature. However, a person has only a limited number of biometric features [5]. In order to overcome the vulnerabilities of biometric systems, both biometrics and crypto research communities have addressed some of the challenges, one of them is cancelable biometric which gained a lot of interest in recent years [6]. The concept behind the cancelable biometrics or cancelability is a transformation of a biometric data or extracted feature into an alternative form, which cannot be used by the imposter or intruder easily, and can be revoked if compromised. This paper proposed a cancelability method based on chaff point encapsulation to cope with biometric drawbacks. The method was tested according to performance evaluation factors.

2 Related works

Cancelable biometric generation has gained a lot of interest in recent years, and it is studied from different point of views, it could be categorized as:

1- Biometric Crypto Systems, this approach is used key binding or key generation schemes, where key binding is a user specific key or a helper data which is independent to the biometric data,

while key generation is generating the helper data from the biometric data using specific notations of crypto systems[7] [8] [9] [10].

2- Biometric Transformations: This approach is based on the transformations of biometric features, where it is categorized into two ways: Bio-Hashing which is used an external key source (PIN or Password) and other functional parameter representation to generate Hash value of the biometric data, it stores the Hash value alone in the data base [11] [12] [13] and Non-invertible transformation [14] [15], such that no information can be revealed from the cancelable biometrics template, which is stored in databases for personal identification/verification, or using biometric data to transform its cancellable domain by polynomial functions and co-occurrence matrices[16].

The proposed method will use encapsulation techniques to protect biometric template. Thus, cancelable template can be attained by template chaff point's encapsulation, where the principal objectives of cancellable biometrics templates can be checked, such as diversity, cancelability, reusability, non-invertability, and performance of technique.

3 Fingerprint feature extraction

The information carrying features in a fingerprint are the line structures, called ridges and valleys[17]. Figure 1, the ridges are black and the valleys are white. It is possible to identify two levels of detail in a fingerprint. Based on carried ridge and valleys minutiae points could be extracted. The minutiae provide the details of the ridge-valley structures, like ridge-endings and bifurcations. Minutiae are subject to post- processing to verify the validity of that are extracted using standard minutiae extraction algorithms. In this study the needed information to be extracted are minutiae coordination's (x, y), type of minutiae (ridge ending or bifurcation), and orientation. Table 1, shows some extracted samples from FVC2004, DB1_B database.

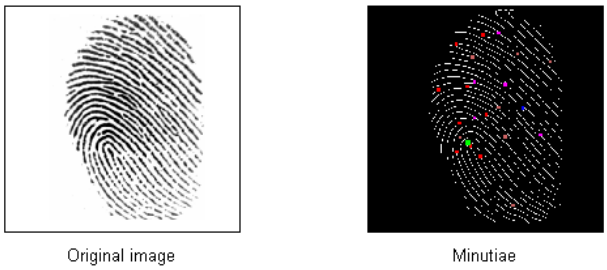


Figure 1: Minutiae-based Fingerprint Extraction.

Due to the importance of extracted fingerprint features (minutiae) and it is criticality as a major step in designing a secure biometric system. The protection of feature templates of the users those are stored either in a central database or on smart cards. If it is compromised, it leads to serious security and privacy threats, it is not possible for a legitimate user to revoke his biometric identifiers and switch to another set of uncompromised identifiers, that why we were looking for a technique to protect this extracted temples, encapsulation technique could solve previous problems. A FVC2002 database[18] with best extraction algorithm based on high scores on distributions, acceptance and rejection rates was chosen to be based for cancelable encapsulation algorithm. For accurate algorithms in extracting minutiae features for creating encapsulation cancelable based system, a comparison result of performance evaluation according to values of False acceptance rate (FAR), False rejection rate (FRR) and Error equal rate (EER) was explored, Table 1, all comparison algorithms took coordination, type and orientation as parameters for extracted features.

FVC2002,DB1_1,101_1				FVC2002,DB1_1,107_1			
X	Y	Type	Orient	X	Y	Type	Orient
216	46	3	0.5030	254	38	1	0.7503
190	49	1	3.5827	218	58	3	0.5566
146	64	1	3.2684	160	68	3	2.6141
247	80	1	0.7002	187	74	1	6.1710
173	86	1	0.3665	155	79	1	5.5414
302	93	1	0.8371	162	87	1	2.3393
176	127	3	0.2761	140	130	1	4.9955
227	131	3	0.5634	107	138	3	5.1562
164	135	1	3.3159	156	139	1	1.8174
117	140	1	5.7642	245	139	1	1.0242
196	181	1	3.7386	195	140	3	0.4140
176	187	3	0.4612	195	151	3	3.7130
151	195	1	5.7175	225	156	3	0.9941
285	215	3	0.7886	196	165	1	4.5301
227	218	1	0.8160	151	186	1	4.7262
152	219	1	2.2884	295	188	3	0.9923
169	233	1	4.1407	135	200	3	4.7436
147	242	1	4.6064	241	218	1	4.1310
186	250	1	4.1676	287	239	3	0.7131

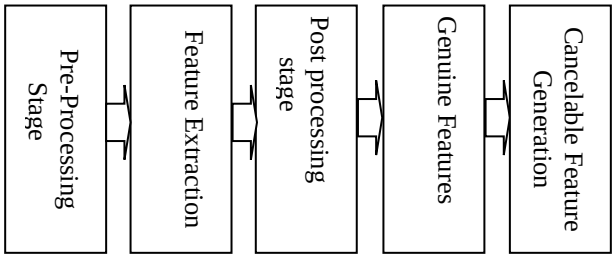
Table 1: Extracted minutiae points with data(x, y, t, Φ).

That why minutiae extraction points with previous references (x, y, t, Φ) was taken as a fundamental step for proposed framework and future method of cancelability.

4 Proposed framework

A novel method is proposed in this section. It is name as encapsulation protection method. It includes the building blocks of phases such as preprocessing, minutiae extraction, post processing and cancelable and irrevocable template generation. The proposed method uses fingerprint biometric to generate cancelable template. The system level design of the proposed method is given in figure 2.

Figure 2: System level design for fingerprint cancelable template generation.



In preprocessing stage a feeding input is the original fingerprint image taken from database DB1_1 [18], where automatic cropping technique was applied based on image background to detect the region of interest (ROI) of target image. ROI image was given to enhancement step as a part of pre-processing stage because the quality of fingerprint structure (ridge, valley) is an important characteristic. An enhancement technique applied in pre-processing phases as normalization, ridge segmentation, structure orientation estimation, frequency enhancement estimation and thinning to get binarization image which is pre extracting feature identification figure 3. After binarization and thinning process, a Cross Number algorithm (CN) described in [19, 20] was applied to get minutiae extraction. The CN algorithm is working on pixel representation to detect all minutiae, while the false minutiae can be eliminated at the post-processing stage by validating algorithm to get only genuine features.

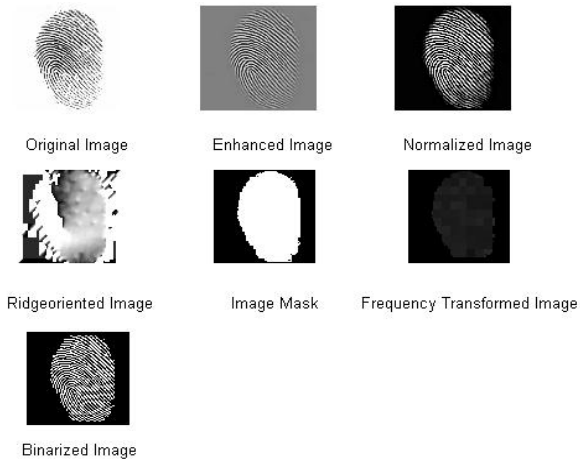


Figure 3: A result of proposed frame work, original, enhanced, normalized, filtered and binarized images.

5 Cancelable feature generation

The basic idea of cancelable feature generation as encapsulation method is to compute encapsulation chaff points (ECP) based on original extracted minutiae, where it used to recover the enrolled template on transmission stage, as well matching on the same stage. Pseudo-code of ECP is given in Algorithm 1:

Algorithm 1: Encapsulation method based on cancelable feature generation.

Input Extracted minutiae template with (x,y) coordinates, T-type of minutiae {3 bifurcation, 1 ridge ending}, Φ-orientation, m number of minutiae, X(x, y, T).

Step 1: Perform chaff points

For k=1: m

Y=change X(x → y, y → x, T=T+1)

End for

Step 2: Mix new chaff point with original minutiae

Z=(X, Y) concatenate

Output Z(x,y,T)

End Algorithm

A representation of original extracted minutiae for FVC2002, DB1_1,101_1 from table1 shown in figure 4.

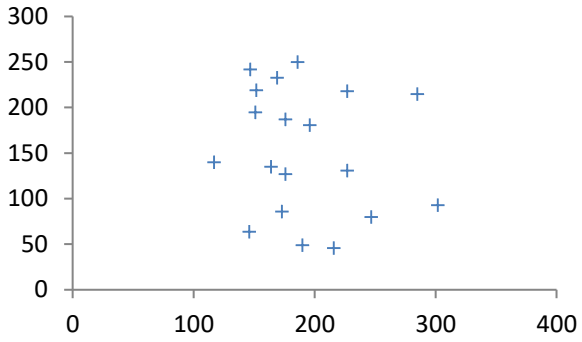


Figure 4: Original extracted minutiae representation.

Applying this algorithm on FVC2002, DB1_1,101_1 from table 1 will give figure 5. a chaff points, while a mixing encapsulation result will be shown in figure 6.

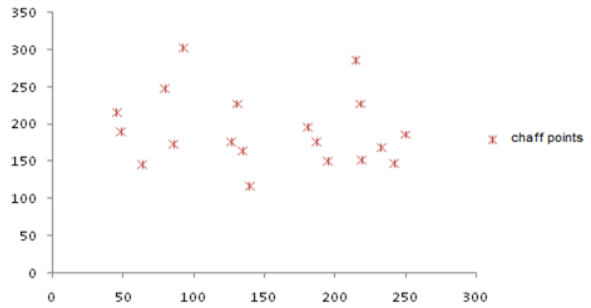


Figure 5: Chaff points representation.

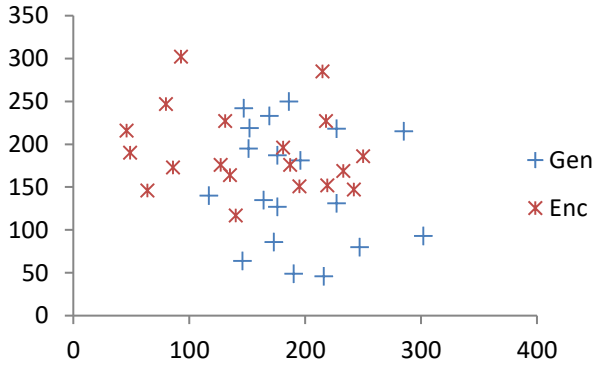


Figure 6: Mixing encapsulation of original minutiae and chaff points representation.

A Decapsulation part of proposed frame work is used to open up transmitted encapsulated data, separate faked chaff points from original minutiae points. The following algorithm explaining the procedure of computing decasulation chaff points (DCP), Pseudo-code of DCP is given in Algorithm 2:

Algorithm 2: Decapsulation method to wrap up genuine minutiae points.

Input Encapsulated template with (x, y) coordinates, T-type of minutiae {3 bifurcation, 1 ridge ending, 2 and 4 fakes}, Φ-orientation, m number of minutiae, X(x, y, T).

Step 1: Read transmitted encapsulated template

X= Find fake chaff points

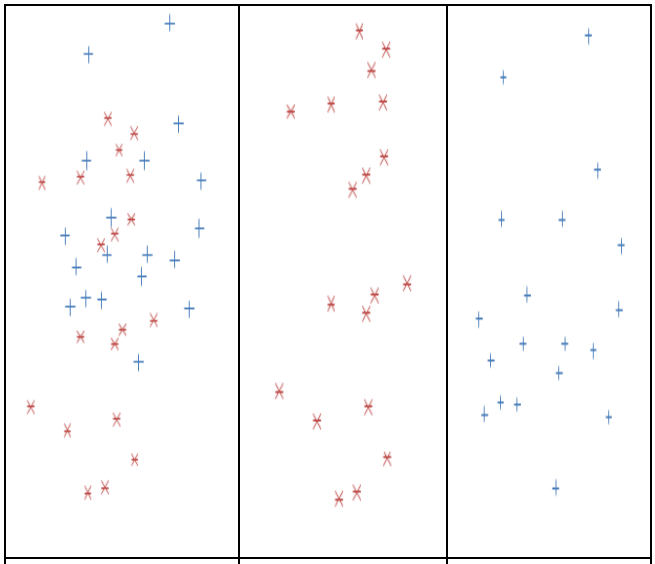


Figure 7: Decapsulation method to wrap up genuine minutiae points.

Step 2: Y= divide a template on base of chaff point with their types

Z=(X, Y) separate

Output Z(x,y,T)

End Algorithm

A representation of this process is shown in figure 7.

6 Experimental study

An empirical study is performed to test the cancelability and irrevocability of the proposed method using linear correlation test of general original clear minutiae with decapsulated minutiae scores, the strength and nature of the linear relationship between two scores of clear and decapsulated minutiae. Applying linear coefficient (R) formula on given results, the value of R is found to be 0.9999. This is a strong positive correlation, which means that high original minutiae scores go with high decapsulated minutiae scores (and vice versa) figure 8. Another test was done to check the performance of proposed method; it was evaluated by calculating false acceptance rate (FAR) as well false reject rate (FRR) for scenario, original extracted and decapsulated templates. Sequence of experiments is made on the proposed method using benchmark databases such as FVC (Fingerprint Verification Contest) in 2002, 2004 figure 9, figure 10.

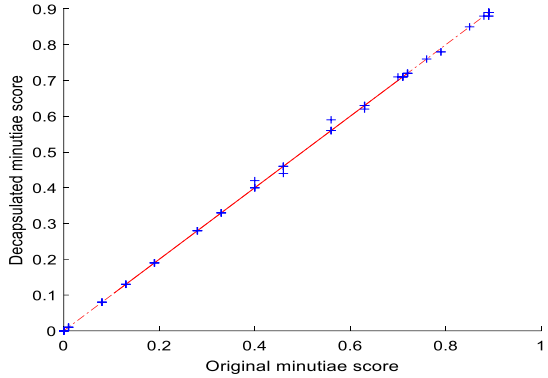


Figure 8: A correlation scores of original minutiae and chaff points representation.

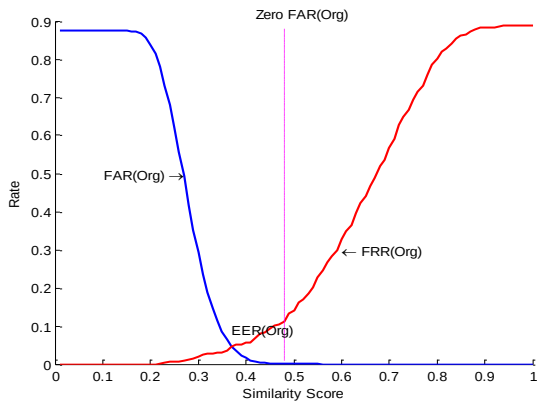


Figure 9: FAR/FRR of the dual fingerprint matcher that employs original minutiae template.

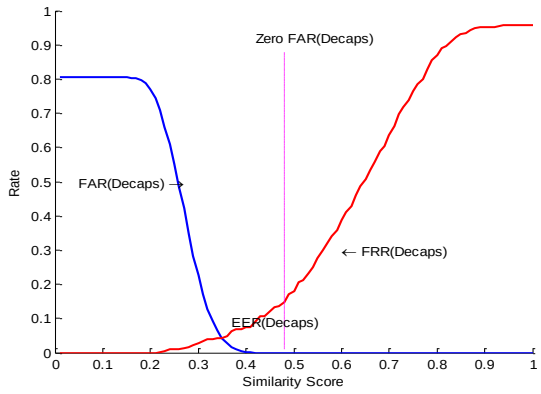


Figure 10: FAR/FRR of the dual fingerprint matcher that employs cancelable template.

7 Conclusion

An approach for generating chaff points for fingerprint template features encapsulation as fingerprint cancelability infrastructure has been presented. The approach takes advantage of fingerprint extracted information (minutiae points) to provide a novel way of generating chaffs from original ones. In addition this approach provides encouraging prospects to be used as platform of cancelable fingerprint feature extraction. From all the results, it could be able to prove that this approach with the usage of general extracted minutiae based new chaff points gave a better performance results and it is experienced as an efficient method for irrevocability and cancelablity of fingerprint template encapsulation.

8 References

[1] Punithavathi, P. and G. Subbiah, *Can cancellable biometrics preserve privacy*. Biometric Technology Today, 2007. 7: p. 8-11. https://doi.org/10.1016/S0969-4765(17)30138-8

[2] Jain, A., K. Nandakumar, and A. Nagar, *Biometric Template Security.* EURASIP Journal on Advances in Signal Processing, 2008: p. 1-17. https://doi.org/10.1155/2008/579416

[3] Ang, R., R. Safavi–Naini, and L. McAven, *Cancelable key-based fingerprint templates*, in Proc of the Australasian Conf. on Information Security and Privacy ACISP'05,242-252 2005. https://doi.org/10.1007/11506157_21

[4] Moujahdi, C., et al., *Spiral Cube for Biometric Template Protection*, in Image and Signal Processing. 2012. p. 235-244. https://doi.org/10.1007/978-3-642-31254-0_27

[5] Hirata, S. and K. Takahashi, *Cancelable Biometrics with Perfect Secrecy for Correlation-Based Matching,* ICB 2009,LNCS, Tistarelli, M and Nixon, M.S. (Eds), Springer, 2009. **5558**: p. 868-878. https://doi.org/10.1007/978-3-642-01793-3_88

[6] Patel, V.M., N.K. Ratha, and R. Chellappa, *Cancelable Biometrics: A Review.* IEEE Signal Processing Magazine, 2015. **32**(5): p. 54-65. https://doi.org/10.1109/MSP.2015.2434151

[7] Reiter, M., et al. *Cryptographic key-generation from voice*. in IEEE Computer Society Symposium on Research in Security and Privacy. 2001. USA. https://doi.org/10.1109/SECPRI.2001.924299

[8] Uludag, U., et al., *Biometric cryptosystems: issues and challenges.* Proceedings of the IEEE, 2004. **92**(6): p. 948-960. https://doi.org/10.1109/JPROC.2004.827372

[9] F.Hao, R. Anderson, and J. Daugman. Combining crypto with biometrics effectively. in IEEE Transactions on Computers 2006. https://doi.org/10.1109/TC.2006.138

[10] Ratha, N.K., et al., *Generating Cancelable Fingerprint Templates.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. **29**(4): p. 561-572 https://doi.org/10.1109/TPAMI.2007.1004

[11] Viellhauer, C., R. Steinmetz, and A. Mayyerhofer. *Biometric hash based on statistical features of online signatures*. in the International conference on Pattern Recognition. 2002. https://doi.org/10.1109/ICPR.2002.1044628

[12] Goh, A. and D.C.L. Ngo, *Computation of Cryptographic Keys from Face Biometrics,* in Communications and Multimedia Security. Advanced Techniques for Network and Data Protection: 7th IFIP-TC6 TC11 International Conference, CMS 2003, , A. Lioy and D. Mazzocchi, Editors. 2003, Springer Berlin Heidelberg: Berlin, Heidelberg. p. 1-13. https://doi.org/10.1007/978-3-540-45184-6_1

[13] R.Ang, R.Safav-Naini, and L.McAven. Cancelable *Key-based Fingerprint Templates*. in 10th Australian Conf, Information Security and Privacy. 2005. https://doi.org/10.1007/11506157_21

[14] Ratha, N.K., et al., *Generating Cancelable Fingerprint Templates.* IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2007. **29**: p. 561-572. https://doi.org/10.1109/TPAMI.2007.1004

[15] Nagar, A. and A.K. Jain. *On the security of non-invertible fingerprint template transforms*. in 2009 First IEEE International Workshop on Information Forensics and Security (WIFS). 2009. https://doi.org/10.1109/WIFS.2009.5386477

[16] Dabbah, M.A., W.L. Woo, and S.S. Dlay. *Secure Authentication for Face Recognition*. in IEEE Symposium on Computational Intelligence in Image and Signal Processing. 2007. USA. https://doi.org/10.1109/CIISP.2007.369304

[17] Palmer, L.R., et al. *Efficient fingerprint feature extraction: Algorithm and performance evaluation*. in 2008 6th International Symposium on Communication Systems, Networks and Digital Signal Processing. 2008. https://doi.org/10.1109/CSNDSP.2008.4610735

[18] FVC2002, F.w.s. [cited; Available from: http://bias.csr.unibo.it/fvc2002].

[19] Zhao, F. and X. Tang. *Preprocessing for skeleton-based fingerprint minutiae extraction*. in Proc. Int'l Conf Imaging Science, Systems, and Technology (CISST). 2002. https://doi.org/10.1016/j.patcog.2006.09.008

[20] Sudiro, S.A., M. Paindavoine, and M. Kusuma. *Simple Fingerprint Minutiae Extraction Algorithm Using Crossing Number On Valley Structure*. in IEEE Workshop on Automatic Identification Advanced Technologies, 2007. 2007. https://doi.org/10.1109/autoid.2007.380590

https://doi.org/10.31449/inf.v42i3.1559 Informatica **42** (2018) 375–399 **375**

Using Semantic Perimeters with Ontologies to Evaluate the Semantic Similarity of Scientific Papers

Samia Iltache
Department of Computer Science, Mouloud Mammeri University, Tizi ouzou, Algeria
E-mail: siltache@gmail.com

Catherine Comparot
IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, France
E-mail: Catherine.Comparot@irit.fr

Malik Si Mohammed
Department of Computer Science, Mouloud Mammeri University, Tizi ouzou, Algeria
E-mail: m_si_mohammed@esi.dz

Pierre-Jean Charrel
IRIT, Université de Toulouse, CNRS, INPT, UPS, UT1, UT2J, France
E-mail: Charrel@univ-tlse2.fr

Keywords: domain ontologies, semantic annotation, classification, conceptual graph, semantic perimeter, text similarity

Received: March 17, 2017

The work presented in this paper deals with the use of ontologies to compare scientific texts. It particularly deals with scientific papers, specifically their abstracts, short texts that are relatively well structured and normally provide enough knowledge to allow a community of readers to assess the content of the associated scientific papers. The problem is, therefore, to determine how to assess the semantic proximity/similarity of two papers by examining their respective abstracts. Given that a domain ontology provides a useful way to represent knowledge relative to a given domain, this work considers ontologies relative to scientific domains. Our process begins by defining the relevant domain for an abstract through an automatic classification that makes it possible to associate this abstract to its relevant scientific domain, chosen from several candidate domains. The content of an abstract is represented in the form of a conceptual graph which is enriched to construct its semantic perimeter. As presented below, this notion of semantic perimeter usefully allows us to assess the similarity between the texts by matching their graphs. Detecting plagiarism is the main application field addressed in this paper, among the many possible application fields of our approach.

Povzetek: Prispevek obravnava uporabo ontologij za primerjavo znanstvenih besedil. Poglavitna uporaba je odkrivanje plagiacije.

1 Introduction

Assessing query-text or text-text similarity is the concern of several research domains such as information retrieval and automatic classification of documents. For many works, a document is represented by a vector of words. The very large size of the vectors reduces the effectiveness of these approaches and often requires reducing the number of dimensions to represent the document vectors. Some approaches are based on a learning corpus to compute the similarity between texts, as is done in the field of document classification. However, a large text corpus may not always be available and the result of the document classification depends and varies according to the chosen learning corpus. The similarity is based on the morphological comparison of the terms composing the query and the documents. The polysemy and synonymy inherent in the presence of certain terms of the language as well as the links between the terms are ignored, which generates erroneous matching.

In this paper, an approach to assess the similarity between texts is presented, focusing on the similarity of scientific abstracts. This approach is based on a semantic classification of documents using domain ontologies which provides a more stable base than a learning corpus. A document is no longer represented by a set of characteristics independent of each other, but by a conceptual graph extracted from the ontology to which the document is attached. The similarity between two documents is evaluated by comparing their respective graphs.

One of our propositions is to refine this process of semantic comparison through a generic structuring of an abstract of a scientific paper into distinct parts whose descriptive roles are different. The global similarity of

two abstracts will indeed be different according to whether one compares, for example, the contribution or the context of the paper, both evoked in the abstract. The proposed process constitutes a solution that can answer many problems requiring semantic comparison, as is the case, for example, in Semantic Information Retrieval. Finally, the relevance of our approach is examined by using it to highlight risks of plagiarism (expressing identical ideas using different terms), or even self-plagiarism (identical results published more than once by their authors, voluntarily using different terms).

In addition to an original process to compare the abstracts of scientific papers based on domain ontologies, and combine a classification process with a semantic comparison of conceptual graphs, one of our main contributions is the introduction of the concept of semantic perimeter which is obtained by an ontology enrichment process. The semantic perimeter plays an important role in semantic comparison as shown by our results. Our approach also introduces the possibility of structuring scientific abstracts in three distinctive parts, generally respected by authors, namely *Context*, *Contribution* and *Application domain*. Finally, this constitutes a complete process for semantic text comparison, starting by using domain ontologies, and reaching text similarity.

Section 2 of this paper covers some work related to our problematic. Section 3 describes the different steps of our text classification and comparison process and explains how to perform this process using scientific abstracts. Finally, Section 4 presents the experimentation results of our process, followed by a conclusion on the interest of such an approach and its applicability on several domains, such as giving a useful approach to constituting a documentary fund on a given knowledge domain by collecting relevant papers, which is more powerful than a mere keyword-based approach, or detecting plagiarism, which is our main purpose here.

2 Related work

2.1 Word similarity

Similarity measures are necessary for various applications in natural language processing such as word sense disambiguation [1] and automatic thesauri extraction [2]. They are also used in Web related tasks such as automatic annotation of Web pages [3]. Two classes of approaches dealing with word similarity measure can be distinguished.

Distributional approaches [4] consider a word based on its context of appearance. Words are represented by a vector of words that co-occur with them. Latent Semantic Indexation [5] is a vectorial approach that exploits co-occurrences between words. It reduces the space of words by grouping co-occurring words in the same dimensions using Singular Value Decomposition. The textual content of Wikipedia [6][7] and the Neural networks [8][9] are used for distributional word similarity to define the context of a word. In the second category, the similarity of two words is based on the similarity of their closest senses. For this purpose, a lexical resource is used, such as WordNet and MeSH. The nodes at these resources represent the meaning of the words. Measures that make it possible to calculate the degree of proximity (distance) between two nodes have been defined. Several approaches can be identified for calculating of such distances: Approaches based only on the hierarchical structure of the resource [10][11][12][13]. The measure proposed in [11] is based on edge counting and the measure proposed in [12] is based on the notion of least common super-concept; that is, the common parent of two nodes, the furthest from the root. In [13], the proposed measure takes into account the minimum distance between two nodes to their most specific common parent (*cp*) and the distance between *cp* and the root. Some approaches include information other than the hierarchical structure information, such as statistics on nodes or the informative content of nodes. To represent information content value, probabilities based on word occurrences in a given corpus are associated with each concept in the taxonomy [14][15]. Resources, such as Wikipedia [16][17] and Wiktionary [18], are also used in measuring word similarity.

2.2 Text similarity

The purpose of calculating text similarity is to identify documents with similar or different content. The different approaches dealing with textual similarity can be classified into three categories: approaches based on vector representation of document content, approaches applying text alignment, and approaches based on a graphical representation of documents and queries. Some approaches relating to each category are cited below.

2.2.1 Vector similarity

A text (document or query) is projected into a vector space where each dimension is represented by an indexing term. Each element of a vector consists of a weight associated with an indexing term. This weight represents the importance of a term and is calculated on the basis of TF-IDF [19] or its variants. The vector similarity is computed using several metrics such as the cosine measurement which measures the cosine of the angle formed by the vectors corresponding to the texts. Two texts are similar if their vectors are close in the vector space in which they are represented.

- Document retrieval

The vector model is proposed by Salton in the SMART system [20]. To retrieve the documents that best meet a user need, a document and a query are represented by a vector. The relevance of a document to a query is measured by a similarity based on the distance between their respective vector. Adaptations of the basic model have been proposed for processing structured documents [21][22]. The Extended Vector Space Model is one of the first adaptations of the vector model proposed by Fox [22]. A document is represented by an extended vector containing different information classes referred to as objective identifiers (denoted by c-type) such as author,

title and bibliographic references. The similarity between a document *d* and a query *q* is computed by a measure of similarity which is a linear combination of the different sub vector similarities.

Conventional Information Retrieval considers documents only based on their textual content. The evolution of the document content towards a structured representation and more precisely towards the XML format raises new issues. In [23], the author presents a Searching XML documents through xml fragments. A fragment is a text delimited by a structure. The queries are transformed into XML fragments and, for each document, a profile is created. This profile is represented by a vector composed of the pairs (t, c), where *c* is the context of appearance of the term *t*. The context is assimilated to the element with its path. An entry in the index is no longer a term but a pair (t, c). Another adaptation of the vector model described in [24] based on the computation of the cosine makes it possible to compute the similarity between a node *n*, belonging to a tree representing a document, and a query *q*. In [25], the corpus is represented by a labeled tree where each sub-tree is considered as a logical document. The authors introduce the notion of structural term (*s-term*) which is a labeled tree. An *s-term* may be an element, an attribute, or a term. The similarity between a query and a document is computed by the scalar product of the vectors. The weight of the terms is computed during the retrieval phase since the notion of logical tree is defined according to the structure of the query.

- Document classification.

Automatic texts classification makes it possible to group documents dealing with similar themes around the same class. Supervised classification approaches assign documents to predefined classes [26][27][28] while unsupervised classification approaches automatically define classes, referred to as clusters, [29].

In the supervised classification, classifiers use two document collections: A collection containing training documents to determine the characteristics of each category and a collection containing new documents to be automatically classified. The classification of a new document depends on the characteristics selected for each category. There are various supervised machine learning classification techniques. In [30], the author provides a comparison of their features.

The method based on the K Nearest Neighbors (KNN) [28][31] assumes that if the vectorial representations of two documents are close in vector space, they have a strong probability of belonging to the same category. A new document *d* is compared with documents belonging to the training set. The category assigned to document *d* depends on the category of its K nearest neighboring documents. To determine the category to be assigned to the document *d*, the most assigned class to the K neighbors closest to *d* is chosen or a weight is assigned to the different classes of k nearest neighbors according to the classification of these neighbors. Thus the class with the highest weight will be retained.

With Support Vector Machines (SVM), documents are represented in a vector space by the indexing terms that compose them. Using a training phase, this method defines a separating surface, called hyperplan, between the documents belonging to two classes which maximize the distance between this hyperplan and the nearest documents and minimizes categorization errors [32]. A category *c* is assigned to a new document *d* as a function of the position of *d* relative to the separating surface.

Some classifiers create a "prototype" class from the training collection [26]. This class is represented by the mean vector of all the document vectors in the collection. Only some features are retained which constitutes a loss of information. Some approaches replace the training collection with data extracted from "world knowledge" such as Open Directory Project (ODP) [33]. Other approaches exploit thesauri or domain ontologies with conventional classifiers (SVM, Naive Bayes, K-means, etc.) and represent a document by a vector whose features are concepts or a set of terms and concepts [29][34][35].

As reported in [36], approaches using the vector representation of documents have several limitations: Their performances decrease as soon as they apply to relatively long texts. With the weighting formulas used, words appearing only once in the document or, on the contrary, words that are often repeated are ignored although they have a meaning with respect to the content of the document. The vector representation as defined does not highlight the relationships between words in a document, thus generating erroneous matching.

A document is represented by a vector whose size is equal to the number of features retained to represent the various categories, in the case of classification, and the number of terms used to represent the corpus, in the case of information retrieval. In [37], the authors studied the impact of the number of dimensions on the "nearest neighbor" problem. Their analysis revealed that when this number increases, the distance to the nearest data point approaches the distance to the farthest data point.

2.2.2 Sentence alignment

Approaches dealing with sentence alignment are divided into three categories. Syntactic approaches based on morphological word comparison, semantic approaches using sentence structure and approaches that combine syntax and semantics. Gunasinghe [38] proposes a hybrid algorithm that combines syntactic and semantic similarity and uses a vectorial representation of sentences by using WordNet. This algorithm takes into account two types of relationship in the sentence pairs: relationships between verbs and relationships between nouns. Liu [39] proposes an approach to evaluate the semantic similarity between two sentences. They use a regression model, Support Vector Regression, combined with features defined using WordNet, corpus, alignment and other features to cover various aspects of sentences. Other approaches perform the text alignment by comparing all the words preserving their order in sentences. However, these algorithms are rather slow and they do not

dissociate terms describing the theme of the document from those used to build sentences. In [40], authors use a text alignment algorithm [41] to align a text with the set of documents in a corpus. This algorithm uses a matrix in which the deletion or insertion of a word is represented by -1, a mismatch by a 0 while a match is represented by its IDF weight. The authors use a full-text alignment where the highest score from any cell in the alignment matrix represents the similarity score of two texts. In [42], authors introduce a new type of sentence similarity called Structural Similarity for informal, social network styled sentences. Their approach eliminates syntactic and grammatical features and performs a disambiguation process without syntactic parsing or POS Tagging. They focus on sentence structures to discover purpose- or emotion-level similarities between sentences.

2.2.3 Graph similarity

Assessing of the graph similarity is used, in particular, in the field of Information Retrieval. The document and query are both represented by a conceptual graph constructed from a domain ontology or a thesaurus.

In the domain of Semantic Information Retrieval, Dudognon [43] represents the documents by a set of "annotations". Each annotation consists of several conceptual graphs. The similarity between two graphs is defined as the weighted average of the similarities between the concepts that compose this graphs and the similarity between two "annotations" is computed by the mean of similarities of their conceptual graphs. Baziz [44] suggests constructing a graph for each document and for each query using concepts extracted from WordNet. A mapping of the graph of a document to that of the query leads the author to represent the two graphs with respect to the same reference graph made up of nodes belonging to the document and to the query. Each graph is then expanded by adding nodes of the reference graph. The weights of the nodes added to the query are zero whereas in the sub-tree of the document where a node is added, the weight of a level s node is updated recursively by multiplying the weight of the level $s + 1$ node (the level s node subsumes the level $s + 1$ node) by a factor which depends on the hierarchy level. The two representations are then compared using fuzzy operators and a relevance value is computed. This value expresses the extent to which the document covers the subject expressed in the query. Shenoy [45] represents a document by a "sub-ontology" constructed using the demo version of ONTO GEN Ontology Learner which is part of the TAO Project. Two documents are compared by applying the alignment of their "sub-ontology" based on the number of concepts, properties and relationships contained in each document. In [46], the authors propose a unified framework of graph-based text similarity measurement by using Wikipedia as background knowledge. They call each article in Wikipedia a Wikipedia concept. For each document, the authors extract representative keywords or phrases and then map them into Wikipedia concepts. These concepts constitute the nodes at the bottom of the bipartite graph. There is an edge between a document node and a concept node if the concept appears in the specific document. The weight of the edge is determined by the frequency of the concept's occurrence in that document. The similarity of two documents is determined by the similarity of the concepts they contain. The authors in [18] present a unified graph-based approach for measuring semantic similarity between linguistic items at multiple levels: senses, words, and sentences. The authors construct different semantic networks. One of them is based on WordNet. The nodes in the WordNet semantic network represent individual concepts, while edges denote manually-crafted concept-to-concept relations. This graph is enriched by connecting a sense with all the other senses that appear in its disambiguated gloss. Measuring the semantic similarity of a pair of linguistic items consists of an Alignment-based Disambiguation and a random Walk on a semantic network. In [47], authors propose a graph-based text representation, which is capable of capturing term order, term frequency, term co-occurrence, and term context in documents. A document is represented by a graph. A node represents a concept: a set of single word or phrase and an edge is constructed based on proximity and co-occurrence relationship between concepts. In addition; the associations among concepts are represented through their contexts. The nodes within the window (e.g. paragraph, sentence) are linked by weighted bidirectional edges. The approach described in [48] presents a graph-based method to select the related keywords for short text enrichment. This method exploits topics as background knowledge. The authors extract topics and re-rank the keywords distribution under each topic according to an improved TF-IDF-like score. Then, a topic-keyword graph is constructed to prepare for link analysis. In [49], the authors create a semantic representation of a collection of text documents and propose an algorithm to connect them into a graph. Each node in a graph corresponds to a document and contains a subset of document words. The authors define a feature and document similarity measures based on the distance between the features in the graph.

2.3 Detecting plagiarism

Plagiarism consists in copying a work of an author and presenting it as one's own original work. Plagiarism detection systems usually have the original document and the suspicious document as inputs. They focus on the following points: an exact copy of the text (copy/paste), inserting or deleting words, substituting words (use of synonyms), reformulation and modification of sentences structure. In n-gram approach, a text is characterized by sequences of n consecutive characters [50][51][52]. Based on statistical measures, each document can be described with so called fingerprints, where n-grams are hashed and then selected to be fingerprints [53]. An overlap of two fingerprints extracted from the suspicious and source documents indicates a possibly plagiarized text passage. Statistical methods [54] do not require an understanding of the meaning of the documents. The

common approach is to construct the document vector from values describing the document such as the frequency of terms. Comparing the source document with the suspicious document, amounts to calculating their degree of similarity on the basis of different measures (BM25, language model, etc.). Vani [55] segments the source document and the suspicious document into sentences. Each sentence is then represented by a vector of weighted terms that compose it. Each sentence of the source document is compared to all the sentences of the suspicious document and similarity between two vectors is computed using, individually, several metrics (cosine, dice, etc.). Vani studies the importance of the combination of these various metrics on detecting plagiarism. He also explores the impact of the use of POS Tagging on calculating of sentence similarity. The sentences labeled by a syntactic parser are thus compared by matching the terms belonging to the same class (nouns with nouns, verbs with verbs, adjectives with adjectives and adverbs with adverbs). Other approaches based on sentences alignment compute the overlapping percentage of words or sentences between the source document and the suspicious document. These methods do not permit the detection of cases of plagiarism where synonymy is used to replace words in the reformulation of sentences. The representation of a document by a graph is also used in detecting plagiarism. In [45], the alignment of "sub-ontologies" is based on the number of concepts, properties and relations corresponding to the original document and the suspicious document. Alignment is expressed as a fraction of the whole. If this fraction is above a given threshold, the system concludes that the two documents are similar in meaning. Osman [56] describes an approach of detecting plagiarism by representing documents (original and suspicious) with a graph deduced from WordNet. This approach is useful in detecting forms of plagiarism where synonymy is used to reformulate sentences. The document is divided into sentences. Each node of the graph constructed for the document represents the terms of a sentence. The terms of sentences are projected on WordNet to extract the concepts corresponding to them. Each relationship between two nodes is represented by the overlap between the concepts of the two nodes. These concepts help in detecting suspicious parts of a document.

An important characteristic of our approach lies in the fact that it is not necessary to have a reference document a priori, since any document can be compared with a corpus dealing with the same knowledge domain as identified in the first step of our process that is proposed here.

3 Proposed approach

The representation of a document by a semantic graph is used in different domains such as information retrieval [43][44], plagiarism detection [45][56] and document summarization [57]. However, these graphs differ in the way they are constructed. The purpose of our approach is to assess the semantic similarity between textual documents. Unlike conventional approaches, a document is not represented by a vector. Our approach is to build a conceptual representation of a text in the form of a semantic graph in which the nodes and arcs correspond respectively to concepts and relationships between concepts extracted from the domain ontology chosen.

The similarity between two texts is evaluated in two steps. The first step is to perform a semantic classification of documents based on domain ontologies. The classification makes it possible to deduce an overall similarity defined by the context in which the content of the document is used. The second step compares and evaluates the similarity of two texts related to the same domain ontology by comparing their constructed and enriched graph as explained in the following sections.

3.1 Classification of documents

The process is based on a semantic classification of texts using domain ontologies [58]. Figure 1 summarizes the classification process.

The classification groups documents according to the knowledge domain covered by their content. This grouping identifies an overall similarity and involves several steps.

- *Projection, extraction of terms and candidate concepts*. The "projection" of a document on different ontologies helps to associate meaning to the terms of the document with respect to concepts belonging to these ontologies and to select the candidate concepts. The notion of concept gives a meaning to a term relative to the domain in which this concept is defined. The whole document is divided into sentences. Each sentence is browsed from left to right from the first word. The words of each sentence are projected, before pruning stop words, on different domain ontologies to extract longer phrases (groups of adjacent words in a sentence called "terms") that denote concepts. This choice is determined by: 1) the concepts are often represented by labels consisting of several words. An example of mono- and multi-word concepts is given in table 1. 2) long terms are less ambiguous and better determine the meaning conveyed by the sentence. Several concepts belonging to the same domain ontology may be candidates for a given term. The following example shows to what extent it is important to bring out the longest terms and the longest concept.

For the sentence: "The **Secretary of State for the Home Department** had clearly indicated that evidence obtained by torture was inadmissible in any legal proceedings," the synsets in Table 1 are extracted from WordNet.

As shown in Table 1, there are several synsets in WordNet that correspond to the words "secretary of state for the home department" in the sentence. These synsets have one or more words.

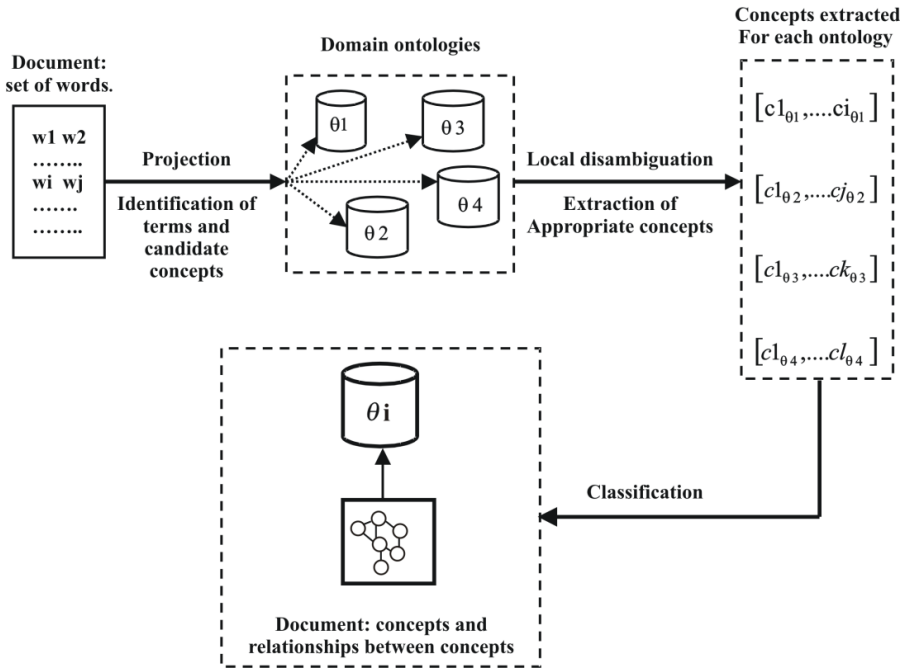


Figure 1: Classification of a document.

Words in a sentence	Synset label in WordNet	N° synset in WordNet
Secretary of State for the home Department	secretary_of_state_for_the_home_department	**09526473**
	secretary_of_state	09883412 09455599 00569400
	secretary	09880743 09880504 09836400 04007053
	state	07682724 08125703 07673557 00024568 07646257 08023668 13192180 13656873
	home	08037383 03141215 07973910 13687178 03398332 07974113 07587703 03399133 08060597
	department	07623945 08027411 05514261

Table 1: Extraction of terms and synsets.

The longest term "secretary of state for the home department" is extracted from the sentence. It corresponds to the synset secretary_of_state_for_the_home_department (09526473), which represents the correct sense in the sentence.

- *Local disambiguation.* In the projection step, for each ontology, all the candidate concepts for a given term are extracted. The local disambiguation process is used to select for a term *t* the most appropriate concept among several candidates belonging to the same ontology. To do this, the context of occurrence of the term *t* in the document is taken into consideration.

The appropriate concept for the term *t* is chosen, taking into account both the semantic distance between the term *t* with neighboring terms, (i.e., which occur in its context), and the semantic distance between concepts associated with the term *t* and concepts corresponding to the neighboring terms in the ontology considered.

The meaning of a term *t* in a document is determined by its nearest unambiguous neighbors terms. *t* will then be disambiguated by its nearest neighbor on the left or by its nearest neighbor on the right. In case the left and right neighbors exist simultaneously, they will both be taken into consideration.

The disambiguation process is then done at three levels, starting at the sentence level. For each sentence, the ambiguous terms are disambiguated considering their left and right neighbors in the sentence. Any disambiguated term helps to move forward in the process of disambiguation of next terms. This process is repeated in case ambiguous terms still remain, considering in a second step the paragraph level, and finally, if necessary, the document level. The local disambiguation process at the sentence level, summarized by the algorithm in Figure 2, considers neighboring terms, unambiguous, that have associated concepts in the ontology considered, surrounding *t*: it retrieves the concepts *Cnl* and *Cnr*, corresponding respectively to *nl*, the nearest neighbor on the left of *t* and *nr*, the nearest neighbor on the right of *t*. The appropriate concept for the term *t* among candidate concepts is the semantically nearest concept of *Cnl* or

Cnr. This amounts to browsing the ontology and calculating the minimum distance between each concept associated with t and candidate concepts Cnl, Cnr.

Several existing metrics in the literature are used to calculate this minimum distance. An example of local disambiguation in the domain anatomy of WordNet is given in the Figure3.

```
Input
    Ec = {extracted concepts for S}  {S, current sentence}
    Et = {terms belonging to S}
    E = {Unambiguous terms of S}
Output
   Ec = {retained concepts for S}

Procedure disambiguation (i:integer)
  var
    j:integer
Begin
  t ←S[i]
  nl (t) ← S[i-1]
  nr (t) ← S[i+1]
  if (nl (t) in E ) and (nr (t) in E)  then
    compute  Min-dist ((Ci,Cnl), (Ci,Cnr))  {Ci, The concepts associated with t}
    E ← E ∪ t
    Ec ← Ec ∪ C        {C, retained concept for t}
  else
    if (nl (t) in E)  then
      compute  Min-dist (Ci,Cnl)    {Cnl: The concepts associated with nl}
    E ← E ∪ t
    Ec ← Ec ∪ C
   else
      if (nr (t) in E) then
        compute  Min-dist (Ci,Cnr)    {Cnr: The concepts associated with nr}
        E ← E ∪ t
        Ec ← Ec ∪ C
      else
        j ← i + 1
        disambiguation(j)
        pos ← pos + 1
        t ← S[j-1]
        compute  Min-dist (Cj-1,Cnr)
        E ← E ∪ t
        Ec ← Ec ∪ C
      End if
    End if
  End if
End
Begin
  Ec←Ø
  pos ← 1
  k ← 1
  t ← S[k]
  while ( not end (S) ) do
    if ( t not in E)  then
      disambiguation (k)
       k ← pos + 1
       pos ← pos + 1
    else
      Ec ← Ec ∪ C
      pos← k+1
      k ←k + 1
    end if
    t ← S[k]
  end while
 end.
```

Figure 2: Local disambiguation at the sentence level.

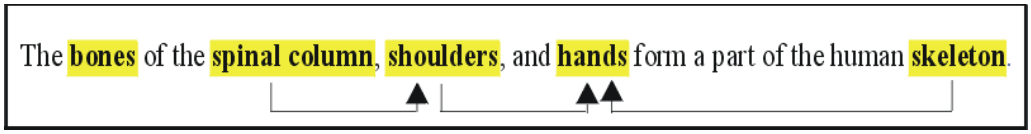


Figure 3: Disambiguation of shoulder and hand.

Table 2 shows the terms and their senses (synsets) in the domain *anatomy* of WordNet. The different calculated distances help in choosing the most appropriate synset for each ambiguous term.

The term *shoulder* in the sentence is ambiguous. To disambiguate it, *spinal column*, its nearest unambiguous neighbor term on the left, is considered. The synset retained is 05231159.

The term *hand* in the sentence is ambiguous. Its disambiguation is done using *shoulder* and *skeleton*, its two nearest unambiguous neighboring terms on the left and right. The synset retained is 05246212.

Words in a sentence	Synset label (Anatomy)	N° synset	Distance between synsets	Terms extracted
Bones	bone	04966339		bone
	Spinal_column	05268544		Spinal column
Spinal	shoulder	**05231159** 05231380	Dist(05268544,05231159)= 0.42857143 Dist(05268544, 05231380)= 0.5	shoulder
Column				hand
Shoulders (ambiguous)	hand	**05246212** 02352577	Dist(05246212,05231159)= 0.42857143 Dist(02352577,05231159)= 0.6363636	skeleton
Hands (ambiguous)			Dist(05246212,05265883)= 0.42857143 Dist(02352577,05265883)= 0.6363636	
skeleton				
	skeleton	05265883		

Table 2: Disambiguation of ambiguous terms.

At the end of the preceding steps, a document d is represented by several sets of concepts extracted from the domain ontologies θ_i on which it was projected. These sets are represented by (1).

$$d = \begin{cases} \theta_1^d = \{c_{11}, c_{21} \ldots\ldots, c_{n1}\} \\ \theta_i^d = \{c_{1i}, c_{2i} \ldots\ldots, c_{ni}\} \\ \ldots\ldots\ldots \\ \ldots\ldots\ldots \end{cases} \quad (1)$$

- *Global disambiguation.* The classifier must be able to conclude about the relevance of a document relative to a given context and to choose from the different ontological representations the one that best corresponds to its context. A score is calculated for each document. The highest score determines the candidate ontology to be selected to represent document d.

The different terms in a document, taken together considering the contextual relations linking them, make it possible to conduct a semantic evaluation of the textual content. A matrix, defined by (2), is associated for each ontology and for each document.

$$M_{\theta_i}^d = \begin{pmatrix} lc_1c_1 & lc_1c_2 \ldots\ldots & lc_1c_n \\ & & \\ lc_nc_1 & lc_nc_2 \ldots\ldots & lc_nc_n \end{pmatrix} \quad (2)$$

The rows and columns of this matrix represent all the concepts extracted from the ontology θ_i for the document d.

C_i is the selected concept for the term t_i after projection of the document d on θ_i and lc_ic_j represents the weight of the link between the concept C_i and the concept C_j (i≠j).

The matrix is initialized to zero.

If a term t_i and a term t_j appear together within the same paragraph of the document d and the concepts Ci and Cj respectively correspond to the terms t_i and t_j, then the weight $lcicj = 1$.

The weight $lcicj$ is updated whenever the terms t_i and t_j appear together in the same paragraph.

The weight $lcici$ corresponds to the appearance of the term t_i in the document. It is equal to 1.

The weight $lcicj$ is updated for all paragraphs of the document d.

The importance of the concept C_i in document d is determined by its total weight in d relatively to the ontology θ_i. This weight is given by the row associated with it in the matrix.

The score for each ontology obtained from the sum of the weights of all concepts extracted from this ontology for the document d measures the extent to which each ontology represents this document. The ontology that gets the highest score will be selected to represent the document d.

For documents belonging to the same knowledge domain, their "local" semantic similarity is computed.

The process compares their content using their semantic perimeter – a notion that is introduced and defined later in the paper – constructed on the basis of their conceptual graph extracted from the ontology to which they are attached.

3.2 Text similarity and semantic perimeter

An author describes the subject of his document by evoking one or more different notions. He can describe them by addressing several sub-notions. These notions and/or sub-notions can be described in a general or precise way according to the level of detail to be highlighted.

In an ontology, there exists a certain structure defining the meaning of information representing a given knowledge domain and the way in which this information is related to each other. This structure is defined by several branches representing different hierarchies. Each hierarchy has branches to separate data with common characteristics but also different characteristics. The tree of Figure 4, inspired by the *geometric figures* ontology proposed in [59], shows two branches *Br1* (*figure)* and *Br2* (*angle*) representing two different data. Branch *Br2* has two sub-branches 2.1 and 2.2 corresponding respectively to a *right angle* and an *acute angle*. *Right angle* and *acute angle* are two concepts with different characteristics but common characteristics defined by their common parent a*ngle*.

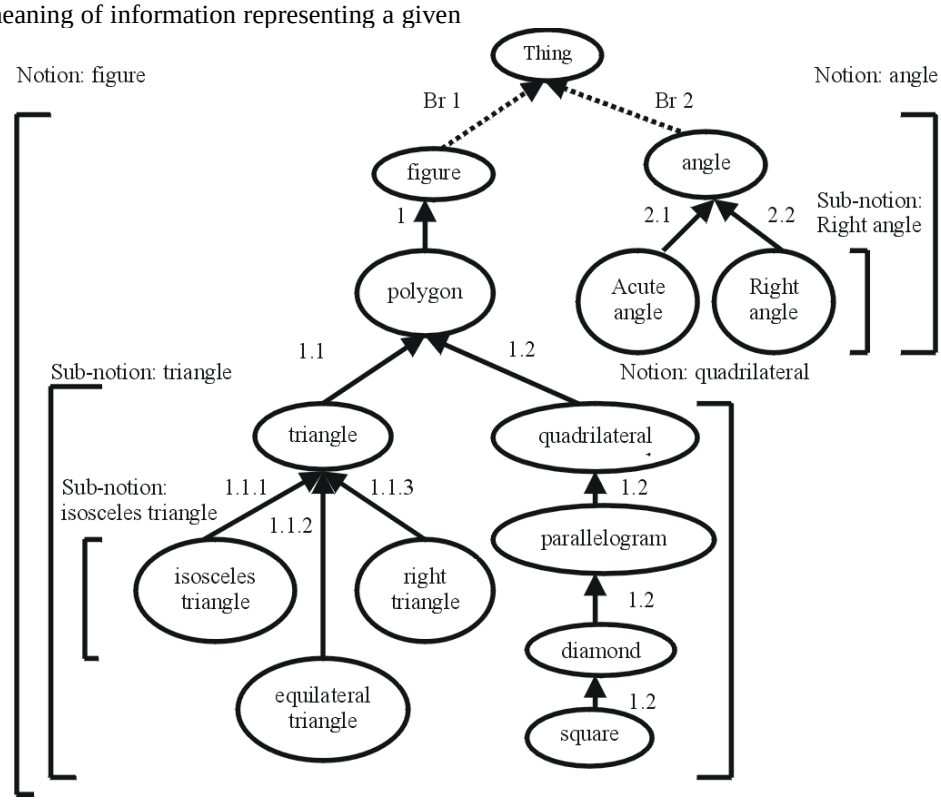


Figure 4: Extract from the *geometric figures* ontology.

3.2.1 Objective of the approach

Consider two texts *Txt1* and *Txt2*, previously classified in the same knowledge domain represented by a domain ontology, whose similarity needs to be assessed: *Sim (Txt1, Txt2*). Our semantic similarity process is based on the following assumptions:

1 Each branch/sub-branch of the ontology is associated with a notion/sub-notion described in a document.

2 Concepts linked by "*is-a*" relations form a branch.

3 A branch can have several sub-branches.

4 Two branches with the root of the ontology as the only common parent represent two different notions.

5 Two sub-branches having a common parent represent two different sub-notions sharing common characteristics defined by their common parent.

6 The weight of an *initial concept* is equal to 1.

7 The weight of an added concept representing implicit information is less than 1.

8 The similarity between two texts varies between 0 and 1.

Our approach is based on the identification of the branches to which the concepts of the documents belong and on the enrichment of the conceptual graphs of these documents. Associating a notion with a branch helps in identifying different and identical notions. It can be said for example that the notion "*angle*" is different from the

notion "*figure*" and that the notion "*triangle*" is different from the notion "*quadrilateral*" because they belong to different branches or sub-branches. The concepts *quadrilateral*, *parallelogram*, *diamond*, and *square* belong to the same sub-branch describing the same notion. Each of them brings a degree of precision knowing that this precision is increasingly higher the further one goes down the hierarchy.

Graph enrichment highlights common notions to two documents without these being explicitly cited in their content and makes it possible to deduce similarities between notions by examining the branches to which their corresponding concepts belong.

3.2.2 Graph enrichment

To describe a given subject, the authors, can choose different words and different levels of description depending on the importance that each of them wishes to give to a notion addressed in the text. Thus, by adding concepts, graph enrichment makes it possible to deduce implicit information that can be shared by these two texts.

Like Baziz [44], our process enriches the text graphs by adding concepts. The applied enrichment differs from that achieved by Baziz in the choice of concepts to be added and the weight assigned to these concepts. For our case, the weight assigned to the concepts helps in defining the implicit or explicit presence of a concept.

A graph is enriched by constructing the semantic perimeter of its corresponding text and comparing it to another graph.

3.2.2.1 Constructing the semantic perimeter of a text

Definition 1: The semantic perimeter of a text is a graph whose nodes are the initial concepts and the link concepts. Initial concepts are extracted from the domain ontology to which the document is attached. These concepts represent the information explicitly described in its content. With these concepts, a conceptual graph is constructed and enriched by link concepts representing the implicit information in the text that is deduced from the initial concepts and through browsing the "is-a" relationships and the transversal relationships defined in the domain ontology. The semantic perimeter thus constructed for each document makes it possible to evaluate their semantic similarity even if these documents describe the same ideas with different terms.

- *Constructing the graph of initial concepts*

During the classification process, a text is projected onto a set of domain ontologies. At the end of this step, the text is represented by a conceptual graph, whose nodes constitute the *initial concepts*.

These concepts correspond to the terms explicitly cited in the document.

- *Constructing the semantic perimeter*

The *link concepts* extracted from the ontology, being on the shortest path linking the *initial concepts Ci* and *Cj* by *is-a* relations or transversal relations, are added to the graph of a document.

Link concepts are selected in order to retain only concepts that make sense in relation to the knowledge domain represented by the ontology. In fact, some concepts represented in an ontology are used to construct the structure of the ontology but have no meaning for the domain in question.

Example: *host* and *hard_disk*, are two synsets extracted for a document classified in the *computer_ science* domain. Figure 5 shows the synsets linking them in WordNet.

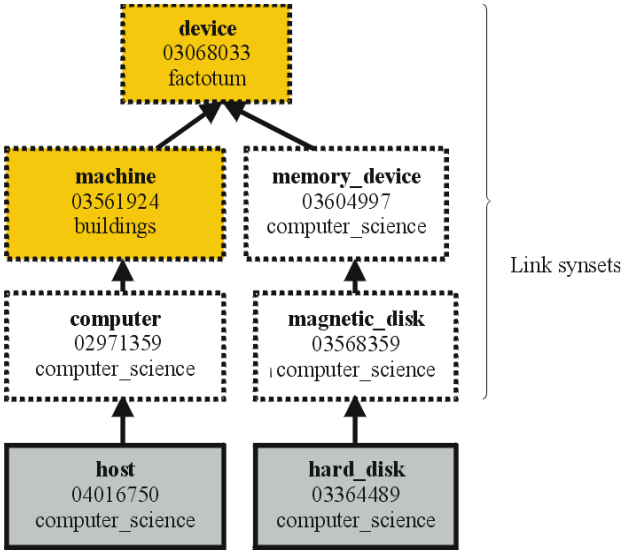


Figure 5: link synset linking host to hard_disk.

The *link synsets* are: {*computer* 02971359, *machine* 03561924, *device* 03068033, *memory_device* 03604997 and *magnetic_disk* 03568359}. The synsets *machine* 03561924 and *device* 03068033 are not retained, since they respectively belong to the *buildings* domain and *factotum* domain.

3.2.2.2 Comparing graphs

Comparing two texts *Txt1* and *Txt2* is carried out from their semantic perimeter *G1* and *G2*. A mutual enrichment of these two graphs is achieved by comparing the concepts belonging to *G1* with the concepts belonging to *G2*. Each graph enriched the other and concepts are added to *G1* and/or to *G2*. This is done by browsing the graphs from leaf nodes to the root as follow:

- If the graph *G1* (the graph *G2*) contains a concept *C1* and the graph *G2* (the graph *G1*) contains a concept *C2* such that *C2* is an ancestor of *C1*, then the concept *C2* is added to the graph *G1* (to the graph *G2*).
- The graphs are also enriched by adding the common parents to concepts belonging to graphs *G1* and *G2*. This enrichment is done in two steps:

- By considering concepts belonging only to the graph *G1* (to the graph *G2*).
- By considering the concepts belonging to graphs G1 and G2.

By adding *common parent* concepts, graph enrichment helps in determining the common branches and sub-branches to *G1* and *G2* and thus to deduce an implicit similarity between *Txt1* and *Txt2*.

As an illustration, in the *geometric figures* domain represented by figure 4, three texts (*T1*, *T2* and *T3*) are considered, and their content is as follows:

T1: *A square is a regular polygon with four sides. It has four right angles and its sides have the same measure.*

T2: *A diamond is a parallelogram. Some diamonds have four equal angles.*

T3: *A triangle has three sides. If it has a right angle, it is a right triangle.*

- Let us compare *T1* and *T2*.

The semantic perimeters of *T1* and *T2* and the comparison of their respective graphs *G1* and *G2* are given in Figure 6.

The projection of the texts *T1* and *T2* on the ontology represented by figure 4, allows us to find the initial concepts to construct graphs *G1* and *G2*.

G1 is represented by the concepts (*square*, *polygon*, *right angle*) and *G2* is represented by the concepts (*diamond*, *parallelogram* and *angle)*. At this stage, the graphs have no common concept.

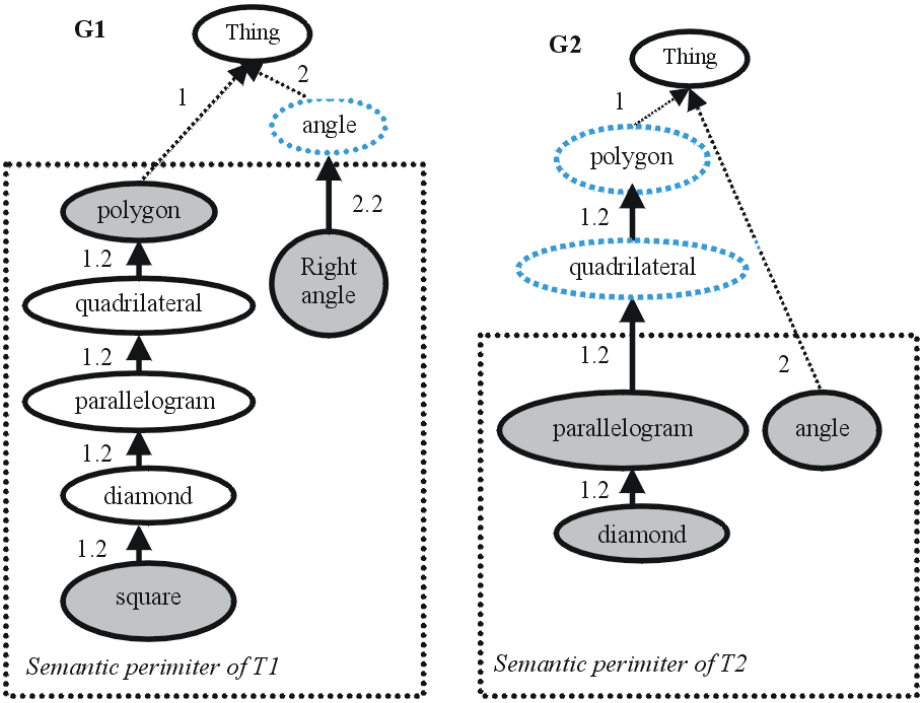


Figure 6: Comparison and enrichment of graphs corresponding to T1 and T2.

The enrichment of these two graphs made it possible to add concepts semantically linked to the initial concepts and to bring out common concepts to the two texts, not explicitly cited in their contents. The common concepts are *diamond*, *parallelogram*, *quadrilateral*, *polygon* and *angle*.

- Let us compare T2 and T3.

The semantic perimeters of *T2* and *T3* and the comparison of their respective graphs *G2* and *G3* are given in Figure 7.

The projection of the texts *T2* and *T3* on the ontology, represented by figure 4, allows us to find the initial concepts to construct graphs *G2* and *G3*.

G2 is represented by the concepts (*diamond*, *parallelogram* and *angle*) and *G3* is represented by the concepts (*triangle*, *right triangle* and *right angle*). The enrichment of the two graphs enabled us to find common concepts (*angle* and *polygon*).

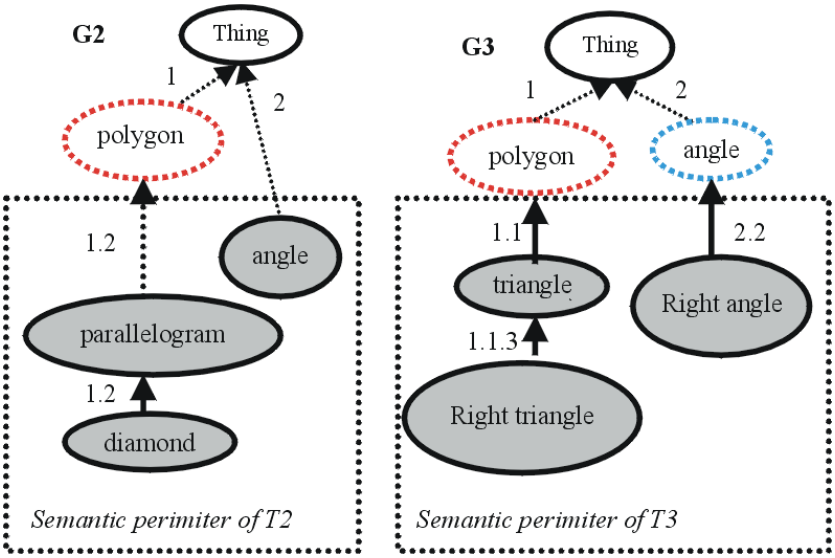


Figure 7: Comparison and enrichment of graphs corresponding to T2 and T3.

3.2.3 Calculating the similarity of two texts

Definition 2: Textual similarity is defined by the set of common notions and sub-notions addressed by two texts. It is a function of the concepts corresponding to these texts, their weight and the branches to which these concepts belong. The similarity of two texts is given by the similarity of their respective graphs according to equation (3).

$$Sim(Txt1, Txt2) = Sim(G_{Txt1}, G_{Txt2}) \quad (3)$$

3.2.3.1 Weight of the concepts

The weight attributed to an *initial concept* is equal to 1. This weight defines the explicit presence of the concept in the document. Concepts belonging to the same branch do not have the same semantic weight: concepts at the top of the hierarchy have a more general meaning than concepts at the bottom of the hierarchy that represent a more precise meaning. The more one descends towards the bottom of the hierarchy, the more precise the meaning of the concepts is. Thus, to a concept added to graph *G1* during the enrichment process, a weight whose value is less than 1 is assigned. This weight represents the value of an implicit information and is calculated based on parameter *g*. *g* expresses the degree of generalization of a *parent concept* vis-a-vis its *child concept*.

Like Fuhr [60] and Baziz [44], who reduce the weight of the nodes of a tree representing a document according to their position with respect to the most specific nodes by multiplying by a factor whose value is between 0 and 1, our process computes the weight of an added concept by using parameter g whose value is between 0 and 0.1 according to equation (4).

$$P(C_j) = 1 - (g \times (length(C_i, C_j)) \quad (4)$$

Cj is the added concept and *Ci* is the initial concept, belonging to *G1* and/or to *G2*, the lowest in the branch to which *Cj* is added and *Length (ci, cj)* indicates the number of arc linking *Cj* to *Ci* in the branch.

3.2.3.2 Semantic similarity of two graphs G1 and G2

A factor is introduced indicating the percentage of common notions described by two texts. Its value is calculated by the number of common branches relative to the total number of branches belonging to the two graphs. The similarity between two graphs *G1* and *G2* is computed using equation (5).

$$Sim(G1, G2) = \frac{nbBc_{(G1,G2)}}{nbB_{(G1,G2)}} \times \frac{\sum_{Bc} \sum_{Ccom \in Bc} P(C_{com})}{\sum_{B} \sum_{C \in B} P(C)} \quad (5)$$

B represents any branch belonging to the graphs while *Bc* represents a common branch to both graphs. *C* is a concept belonging to graphs *G1*, *G2* and *Ccom* is a common concept to both graphs. $nbBc_{(G1,G2)}$ and $nbB_{(G1,G2)}$ respectively represent the number of common branches and the total number of branches belonging to the two graphs.

3.2.3.3 Example

Let us again take the examples shown in Figures 6 and 7 and summarize the various results in Tables 3 and 4. For parameter *g*, the value 0.05 is used.

Initially, *G1* and *G2* showed no concept in common and, therefore, a priori no similarity. The same applies to graphs *G2* and *G3*. The enrichment of the graphs helped to bring out a similarity between *T1* and *T2*, as well as between *T2* and *T3* that is not explicitly described in their content. The results also show that text *T2* is semantically closer to *T1* than to *T3*.

Texts	Concepts	Type	Weight
T1	square	initial	1
	diamond	link	0,95
	parallelogram	link	0,90
	quadrilateral	link	0,85
	polygon	initial	1
	angle	Ancestor	0,95
	Right angle	initial	1
T2	diamond	initial	1
	parallelogram	initial	1
	quadrilateral	Ancestor	0,85
	polygon	Ancestor	0,80
	Angle	initial	1
Common branches		1 1.2 2	
All branches		1 1.2 2 2.2	

Table 3: Concepts of T1 and T2 after enriching their respective graphs.

Texts	Concepts	Type	Weight
T2	diamond	initial	1
	parallelogram	initial	1
	polygon	Common parent	0,85
	angle	initial	1
T3	right angle	initial	1
	right triangle	initial	1
	triangle	initial	1
	polygon	Common parent	0,85
	angle	ancestor	0,95
Common branches		1 2	
All branches		1 1.1 1.2 1.1.3 2 2.2	

Table 4: Concepts of T2 and T3 after enriching their respective graphs.

$$Sim(T1,T2) = \frac{3}{4} \times \frac{(0{,}80) + (0{,}85 + 0{,}90 + 0{,}95) + (0{,}95)}{(1) + (0{,}85 + 1 + 1 + 1) + (1) + (1)} = 0{,}49$$

$$Sim(T2,T3) = \frac{2}{6} \times \frac{(0{,}85) + (0{,}95)}{(0{,}85) + (1+1) + (1) + (1) + (1) + (1)} = 0{,}09$$

3.3 Similarity of scientific abstracts

Refining the process of semantic comparison of two texts (defined in section 3.2) is performed through a generic structuring of an abstract of a scientific paper into distinct parts whose descriptive roles are different.

Several works have taken interest in the annotation of the discursive structure of scientific papers: text zoning [61] [62]. Their objective is to better characterize the content of the papers by defining several classes (objective, method, results, conclusion, etc.), knowing that the existence of these classes depends on the corpus studied. Categorization is performed at the sentence level. For each sentence of an abstract, authors associate a class chosen from the defined classes.

This work deals with decomposing scientific abstracts into zones for the purpose of detecting plagiarism. From the structure generally reproduced by the authors of scientific papers, the content of a scientific abstract is divided into three distinct parts which are referred to as zones that define the *context*, the *contribution* and the *application domain*. This decomposition is generally reflected in most scientific papers that aim, in principle, at making a scientific contribution in a given domain. This decomposition aims to extract the notions relating to each zone and thus permits a comparison between zones of the same type. The process can then evaluate, in a progressive approach, whether two abstracts deal with the same context, whether their contributions are similar and whether they apply their approach to the same application domain, the risk of plagiarism evidently increasing with each conclusive comparison.

Categorization at the sentence level poses a problem when information from one class is cited in another class. In analyzing several abstracts, it was found that there is no strict uniformity in writing abstracts: all the sentences belonging to a given zone do not contain only the terms describing this zone but may contain terms representing another zone. For example, a sentence assigned to the *application domain* zone may contain terms defining an algorithm or a method (terms that instead define the *contribution* zone). This overlapping of several zones in the same sentence then generates labeling errors.

To illustrate the categorization at the sentence level, each sentence of abstract 2 provided in section (3.3.1), is associated with one of the three selected zones.

"*Recently, new approaches have integrated the use of data mining techniques in the ontology enrichment process.* <*context*>

Indeed, the two fields, data mining and ontological meta-data are extremely linked: on one hand data mining techniques help in the construction of the semantic Web, and on the other hand the semantic Web assists in the extraction of new knowledge. <*context*>

Thus, many works use ontologies as a guide for the extraction of rules or patterns, allow to discriminate the data by their semantic value and thus to extract more relevant knowledge. <*context*>

It turns out, however, that few works aimed at updating the ontology are concerned with data mining techniques. <*context*>

In this paper, we present an approach to support the onologies management of websites based on the use of Web Usage Mining techniques. <*contribution*>

The presented approach has been tested and evaluated on an website ontology , which we have constructed and then enriched based on the sequential patterns extracted on the log. <*Application domain*>"

The following inconsistencies are noted:

- The term *sequential pattern* is assigned to the *Application domain* zone while it represents the algorithm and method used by the author and, therefore, defines the *contribution*.

- The term *Data mining technique* is assigned to the *context* zone while it represents the *contribution*.

- The term *ontologies management* is assigned to the *contribution* zone while it defines the *context*.

To evaluate the semantic similarity of the two abstracts given in section (3.3.1), their content was previously divided as illustrated above. For each abstract, three graphs are constructed and enriched (a graph for each selected zone). For each zone, a similarity value is calculated. The similarity values obtained are very low. This is justified by assigning the terms to a zone while they semantically define another zone, a consequence of the decomposition based on categorization at the sentence level and of the overlapping of zones.

To overcome this problem of overlapping of zones, the terms are assigned to each zone of an abstract according to the overall meaning conveyed by its content. From the global meaning of an abstract, the meaning and the role of its terms are deduced. A term can describe the context of the paper (document categorization, document clustering, image categorization, ontologies enrichment, information retrieval, etc.) or contribution (the methods and algorithms as well as notions used to describe them) or the application domain (classification applied to a given corpus, data mining applied to textual documents, data mining applied to the web, data mining applied to images, etc.). In addition, the terms contained in the title and in the keywords are used, as they often contain information that is not cited in the abstract.

The role of each term is defined according to the knowledge domain in which it is used.

The semantic annotation of the concepts was achieved especially in WordNet Domains [63]. In WordNet Domains, different subject fields are defined, such as medicine, computer science, and architecture. Each synset of WordNet [64] is annotated by one or more Subject Fields where this synset has a meaning. On the basis of the principle that a term describes one of the three zones selected to characterize a scientific abstract, each concept is annotated in the ontology associated with this abstract by one of the three zones (*context*, *contribution* and *application domain*).

The extraction of the concepts corresponding to each zone is performed by projecting the terms composing the content of an abstract on the ontology. The comparison of two abstracts amounts to comparing the zones playing the same role. Three partial similarities are then calculated on the basis of the concepts belonging to the same zone. Two abstracts are compared at three levels. A global similarity of two scientific abstracts *A1* and *A2* is obtained by combining the three partial similarities according to equation (6). The global similarity makes it possible to rank abstracts in descending order of their similarity as illustrated in Tables 10, 11 and 12.

$$\begin{aligned} Sim(A1, A2) = & \\ & \alpha\ sim_{context}(A1, A2) \\ & + \beta\ sim_{\text{contribution}}(A1, A2) \\ & + \gamma\ sim_{\text{applicationdomain}}(A1, A2) \end{aligned} \tag{6}$$

α, β, γ are parameters whose values are between 0 and 1. They define the importance attributed to the *context*, the *contribution* and the *application domain*. α + β+ γ=1.

The documents processed are not necessarily suspicious, since it is possible to implement this approach in comparing a document under review, for example, to an entire corpus, without a priori as to its respect for scientific ethics.

A similarity threshold determined by experimentation and according to the ontology and to the collection of abstracts used determines if a risk of plagiarism exists. Abstracts with high similarity will then require a full review of the entire document.

3.3.1 Example

Figure 8 provides an extract of an ontology associated with the domain *ontologies enrichment* and shows the annotation of the concepts by the three zones defined to characterize the content of a scientific abstract.

Let us consider two abstracts from two scientific papers. These papers published in French were translated for the need of our work. The construction of their graphs and calculation of their partial similarities and global similarity is given in section 3.3.2.

Abstract1: *Ontology enrichment based on sequential pattern.*

The mass of information now available via the web, in constant evolution, requires structuring in order to facilitate access and knowledge management. In the context of the Semantic Web, ontologies aim at improving the exploitation of informational resources, positioning themselves as a model of representation. However, the relevance of the information they contain requires regular updating, and in particular the addition of new knowledge. In this paper, we propose an ontologies enrichment approach based on data mining techniques and more specifically on the search for sequential patterns in textual documents.

The presented approach has been tested and evaluated on an ontology of the water domain, which we have enriched from documents extracted from the Web.

Key words: ontology, enrichment, semantic web, data mining, sequential pattern

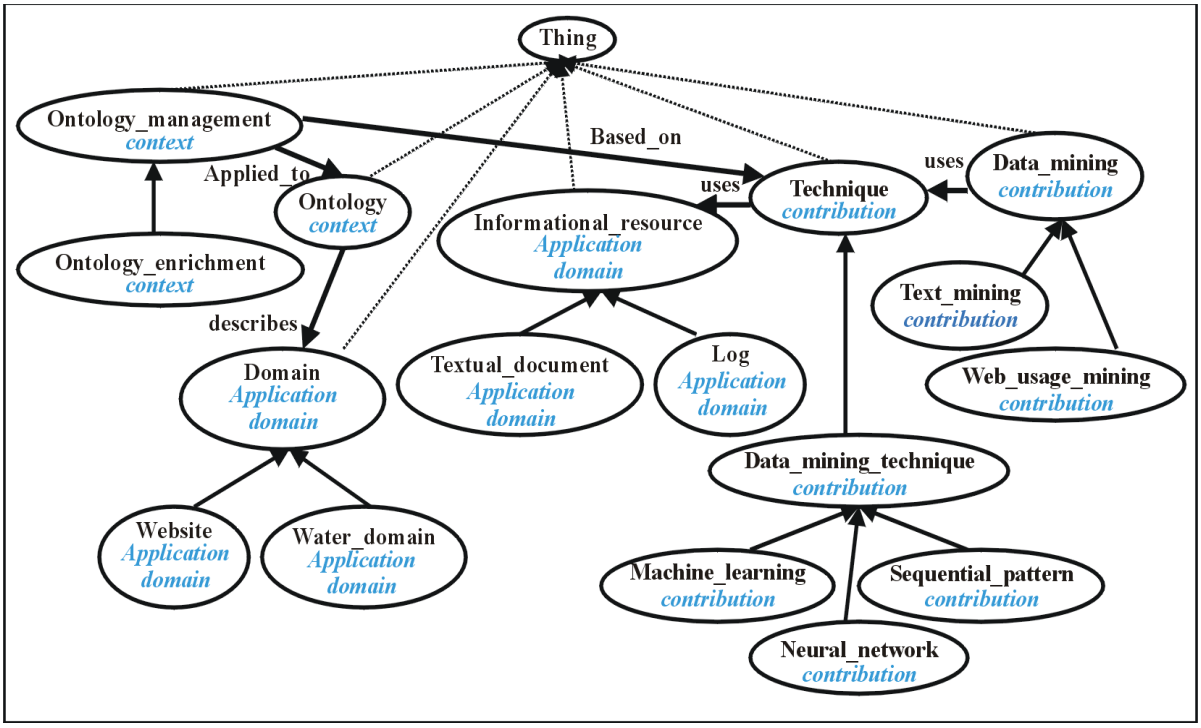


Figure 8: Extract of the *ontologies enrichment* domain ontology, and annotation of concepts by their zone.

Abstract2: *Web usage mining for ontology enrichment.*

Recently, new approaches have integrated the use of data mining techniques in the ontologies enrichment process. Indeed, the two fields, data mining and ontological meta-data are extremely linked: on one hand data mining techniques help in the construction of the semantic Web, and on the other hand the semantic Web assists in the extraction of new knowledge. Thus, many works use ontologies as a guide for the extraction of rules or patterns, allow to discriminate the data by their semantic value and thus to extract more relevant knowledge. It turns out, however, that few works aimed at updating the ontology are concerned with data mining techniques. In this paper, we present an approach to support the onologies management of websites based on the use of Web Usage Mining techniques. The presented approach has been tested and evaluated on an website ontology , which we have constructed and then enriched based on the sequential patterns extracted on the log.

Key words: Semantic Web, ontology, Web Usage Mining, enrichment, data mining, sequential pattern.

3.3.2 Applying our approach

3.3.2.1 Extracting the initial concepts for each abstract

Initial concepts are extracted at the classification step. The two abstracts are attached to the ontology represented in Figure 8. The concepts are assigned to their appropriate zone according to their annotation.

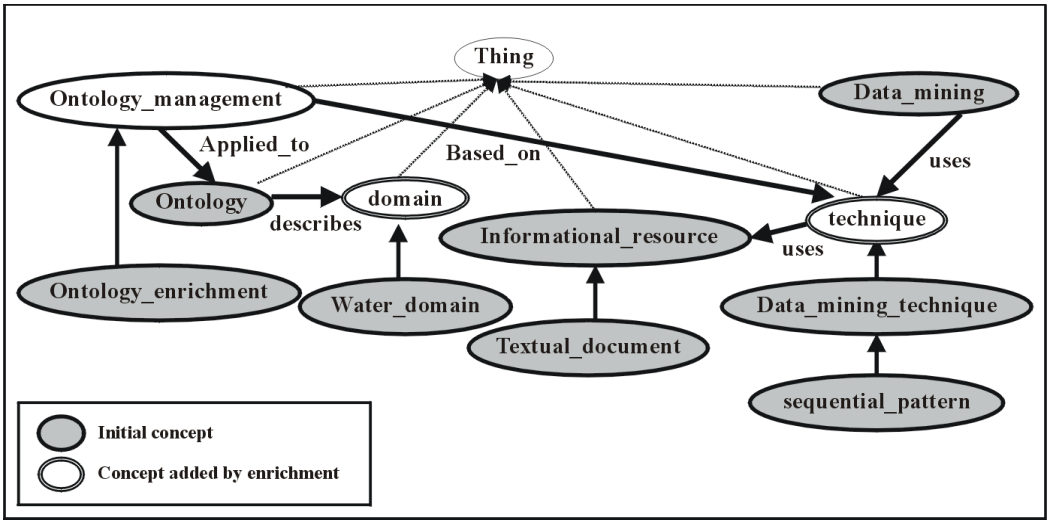


Figure 9: Enriched graph of Abstract1.

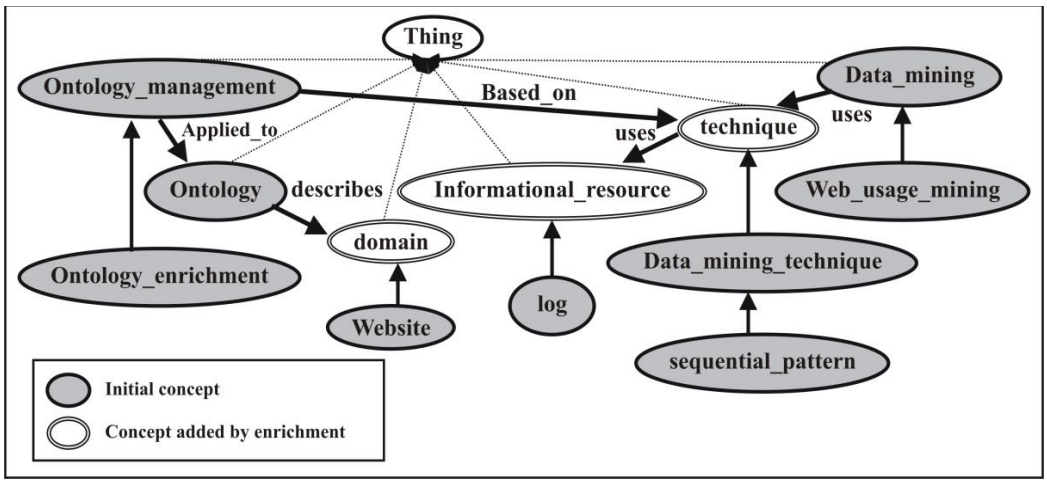


Figure 10: Enriched graph of Abstract2.

	Abstract1		Abstract2	
Zones	**Concepts of Abstract 1**	**Concept type**	**Concepts of Abstract 2**	**Concept type**
context	Ontology_management	Added	Ontology_management	Initial
	Ontology_enrichment	Initial	Ontology_enrichment	Initial
	Ontology	Initial	Ontology	Initial
contribution	Data_mining	Initial	Data_mining	Initial
	Technique	Added	Technique	Added
	Data mining_technique	Initial	Data_mining_technique	Initial
	Sequential_pattern	Initial	Sequential_pattern	Initial
			Web_usage_mining	Initial
Application domain	Informational_resource	Initial	Informational_resource	Added
	Textual_document	Initial	log	Initial
	Domain	Added	Domain	added
	Water_domain	Initial	Website	Initial

Table 5: Distribution by zone of the concepts of Abstract1 and Abstract2.

3.3.2.2 Enrichment of the graphs corresponding to the two abstracts

The *initial concepts* are used to enrich the graphs of the two abstracts by constructing their semantic perimeter and by comparing their graphs. The enriched graphs of the two abstracts *Abstract1* and *Abstract2* are represented in Figure 9 and Figure 10. The distribution by zone of the initial concepts and the added concepts by enrichment is given in Table 5.

3.3.2.3 Similarity calculating between Abstract1 and Abstract2

Table 6 provides values of the global similarity and partial similarities. (Values obtained with $\alpha = 0.35$, $\beta = 0.63$, $\gamma = 0.02$, $g = 0.05$).

$Sim_{context}$(abstract1,abstract2)	0,98
$Sim_{contribution}$ (abstract1,abstract2)	0,59
$Sim_{applicatiodomain}$(abstract1,abstract2)	0,10
Sim (abstract1,abstract2)	0,72

Table 6: Similarities between abstract1 and abstact2.

3.3.2.4 Result

The results obtained indicate that these two abstracts process the same context (sim context = 0.98) with similar approaches. The similarity obtained for the contribution is high (sim contribution = 0.59). These two abstracts differ at the application domain level since the similarity value obtained for this zone is very low (sim application domain = 0.10). The global similarity

obtained is high. This value indicates that the papers associated with these two abstracts should be the subject of a more in-depth analysis that could possibly reveal a case of plagiarism.

4 Experimentations

Our approach is evaluated at two levels. The first evaluation concerns our semantic classification process based on domain ontologies (CBO) and the second concerns the textual similarity calculation process of scientific abstracts.

4.1 Semantic classification process

4.1.1 The data

The implementation of our semantic classification process was performed using WordNet and WordNet Domains simultaneously. In WordNet Domains several knowledge domains are used. These different domains were assimilated to domain ontologies. The Rita similarity measure [13] was used to measure the semantic distance between two synsets in WordNet. The terms within sentences were annotated with their type (noun, verb, adverb and adjective) by Stanford Part-Of-Speech Tagger (POS Tagger) [65].

To evaluate conventional classifiers with our corpus, a pre-processing was performed on the documents. Nouns, verbs and adjectives used in each document were retained. The lemmas relative to these terms were extracted and their weight based on Tf-Idf was then calculated. These lemmas constitute the vector representation of documents. For conventional classifiers, the implementation of three algorithms, SVM, Naive Bayes and decision tree of Weka [66] were used.

Our evaluation covers 10 domains defined in WordNet Domains and a corpus consisting of 976 abstracts of scientific papers. Some abstracts of the domain medicine were extracted from the corpus Muchmore which is a parallel corpus of English-German scientific medical abstracts obtained from the Springer Link web site. All the other abstracts of our corpus were extracted from several scientific journals specialized in the retained domains browsing their Web site. Table 7 gives the distribution of the abstracts relative to the selected domains.

Domains	Number of abstracts
Music	106
Law	83
Computer_science	101
Politics	76
Physics	101
Chemistry	83
Economy	104
Buildings	104
Medicine	117
Mathematics	101
Total	**976**

Table 7: Distribution of abstracts by domains.

4.1.2 Results and discussion

Measures traditionally used in categorization are considered in this work: precision, recall, F-measure and baseline accuracy. The results of our process were compared with those of conventional classifiers. The results obtained are summarized in Table 8.

The recall (*Rc*) determines the number of documents that are correctly classified in a class divided by the total number of documents belonging to that class. Precision (*Pr*) defines the number of documents that are correctly classified in a class divided by the number of documents assigned to that class. A measure that combines precision and recall is their harmonic mean, referred to as the F-measure (*F*). Baseline accuracy (*Acc*) gives the percentage of documents correctly classified relative to the total number of documents in the corpus.

Classes	CBO			Naive Bayes			SVM			Tree C4.5		
	Pr	**Rc**	**F**	**Pr**	**Rc**	**F**	**Pr**	**Rc**	**F**	**Pr**	**Rc**	**F**
Music	0,962	0,943	0,952	0,835	0,906	0,869	0,963	0,981	0,972	0,913	0,887	0,900
Law	0,952	0,964	0,958	0,777	0,880	0,825	0,947	0,867	0,906	0,766	0,711	0,737
Computer_science	0,970	0,950	0,960	0,845	0,861	0,853	0,872	0,941	0,905	0,474	0,644	0,546
Politics	0,949	0,974	0,961	0,788	0,829	0,808	0,944	0,882	0,912	0,754	0,645	0,695
Physics	0,960	0,960	0,960	0,833	0,842	0,837	0,887	0,931	0,908	0,513	0,386	0,441
Chemistry	0,940	0,952	0,946	0,947	0,867	0,906	0,986	0,880	0,930	0,848	0,807	0,827
Economy	0,980	0,962	0,971	0,820	0,788	0,804	0,855	0,904	0,879	0,541	0,442	0,487
Buildings	0,980	0,962	0,971	0,950	0,913	0,931	0,925	0,952	0,938	0,757	0,750	0,754
Medicine	1,000	0,983	0,991	0,982	0,940	0,961	0,991	0,991	0,991	0,894	0,863	0,878
Mathematics	0,925	0,980	0,952	0,904	0,842	0,872	0,898	0,871	0,884	0,493	0,673	0,569
Average	**0,964**	**0,963**	**0,963**	**0,872**	**0,869**	**0,870**	**0,926**	**0,924**	**0,924**	**0,694**	**0,682**	**0,683**
Accuracy	**0,963**			**0,869**			**0,924**			**0,682**		

Table 8: Comparison of the results of the various classifiers.

To calculate these different values for SVM, Naive Bayes, and tree C4.5, cross-validation was performed and the results obtained with the best parameters were retained. Table 8 shows that for our process the values of recall and precision are close. These values are close to 1. This is an indicator of the good performance of our classifier. Considering the average of precisions, recalls and F-measure, our process obtains better results than the three conventional classifiers considered. The best percentage of documents correctly classified relatively to all documents in the corpus is obtained by our semantic classification process.

A Wilcoxon Signed-Rank test was used in order to study the statistical significance of the improvement brought about by our process. The p-value between our system and the three conventional classifiers was calculated. This Wilcoxon Signed-Rank test is based on the values of the F-measure obtained for CBO, SVM, Naive Bayes and tree C4.5. This improvement is considered statistically significant if p-value <0.05 and very significant if p-value <0.01. The results of the test are summarized in Table 9.

	CBO - SVM	**CBO - Naive Bayes**	**CBO - Tree C4.5**
P-value (F-measure)	0.00885858	0.00294464	0.000976562

Table 9: Wilcoxon test result.

The p-values obtained with the Wilcoxon test are all less than 0.01. These are very significant p-values. This allows us to conclude that our system significantly improves the classification process of documents compared to conventional classifiers at the threshold $\alpha = 0.01$.

The three conventional classifiers have in common the representation of the documents by words independent of each other as well as a morphological comparison of the words belonging to the documents. The comparison is performed at the word level, whereas in our process, the comparison is performed at the overall context level of the document. A document is represented by the domain described in its content. This domain is deduced by the words of the document taken together considering their relationships in the context in which they appear. In addition, our process is built from domain ontologies, which is a more stable base than a training collection. Indeed, a modification in the choice of the documents constituting this training collection leads to a modification of the results of conventional classifiers.

4.2 Semantic similarity process of scientific abstracts

4.2.1 The data

Our implementation was extended by adding processes to build the semantic perimeters, to divide scientific abstracts into three zones and to compare graphs. To evaluate our approach defining the semantic similarity of scientific abstracts, we constructed an ontology representing the domain of *automatic classification of documents*. To construct our corpus, a set of scientific abstracts related to this domain was extracted from the web. In our different tests, the abstract, the title of the paper and the keywords were taken into account. Each abstract was compared with all the abstracts in the corpus. The abstracts were compared in pairs. For example, the results were obtained by comparing twenty abstracts for which 190 comparisons were made. The construction of the initial graph, the semantic perimeter of each abstract and the comparison of the graphs is done according to the process defined in the previous sections.

Each concept of our ontology was annotated by one of the three selected zones characterizing the content of the scientific abstracts: *context*, *contribution* and *application domain*. This annotation is performed according to the role that each concept plays depending on the chosen domain. For example, *clustering*, *classification* and *document* concepts are annotated by the *context* zone, the concepts representing the different algorithms and methods used by the authors as well as all the concepts describing these methods are annotated by the *contribution* zone. The concepts representing the type of document (*Text*, *Web*) and the corpus used are annotated by the *application domain* zone.

Our approach was compared to two existing approaches.

The first approach is based on a vector representation of the content of the text: *Bag-of-words*.

The process of extracting terms is similar to the one performed in section 4.1.1. An abstract vector contains the lemmas corresponding to the nouns, verbs and adjectives extracted from the text. Lemmas are represented by their weight based on Tf-Idf. The similarity of two abstracts is calculated by measuring the cosine of the angle between their respective vectors.

The second *n-grams* approach is based on the representation of an abstract by a set of words called *n-grams*. The text is divided into a set of n-grams. The size of an n-gram is determined by a chosen number of consecutive characters, *n*. Several values of *n* were tested (n= 2, 4 and 8) and for each, the similarity between two abstracts was calculated using equation (7) [51] [52] and (8) [53]. For any pair of abstracts *x* and *y*, the similarity Sim(*x*,*y*) is computed as bellow :

$$Sim(x,y) = \frac{1}{|Dn(x)| + |Dn(y)|} \times \sum_{w \in Dn(x) \cup Dn(y)} \frac{(f_y(w) - f_x(w))^2}{(f_y(w) + f_x(w))^2} \quad (7)$$

$$Sim(x,y) = \frac{|x \cap y|}{|x \cup y|} \quad (8)$$

w denotes an arbitrary n-gram, fx(w) denotes the relative frequency with which w appears in the abstract x

Text1	Text2	Similarity			
		context	contribution	application domain	global
A1.clustering	A3.clustering	1,000	0,401	1,000	0,622
A1.clustering	A10.clustering	1,000	0,295	0,157	0,539
A1.clustering	A2.clustering	0,982	0,306	0,065	0,538
A1.clustering	A9.clustering	1,000	0,227	0,153	0,496
A1.clustering	A16.clustering	1,000	0,169	0,065	0,458
A1.clustering	A15.clustering	1,000	0,103	0,237	0,419
A1.clustering	A17.clustering	1,000	0,092	0,345	0,415
A1.clustering	A5.clustering	1,000	0,095	0,237	0,414
A1.clustering	A18.clustering	1,000	0,022	0,353	0,371
A1.clustering	A19. classif-clust	0,558	0,016	0,065	0,207
A1.clustering	A14.classification	0,244	0,125	0,431	0,172
A1.clustering	A6.classification	0,240	0,074	0,016	0,131
A1.clustering	A8.classification	0,225	0,060	0,541	0,127
A1.clustering	A7.classification	0,225	0,060	0,065	0,118
A1.clustering	A4.classification	0,244	0,036	0,065	0,109
A1.clustering	A11.classification	0,237	0,034	0,108	0,107
A1.clustering	A13.classification	0,230	0,014	0,065	0,090
A1.clustering	A12.classification	0,231	0,007	0,125	0,088
A1.clustering	A20.classification	0,237	0,005	0,031	0,087

Table 10: similarities between A1 and the others abstracts.

Text1	Text2	Similarity			
		context	contribution	Application domain	global
A12.classification	A13.classification	1,000	0,015	0,000	0,360
A12.classification	A4.classification	0,966	0,032	0,000	0,358
A12.classification	A20.classification	0,964	0,012	0,483	0,355
A12.classification	A6.classification	0,965	0,015	0,193	0,351
A12.classification	A14.classification	0,966	0,012	0,066	0,347
A12.classification	A11.classification	0,964	0,007	0,023	0,342
A12.classification	A8.classification	0,900	0,005	0,185	0,322
A12.classification	A7.classification	0,900	0,005	0,000	0,318
A12.classification	A19.classif-clust	0,541	0,107	0,000	0,257
A12.classification	A5.clustering	0,234	0,027	0,329	0,105
A12.classification	A3.clustering	0,234	0,032	0,125	0,105
A12.classification	A18.clustering	0,234	0,026	0,125	0,101
A12.classification	A17.clustering	0,234	0,024	0,123	0,100
A12.classification	A9.clustering	0,227	0,019	0,189	0,095
A12.classification	A15.clustering	0,231	0,004	0,329	0,090
A12.classification	A1.clustering	0,231	0,007	0,125	0,088
A12.classification	A2.clustering	0,233	0,005	0,000	0,085
A12.classification	A16.clustering	0,231	0,006	0,000	0,085
A12.classification	A10.clustering	0,227	0,004	0,032	0,083

Table 11: Similarities between A12 and the others abstracts.

and Dn(x) represents the so called n-gram dictionary of x. | | is the number of n-grams.

The best results were obtained with n = 8 and equation (8), for which the fewest erroneous matching was noted.

4.2.2 Results and discussion

Parameter values (α, β and γ) depend on the ontology and on the corpus used. Several values for these parameters were tested. The goal of this study is to attribute more importance to the *context* zone and the *contribution* zone since it aims to look for matches that primarily indicate documents dealing with the same *context* and similar *contributions*. The following values were retained: α = 0.35, β = 0.63, γ = 0.02, g = 0.05. These values led to the abstracts being grouped based on their context. Table 10 and Table 11 provide the results obtained when comparing respectively the abstracts A1 and A12 with the other abstracts. These tables provide the three partial similarities computed for each pair of abstracts as well as their global similarity. The results, ranked in descending order of global similarity, show a grouping of the abstracts by *context*. Abstract *A1* deals with the *document clustering* context. Abstracts that have the highest similarity with A1 correspond to this context. The abstract *A12* deals with the *document classification* context. Abstracts that have the highest similarity with A12 also correspond to this context hers abstracts.

Table 10 provides a comparison of the similarities between A1 and the other abstracts at three levels. Their similarity can be compared at the *context* level, at the *contribution* level and at the *application domain* level. The values obtained comparing A1 with A3 indicate that these two abstracts deal with the same context (sim context = 1), present similar contributions (Sim contribution = 0, 401) and apply their approach to the same domain (sim application domain = 1). The value of their global similarity is high. These values enable us to retain these two abstracts as suspicious documents, thus requiring further reading and analysis of their entire contents.

Table 11 provides a comparison of the similarities between A12 and the other abstracts at three levels. For the last ten rows of Table 11, very low partial and global similarities were obtained. The first eight rows of Table 11 show that the corresponding abstracts deal with the same context as abstract A12 (sim context >= 0.900) but use different approaches (sim contribution <= 0.032). Their global similarity is low (<= 0,360). This enables us to conclude that abstract A12 does not present any risk of plagiarism with the other abstracts.

The goal of our approach is to be able to find suspicious documents; that is, documents with high similarities. To find these documents, a threshold for the calculated similarity values is determined by experimentation.

To compare the results obtained with our approach to those of *Bag-of-words* and *n-grams*, similarities between the different abstracts of our corpus using the *Bag-of-words* and *n-grams* approaches were calculated. The abstracts were then ranked in descending order of their similarity. For these two approaches, several erroneous matching were found. Table 12, gives an example of the comparison of the similarities between A4 and the other abstracts obtained by our approach, and the *Bag-of-words* and *n-grams* approaches. A4 deals with the context *classification*. With *Bag-of-word* and *n-grams* approaches, most of the abstracts semantically closest to A4 deal with the *clustering* context.

For the *Bag-of-words* approach, abstracts belonging to the context *clustering* (A10, A3, A2, A5, A15, A1) obtain a better similarity score than those (A11, A8, A12, A20, A7, A14) that deal with the same context that A4. It is the same for the *n-grams* approach. Abstracts

Text1	Text2	Our approach	Bag-of-words		N-grams	
A4.classification	A6.classification	0.417272	A06.classification	0.125685	A11.classification	0,042080
A 4.classification	A11.classification	0.401363	**A10.clustering**	0.108323	**A18.clustering**	0,038287
A 4.classification	A13.classification	0.373563	A13.classification	0.097182	**A03.clustering**	0,036313
A 4.classification	A12.classification	0.358287	A19.classif-clust	0.095763	A06.classification	0,035757
A 4.classification	A14.classification	0.358132	**A03.clustering**	0.092988	**A10.clustering**	0,035634
A 4.classification	A7.classification	0.353878	**A02.clustering**	0.092751	A08.classification	0,035602
A 4.classification	A20.classification	0.353120	**A05.clustering**	0.089178	A12.classification	0,034261
A 4.classification	A8.classification	0.330633	**A15.clustering**	0.073636	**A01.clustering**	0,033475
A 4.classification	A19.classif-clust	0.257688	**A01.clustering**	0.066826	A19.classif-clust	0,033400
A 4.classification	A5.clustering	0.191517	**A11.classification**	0.061259	**A07.classification**	0,033071
A 4.classification	A3.clustering	0.180843	**A08.classification**	0.045829	A17.clustering	0,032417
A 4.classification	A9.clustering	0.176679	A18.clustering	0.043951	A09.clustering	0,029097
R4.classification	A2.clustering	0.175801	**A12.classification**	0.042752	A15.clustering	0,026786
R4.classification	A15.clustering	0.147094	A16.clustering	0.041947	A05.clustering	0,025901
A4.classification	A10.clustering	0.135412	**A20.classififcation**	0.033817	**A13.classification**	0,025269
A4.classification	A18.clustering	0.129238	**A07.classification**	0.031982	**A14.classification**	0,023015
A4.classification	A17.clustering	0.119075	A17.clustering	0.028876	A02.clustering	0,020426
A4.classification	A16.clustering	0.114507	**A14.classification**	0.026670	A16.clustering	0,018511
A4.classification	A1.clustering	0.109055	A09.clustering	0.023351	**A20.classification**	0,015968

Table 12: Similarities between A4 and the others abstracts using our approach, Bag-of-words, and N-grams

Abstracts	P5			R-precision		
	Bag-of-words	**N-grams**	**Our approach**	**Bag-of-words**	**N-grams**	**Our approach**
A1	1,000	1,000	1,000	0,800	1,000	1,000
A2	0,800	1,000	1,000	0,800	1,000	1,000
A3	0,800	1,000	1,000	0,800	0,900	1,000
A4	0,600	0,400	1,000	0,333	0,556	1,000
A5	0,800	0,600	1,000	0,900	0,800	1,000
A6	1,000	1,000	1,000	0,667	0,778	1,000
A7	0,800	0,800	1,000	0,778	0,667	1,000
A8	0,800	0,800	1,000	0,778	0,556	1,000
A9	0,800	1,000	1,000	0,900	0,900	1,000
A10	0,800	1,000	1,000	0,800	0,900	1,000
A11	1,000	1,000	1,000	0,778	0,889	1,000
A12	0,800	0,800	1,000	0,778	0,667	1,000
A13	0,800	0,800	1,000	0,667	0,667	1,000
A14	1,000	1,000	1,000	0,778	0,667	1,000
A15	0,800	1,000	1,000	0,800	1,000	1,000
A16	1,000	1,000	1,000	0,800	0,900	1,000
A17	0,600	1,000	1,000	0,700	0,900	1,000
A18	0,800	1,000	1,000	0,600	0,800	1,000
A19	1,000	1,000	1,000	1,000	1,000	1,000
A20	0,800	0,800	1,000	0,778	0,667	1,000
Average	**0,840**	**0,900**	**1,000**	**0,762**	**0,811**	**1,000**

Table 13: Precision values for, Bag-of-words, n-grams and our approach.

belonging to the context *clustering* (A18, A3, A10, A1) obtain a better similarity score than those (A7, A13, A14, A20) that deal with the same context as A4.

For all the comparisons made between the abstracts in the corpus, our approach is able to correctly rank the abstracts by context as shown in Table 10, 11, 12 and 13. Clustering and classification are two different contexts. For these two contexts, the methods and algorithms used are different. For that reason, the similarity between two abstracts belonging to these two contexts must be low (low context similarity and low contribution similarity) and, therefore, the risk of plagiarism is very low, or even non-existent. To determine which approach performs the correct matching between abstracts of our corpus, the precision P5 and the R-precision for each approach and for each abstract were computed.

An abstract *Ab1* is assumed relevant to an abstract Ab2, if Ab1 deals with the same context as *Ab2*. Precision Px at point x (x=5, R) is the ratio of the relevant abstracts among the first x returned ones. *R* in the R-precision represents the number of the relevant abstracts to a given abstract in the corpus. Table 13 summarizes the different values.

Our process obtains better results than *Bag-of-words* and *n-grams* approaches. Our process is able to match correctly abstracts dealing with the same context and, therefore, it is more precise than the other approaches.

The Wilcoxon Signed-Rank test was used in order to study the statistical significance of the improvement brought about by our process. The p-value between our system and the two other approaches was calculated.

The results of the Wilcoxon test are summarized in Table 14. The p-values obtained with the Wilcoxon test are all less than 0.01. These are very significant p-values. This leads us to conclude that our system is able to match abstracts by context more correctly than the bag-of-word and n-grams approaches. Others results are summarized in Table 15.

	Our approach / Bag-of-word	**Our approach / n-grams**
P-value at P5	0.000213431	0.0089409
P-value at R-precision	0.0000638361	0.000219794

Table 14: Wilcoxon test result.

- The content of abstracts *A1*, *A2*, *A3* and *A10* indicates great similarity between abstracts (A1-A3) and (*A2-A10*). These two pairs of abstracts deal with the same context, use the same algorithms and use ontologies to solve similar problematic a priori. As shown in Table 15, our approach makes it possible to select these abstracts as suspicious, while the Bag-of-words and n-grams approaches select only the abstracts (*A1-A3*). *A1* and *A3* use almost the same words in their content. As for the abstracts *A2* and *A10*, their content is described with different words and different sentences, but both are interested in ontology-based feature selection and use the

Text1	Text2	Our approach				Bag-of-Words	N-grams
		context	contribution	Application domain	global		
A1.clustering	A3.clustering	1.000000	0.400673	1.000000	0.622424	0.724688	0,352187
A2.clustering	A10.clustering	0.982456	0.486622	0.112994	0.652692	0.198869	0,050761
A15.clustering	A16.clustering	1.000000	0.188889	0.000000	0.469000	0.470623	0,108580

Table 15: Comparison between Bag-of-words, N-grams and our approach.

same clustering algorithm. Our approach is able to capture the meaning of the abstract and, therefore, retains these two abstracts for a complete examination of their corresponding papers.

- The *Bag-of-words* approach indicates a matching between abstracts *A15* and *A16*. These two abstracts have a high similarity whereas the authors of these two abstracts use different methods in their contribution. Our approach has the advantage of comparing abstracts at three levels. For our approach, the *contribution* similarity between *A15* and *A16* indicates a very low value, which means that the methods used by the authors to solve their problematic are different. This makes it possible to conclude that even if these two abstracts present similar contexts, the risk of plagiarism is low.

Our approach assesses the similarity of texts in two steps. The documents are first assigned to a domain ontology that best describes their content. This overall similarity is achieved by a semantic classification process. This process emphasizes the overall context of the document that can be deduced from the terms of the document taken together, unlike conventional classifiers that consider words independently of each other. For documents belonging to the same ontology, a "local" similarity is calculated. This similarity is based on graphs corresponding to the texts. The enrichment of the graphs through the construction of the semantic perimeter of the texts and comparing of their graphs makes it possible to deduce a similarity not explicitly cited in the texts. The similarity calculation of scientific abstracts is refined by dividing their contents into three zones. Partial similarity values are then calculated. This helps to bring out the notions common to both texts. A grouping by context and a ranking in descending order of the global similarity value can be achieved by combining the three partial similarities. The objective of our approach is to find suspicious documents. It has the advantage of comparing the content of the documents based on three levels. The examination of the similarity obtained for each zone makes it possible to conclude on the existence of a risk of plagiarism.

5 Conclusion

The approach proposed in this paper is meant to assess text similarity. This similarity is based on an overall similarity calculation obtained by a classification process. Our classification process is based on domain ontologies and takes into account the relationships between the terms relative to their context of appearance in the document. The evaluation of our process showed better results than those of conventional classifiers. The construction of the semantic perimeter and the comparison of the graphs of texts based on the domain ontology to which they are attached make it possible to enrich the graphs and to deduce implicit information. Our approach thus present the advantage of taking into account the synonymy and polysemy present in a language and of deducing a similarity between two texts not explicitly cited in their content.

Assessing the similarity between the scientific texts represented by their abstracts is our main interest. In the process of semantic comparison, three distinct parts were defined to structure the abstracts of scientific texts: *context*, *contribution* and *application domain* and three partial similarities were calculated. The comparison of two scientific abstracts is then performed at three levels. The global similarity value of two abstracts, calculated by combining partial similarities, makes it possible to rank the abstracts in descending order of their global similarity. A threshold applied to the calculated similarities is useful in finding suspicious documents and highlighting a risk of plagiarism. Tests were performed on a set of scientific abstracts. The enrichment of the graphs makes it possible to bring out common notions not explicitly cited. Moreover, dividing the contents of abstracts into three distinct zones helps in extracting the notions relative to the *context*, *contribution* and *application domain* and thus makes comparisons between zones of the same type. An evaluation can be made to determine whether two abstracts deal with the same context, whether their contributions are similar and whether they apply their approach to the same application domain.

The quality of our process depends on domain ontologies that must cover the entire vocabulary of the knowledge domain represented for the process to be effective. This may constitute a limitation of this work since the process used does not support the building of domain ontologies. It is, therefore, assumed that they are available. Even if this can be assumed for scientific texts or abstracts structured as shown in this work, the process obviously needs to be refined for it to be used in comparing general texts. Indeed, one of the ways of improving our approach is to generalize the concept of semantic perimeter so as to consider any text rather than just scientific abstracts.

6 References

[1] P. Resnik (1999). Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural

language. Journal of Aritificial Intelligence Research, Vol.11, Issue 1, pp. 95-130. https://doi.org/10.1613/jair.514

[2] J. Curran (2002). *Ensemble methods for automatic thesaurus extraction*. In Proceedings of the conference on Empirical methods in natural language processing (EMNLP), Philadelphia, Vol.10, pp. 222-229.

[3] P. Cimano, S. Handschuh, and S. Staab (2004). *Towards the self-annotating web*. In Proceedings of the 13th international conference on World Wide Web, New York, USA, pp. 462-471.

[4] Z.S. Harris (1954). *Distributional structure*. Word, Vol. 10, Issue 2-3, pp. 146–162. https://doi.org/10.1080/00437956.1954.11659520

[5] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman (1990). *Indexing by latent semantic analysis*. Journal of the American Society of Information Science, Vol. 41, Issue 6, pp. 391–407. https://doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9

[6] E. Gabrilovich and S. Markovitch (2007). *Computing semantic relatedness using Wikipedia-based explicit semantic analysis*. In Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, pp. 1606–1611.

[7] M.Yazdani and A. Popescu-Belis (2013). *Computing text semantic relatedness using the contents and links of a hypertext encyclopedia: extended abstract*. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China, pp. 3185–3189.

[8] J. Turian, L. Ratinov and Y. Bengio (2010). *Word representations: a simple and general method for semi-supervised learning*. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, pp. 384–394.

[9] M. Baroni, G. Dinu and G. Kruszewski (2014). *Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors*. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Volume 1, Baltimore, Maryland, pp. 238–247.

[10] C. Leacock, G. A. Miller, and M. Chodorow (1998). *Using corpus statistics and WordNet relations for sense identification*. Journal of Computational Linguistics, Vol 24, Issue 1, pp. 147-165.

[11] R. Rada, H. Mili, E. Bicknell and M. Blettner (1989). *Development and application of a metric on semantic nets.* IEEE Transactions on systems, Man and Cybernetics, Vol 19, Issue 1, pp.17-30. https://doi.org/10.1109/21.24528

[12] Z. Wu and M. Palmer (1994). *Verb semantics and lexical selection*. In Proceedings of the 32nd Annual Meetings of the Associations for Computational Linguistics, Las Cruces, New Mexico, pp. 133-138. https://doi.org/10.3115/981732.981751

[13] D. C. Howe (2009). *RiTa: creativity support for computational literature*. In Proceedings of the seventh ACM conference on Creativity and cognition (C&C '09), Berkeley, California, USA, pp. 205-210.

[14] D. Lin (1998). *An information-theoric definition of similarity*. In Proceedings of the 15th international conference on Machine Learning, pp. 296-304.

[15] P. Resnik (1995). *Using information content to evaluate semantic similarity in a taxonomy*. In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Vol 1, Montreal, Quebec, Canada, pp. 448-453.

[16] S. P. Ponzetto and M. Strube (2007). *Knowledge derived from Wikipedia for computing semantic relatedness*. Journal of Artificial Intelligence Research, Vol 30, Issue 1, pp. 181–212. https://doi.org/10.1613/jair.2308

[17] D. Milne, I. H. Witten (2008). *Learning to link with Wikipedia*. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, California, USA, pp. 509–518.

[18] M. T. Pilehvar and R. Navigli (2015). *From senses to texts: An all-in-one graph-based approach for measuring semantic similarity*. Journal of Artificial Intelligence Vol. 228, pp. 95–128. https://doi.org/10.1016/j.artint.2015.07.005

[19] G. Salton and M.J. McGill (1983). *Introduction to modern information retrieval*. McGraw-Hill computer Science Series.

[20] G. Salton (1971). *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice-Hall.

[21] C.J. Crouch, S. Apte, et H. Bapat (2002). *Using the extended vector model for xml retrieval*. In Proceedings of the First Workshop of the Initiative for the Evaluation of XML Retrieval (INEX), Schloss Dagstuhl, pp. 95-98.

[22] E.A. Fox (1983). *Extending the Boolean and Vector Space Models of information retrieval with p-norm queries and multiple concept types*. PhD thesis, Department of Computer Science, Cornell University.

[23] D. Carmel, Y. Maarek, M. Mandelbrod, Y. Mass and A. Soffer (2003). *Searching xml documents via xml fragments*. In Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, Toronto, Canada, pp. 151– 158. https://doi.org/10.1002/asi.10060

[24] M. Fuller, E. Mackie, R. Sacks-Davis, and R. Wilkinson (1993). *Structural answers for a large structured document collection*. In Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, Pitthsburgh, pp. 204–213.

[25] T. Schileder and H. Meus (2002). *Querying and ranking XML documents*. Journal of the American Society for Information Science and Technology, Vol. 53, Issue 6, pp. 489–503.

[26] T. Joachims (1997). *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. In Proceedings of the Fourteenth

International Conference on Machine Learning, Tennessee, pp.143-151.

[27] S. Jaillet, A. Laurent and M. Teisseire (2006). *Sequential patterns for text categorization*. Journal of Intelligent Data Analysis, IOS Press, Vol.10, issue 3, pp.199–214.

[28] P. Soucy, G. W. Mineau (2001). *A Simple k-NN Algorithm For Text Categorization*. In Proceedings of IEEE International Conference on Data Mining, San Jose, USA, pp.647–648.

[29] A. Hotho, A. Maedche and S. Staab (2002). *Ontology-based Text Document Clustering*. KI, Vol. 16, Issue 4, pp. 48-54.

[30] S. B. Kotsiantis (2007). *Supervised Machine Learning: A Review of Classification Techniques*. Informatica Vol. 31, Issue 3, pp. 249-268.

[31] Y. Yang and X. Liu (1999). *A re-examination of text categorization methods*. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, Berkley, pp. 42–49.

[32] T. Joachims (1998). *Text categorization with support vector machines: learning with many relevant features.* In Proceedings of ECML-98, 10th European Conference on Machine Learning, Chemnitz, Germany, pp. 137–142.

[33] E. Gabrilovich and S. Markovitch (2005). *Feature Generation for Text categorization Using World Knowledge*. In Proceedings of IJCAI 2005: the Nineteenth International Joint Conference on Artificial Intelligence, Edinburgh, Scotland, UK, pp. 1048-1053.

[34] A. Hotho, S. Staab and G. Stumme (2003). *Ontologies Improve Text Document Clustering*. In Proceedings of ICDM:3rd IEEE International Conference on Data Minin, Melbourne, FL, USA, pp. 541-544.

[35] H. H. Tar and T.T. Soe.Nyunt (2011). *Ontology-Based Concept Weighting for Text documents*. International Conference on Information Communication and Management IPCSIT vol.16, IACSIT Press, Singapore.

[36] B. Pincemin (2000). *Similarites texte–texts expérience d'une application de diffusion ciblée et propositions*. In Matemáticas y Tratamiento de Corpus, Actes du 2ème séminaire de l'Ecole interlatine de linguistique appliquée,San Millán de la Cogolla, Logroño, Espagne, Logroño : Fundación San Millán de la Cogolla, 2002, pp 35-52.

[37] K. Beyer, J. Goldstein, R. Ramakrishnan and U. Shaft (1999). *When is `nearest neighbor' meaningful*. In Proceedings of ICDT, International Conference on Database Theory, pp. 217-235. https://doi.org/10.1007/3-540-49257-7_15

[38] U.L.D.N. Gunasinghe, W.A.M. De Silva, N.H.N.D. de Silva, A.S. Perera, W.A.D. Sashika and W.D.T.P. Premasiri (2014). *Sentence similarity measuring by vector space model*. In Proceedings of the 14 th International Conference on Advances in ICT for Emerging Regions (ICTer), Colombo, Sri Lanka, pp. 185-189.

[39] Y. Liu, C. Sun, L. Lin, Y. Zhao and X. Wang (2015). *Computing Semantic Text Similarity Using Rich Features*. In Proceedings of PACLIC: 29th Pacific Asia Conference on Language, Information and Computation, Shanghai, China, pp. 44 – 52.

[40] J. Lewis, S. Ossowski, J. Hicks, M. Errami and H. R. Garner (2006). *Text similarity: an alternative way to search MEDLINE*. Bioinformatics Vol. 22, Issue 18, pp. 2298–2304. https://doi.org/10.1093/bioinformatics/btl388

[41] E. Yamamoto, M. Kishida, Y. Takenami, Y. Takeda and K. Umemura (2003). *Dynamic programming matching for large scale information retrieval*. In Proceedings of the Sixth International Workshop on Information Retrieval with Asian Languages, Vol.11, Sapporo, Japan, pp. 100–108. https://doi.org/10.3115/1118935.1118948

[42] W. Ma and T. Suel (2016). *Structural Sentence Similarity Estimation for Short Texts*. In Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference, Florida, pp. 232–237.

[43] D. Dudognon, G. Hubert and B. Ralalason (2010). *Proxigénéa : Une mesure de similarite conceptuelle*. In Proceedings of the Colloque Veille Strategique Scientifique et Technologique (VSST 2010).

[44] M. Baziz, M. Boughanem, H. Prade and G. Pasi (2005). *A Fuzzy Set Approach to Concept-based Information Retrieval*. In Proceedings of the 4th Conference of the European Society for Fuzzy Logic and Technology and the 11ème Eleventh Rencontres Francophones sur la Logique Floue et ses Applications (Eusflat-LFA 2005 joint Conference), Barcelona, Spain, pp. 1287–1292.

[45] K. M. Shenoy, K.C. Shet, U.D. Acharya (2012). *Semantic plagiarism detection system using ontology mapping*. Advanced Computing: An International Journal (ACIJ), Vol.3, Issue 3, pp. 59–62.

[46] L. Zhang, C. Li, J. Liu and H. Wang (2011). *Graph-Based Text Similarity Measurement by Exploiting Wikipedia as Background Knowledge*. International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol.5, Issue 11, pp. 1328–1333.

[47] W. Jin and R. K. Srihari (2007). *Graph-based Text Representation and Knowledge Discovery*. In Proceedings of the 2007 ACM symposium on Applied computing, Seoul, Korea, pp. 807-811. https://doi.org/10.1145/1244002.1244182

[48] P. Wang, H. Zhang, B. Xu, C. Liu, and H. Hao (2014). *Short Text Feature Enrichment Using Link Analysis on Topic-Keyword Graph*. In Proceedings of Natural Language Processing and Chinese Computing, Springer, pp. 79–90.

[49] J. Leskovec and J. Shawe-Taylor (2005). *Semantic text features from small world graphs*. Workshop on Subspace, Latent Structure and Feature Selection techniques: Statistical and Optimization perspectives, Bohinj.

[50] S. Brin, J. Davis and H. Garcia-Molina (1995). *Copy detection mechanisms for digital documents*. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, California, pp. 398–409. https://doi.org/10.1145/223784.223855

[51] C. Basile, D. Benedetto, E. Caglioti, and M. D. Esposti (2008). *An example of mathematical authorship attribution*. Journal of Mathematical Physics, Vol. 49, Issue 12, pp. 125211-1–125211-20. https://doi.org/10.1063/1.2996507

[52] C. Basile, D. Benedetto, E. Caglioti, G. Cristadoro and M. D. Esposti (2009). *A plagiarism detection procedure in three steps: selection, matches and squares*. 3rd Workshop on Uncovering Plagiarism, Authorship and Social Software Misuse, PAN 2009.

[53] B. Stein, S.M. zu Eissen (2005). *Near Similarity Search and Plagiarism Analysis*. In Proceeding of the 29th Annual Conference of the GfKl Springer, pp. 430-437.

[54] R. Lukashenko, V. Graudina and J. Grundspenkis (2007). *Computer-Based Plagiarism Detection Methods and Tools: An Overview*. In Proceeding of the 2007 International Conference on Computer Systems and Technologies - CompSysTech'07, Bulgaria, article N° 40. https://doi.org/10.1145/1330598.1330642

[55] K. Vani, D. Gupta (2015). *Investigating the Impact of Combined Similarity Metrics and POS tagging in Extrinsic Text Plagiarism Detection System*. In Proceeding of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, pp. 1578-1584.

[56] A. H. Osman, N. Salim, M. S. Binwahlan, H. Hentably and A. M. Ali (2011). *Conceptual similarity and graph-based method for plagiarism detection*. Journal of Theoretical and Applied Information Technology, Vol. 32, Issue 2, pp. 135-145.

[57] D. Rusu, B. Fortuna, M. Grobelnik and D. Mladenić (2009). *Semantic Graphs Derived from Triplets with Application in Document Summarization*. Informatica Vol.33, Issue 3, pp. 357–362.

[58] S. Iltache, C. Comparot, M. Si Mohammed and P. J. Charrel (2016). *Using domain ontologies for classification and semantic interpretation of documents*. In Proceedings of ALLDATA 2016: 2nd International Conference on Big Data, Small Data, Linked Data and Open Data, pp. 76-81.

[59] R. Bendaoud, (2009). *Analyses formelle et relationnelle de concepts pour la construction d'ontologies de domaines à partir de ressources textuelles hétérogènes*. PhD thesis, Henri Poincaré University, Nancy 1.

[60] N. Fuhr and K. Grossjohann (2001). *XIRQL: a query language for information retrieval in XML documents*. In Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval, New Orleans, Louisiana, USA, pp. 172-180.

[61] E. Omodei, Y. Guo, J. P. Cointet and T. Poibeau, (2014). *Analyse discursive automatique du corpus ACL Anthology*. In : Actes de la 21ème conférence Traitement Automatique des Langues Naturelles, Marseille.

[62] Y. Guo, A. Korhonen and T. Poibeau (2011). *A Weakly-supervised Approach to Argumentative Zoning of Scientific Documents.* In Proceedings of the 2011 conference on Empirical Methods in Natural Language Processing, Edinburgh, UK, pp. 273–283.

[63] B. Magnini and G. Cavaglià (2000). *Integrating Subject Field Codes into WordNe*t. In Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation, Athens, Greece, pp. 1413-1418.

[64] C. Fellbaum (1998). *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge MA.

[65] K. Toutanova, D. Klein, C. Manning, and Y. Singer (2003). *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*. In Proceedings of HLT-NAACL, pp. 252-259.

[66] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten (2009). *The WEKA Data Mining Software: An Update*. SIGKDD Explorations, Vol. 11, Issue 1. pp. 10-18. https://doi.org/10.1145/1656274.1656278

Improved Local Search Based Approximation Algorithm for Hard Uniform Capacitated k-Median Problem

Sapna Grover, Neelima Gupta and Aditya Pancholi
Department of Computer Science, University of Delhi, India
E-mail: sgrover@cs.du.ac.in, ngupta@cs.du.ac.in, apancholi@cs.du.ac.in

Keywords: NP completeness, approximation algorithm, k-median problem

Received: January 7, 2017

In this paper, we study the hard uniform capacitated k - median problem. We give $(5+\epsilon)$ factor approximation for the problem using local search technique, violating cardinality by a factor of 3. Though better results are known for the problem using LP techniques, local search algorithms are well known to be simpler. There is a trade-off viz-a-viz approximation factor and cardinality violation between our result and the result of Korupolu et al. [10] which is the only result known for the problem using local search. They gave $(1+\alpha)$ approximation factor with $(5+5/\alpha)$ factor loss in cardinality. In a sense, our result is an improvement as they violate the cardinality by more than a factor of 6 to achieve 5 factor in approximation. Though in their result, the approximation factor can be made arbitrarily small, cardinality loss is at least 5 and small approximation factor is obtained at a big loss in cardinality. Thus, we improve upon their result with respect to cardinality.

Povzetek: Obravnavan je NP problem optimiranja iskanja k median in predlagana izvirna rešitev, ki dosega boljše rezultate v določenih primerjavah.

1 Introduction

k - Median Problem is one of the well studied NP-hard optimization problem. The input instance consists of a set of clients, a set of facilities, a non-negative number k and a non-negative cost of connecting a facility to a client. The goal is to select a set of at most k facilities as centers and assign clients to them such that the total cost of serving the clients from centers is minimum.

Several versions of the problem exist in literature with different properties, the most common being Uncapacitated k Median Problem (UkM) and Capacitated k Median Problem (CkM). In the former case, each facility has infinite capacity (*i.e.* there is no limit on the amount of demand it can serve) in comparison to finite capacity in the latter case. In CkM, capacities may be soft or hard. In soft capacitated version, multiple copies of a facility can be opened at a location whereas in case of hard capacities, each facility is either opened at some location or not. Also, the capacities may be uniform or non-uniform. In the former case, all facilities have the same capacity in contrast to the latter one where-in different facilities have different capacities. Another variation of CkM is with respect to assignments of clients to facilities: in un-splittable assignments, the entire demand of a client has to be served by only one facility, in comparison to splittable assignments in which the demand of a client can be split among multiple facilities.

Several techniques have been used to obtain results for the problem. One of the most widely used technique to approximate the problem is LP Rounding ([4, 5, 7, 8, 9, 11, 12, 13, 14]). Charikar *et al.* [7] gave a $20/3$ factor approximation algorithm for UkM, which was further improved to 3.25 factor by Charikar and Li in [8]. Li and Svensson [14] further improved the ratio to $1+\sqrt{3}+\epsilon$. Their algorithm has a running time of O($n^{(1/\epsilon^2)}$).

Obtaining a constant approximation factor for CkM problem without violating capacity constraint and cardinality constraint is challenging as natural LP of the problem is known to have an unbounded integrality gap. Approximation results violate either capacity constraint or cardinality constraint, or both.

Cardinality violation: Li [12] gave a novel linear program called *rectangle LP* and presented an improved approximation algorithm ($exp(O(1/\epsilon^2))$) using at most $(1+\epsilon)k$ facilities for hard uniform CkM problem. The running time of the algorithm is $n^{O(1)}$, where the constant in the exponent does not depend on ϵ. He then extended this result to non-uniform soft capacitated variant of the problem in [13] and gave an $(O(1/\epsilon^2\ log(1/\epsilon)))$ approximation factor bounding softness by a factor of 2. The algorithm has a running time of $n^{O(1/\epsilon)}$.

Capacity violation: Charikar *et al.* [7] gave a 16 factor approximation algorithm for hard uniform CkM violating capacities by a factor of 3 in case of splittable demands and 4 in case of un-splittable demands. In 2015, Byrka *et al.* [4] gave an $O(1/\epsilon)$ approximation algorithm violating capacities by a factor of $(3+\epsilon)$ for hard non-uniform CkM. Demirci *et al.* [9] improved the approximation ratio to $O(1/\epsilon^5)$ with capacity violation of $(1+\epsilon)$ for the same

version of the problem. The running time of their algorithm is $n^{O(1/\epsilon)}$. Recently, Byrka *et al.* [5] gave an $O(1/\epsilon^2)$ approximation violating capacities by a factor of $(1+\epsilon)$ for hard uniform CkM. The algorithm uses randomized rounding to round a fractional solution to the configuration LP.

Aardal *et al.* [1] exploited the structure of an extreme point solution to give a $(7+\epsilon)$ factor algorithm for hard non-uniform Capacitated k- Facility Location Problem (CkFLP) violating cardinality constraint by a factor of 2. As a special case of CkFLP, their result applies on hard non-uniform CkM with all facility costs being zero. In the same manner, the CkFLP result $(1/\epsilon^2)$ of Byrka *et al.* [4] is applicable on hard uniform CkM. The result violates capacities by a factor of $2+\epsilon$.

The other commonly used technique for the problem is local search [2, 6, 10]. Charikar and Guha [6] gave 4 factor algorithm without violating cardinality constraint for the un-capacitated variant of the problem. Korupolu *et al.* [10] gave O$(1+\epsilon)$ factor approximation algorithm for UkM using at most $3+5/\epsilon$ facilities. Arya *et al.* [2] gave an improvised result of $3+2/p$ factor algorithm for UkM by using p-swaps.

We present a $(5+\epsilon)$ factor algorithm for hard uniform CkM violating the cardinality by a factor of 3 using Local Search. Algorithms based on local search are well known to be simpler as compared to the LP-based algorithms. The only result known for the problem using local search is due to Korupolu *et al.* [10]. They give an algorithm with a trade-off between approximation factor and cardinality loss. They give $(1+\alpha)$ approximation factor with $(5+5/\alpha)$ factor loss in cardinality. To achieve 5 factor in approximation, cardinality violation is more than 6. Though the approximation factor can be made arbitrarily small, cardinality loss is at least 5. Note that small approximation factor is obtained at a big loss in cardinality. For example, for α anything less than 1, cardinality violation is more than 10. Though we somewhat loose on the approximation factor, we surely improve upon the cardinality violation. Thus, there is a trade-off between cardinality violation and approximation factor amongst their result and ours. In particular, we present the following result:

Theorem 1. *There is a polynomial time algorithm that approximates hard uniform capacitated k median problem within 5 factor violating the cardinality by a factor of 3.*

High Level Idea: We extend the idea of 'mapping' of Arya *et al.* [2] to the capacitated version of the problem. However, for the capacitated case, mapping needs to be done a little intelligently. Mapping to an almost fully utilized facility may not be able to accommodate all the clients mapped to it and vice-versa. That is, a partially utilized facility may not be able to accommodate the load of an almost fully utilized facility. Thus, mapping is done only between the partially utilized facilities. To ensure that there are sufficient number of partially utilized facilities, we need to assume that we have sufficient number $(3k)$ of opened centers.

2 Notation and preliminaries

2.1 Capacitated k-median problem

In Capacitated k-Median Problem, we are given a set of $\mathcal{F}$ of facilities, a set $\mathcal{C}$ of clients and a real valued distance function c on $\mathcal{F} \cup \mathcal{C}$ in metric space. Each client $j \in \mathcal{C}$ has a non-negative demand d_j and each facility $i \in \mathcal{F}$ has a capacity u_i indicating the amount of demand it can serve. The cost of serving one unit of demand of a client $j \in \mathcal{C}$ from facility $i \in \mathcal{F}$ is denoted as $c(i,\ j)$. The goal is to select a subset $\mathcal{S} \subseteq \mathcal{F}$ of at most k facilities and assign clients to them without violating the capacities such that the total cost of serving all the clients by the opened facilities is minimum.

We consider the hard uniform capacitated k-median version of the problem *i.e.* $u_i = U\ \forall i \in \mathcal{F}$ and at most one instance of a facility can be opened at its location. We assume unit demand at each client *i.e.* $d_j = 1\ \forall j \in \mathcal{C}$.

2.2 Local search paradigm

Given a Problem P, let S be any arbitrary feasible solution to it. A new solution S′ is called a neighborhood solution of S if it can be obtained by performing local search operations such as adding one or more facilities $\mathtt{s} \notin \mathtt{S}$ to S or deleting one or more facilities s from S or swapping one or more facilities of S with facilities not in S. We now formally describe the steps of the algorithm.

The paradigm:

1. Compute an arbitrary feasible solution S to P.
2. While S′ is a neighborhood solution of S such that $cost(\mathtt{S}') < cost(\mathtt{S})$ do, set $\mathtt{S} = \mathtt{S}'$.

The solution S so obtained is called a locally optimal solution. Note that $cost(\mathtt{S}') \geq cost(\mathtt{S})$ for every neighborhood solution S′, for otherwise S would not have been locally optimal. More formally, a solution S is said to be locally optimal if no further operation results in improvement in cost.

3 $(5+\epsilon, 3)$ algorithm

For the k-median problems, we define an (a, b) - approximation algorithm as a polynomial-time algorithm that computes a solution using at most bk number of facilities with cost at most a times the cost of an optimal solution using at most k facilities.

We select an arbitrary set of facilities $\mathcal{S} \subseteq \mathcal{F}$ such that $|\mathcal{S}| = 3k$. This set acts as our initial feasible solution. Note that, defining a subset of opened facilities completely specifies a solution. We can obtain the assignments by solving an appropriately defined instance of transportation problem.

The only operation permitted by our algorithm is $swap(s, o)$, defined as follows: $\mathcal{S} = \mathcal{S} - \{s\} + \{o\}$, $o \in \mathcal{F} \setminus \mathcal{S}$, $s \in \mathcal{S}$. Reassign all the clients served by o in optimal to o in our new solution.

We run the local search algorithm on $\mathcal{S}$. Since $\mathcal{S}$ is now locally optimal, for all neighborhood solutions $\mathcal{S}'$ of $\mathcal{S}$, we have, $cost(\mathcal{S}') \geq cost(S)$.

3.1 Analysis

Let $\mathcal{O}$ denote the optimal solution to the problem. We now show that the local optimal solution $\mathcal{S}$ is within 5 factor of the optimal solution *i.e.* $cost(\mathcal{S}) \leq 5cost(\mathcal{O})$.

For a client j, let $\pi_{\mathcal{S}}(j)$ and $\pi_{\mathcal{O}}(j)$ denote the facilities serving j in $\mathcal{S}$ and $\mathcal{O}$ respectively. Also, let S_j and O_j denote the service costs paid by j in $\mathcal{S}$ and $\mathcal{O}$ respectively.

Let $s \in \mathcal{S}$ and $o \in \mathcal{O}$. Consider Figure 1. Let $\mathcal{B}_{\mathcal{S}}(s)$ denote the ball of s, that is, the set of clients served by s in $\mathcal{S}$. Similarly, let $\mathcal{B}_{\mathcal{O}}(o)$ denote the ball of $o \in \mathcal{O}$. Also, let $\mathcal{B}(s, o)$ be the set of clients served by $s \in \mathcal{S}$ and $o \in \mathcal{O}$.

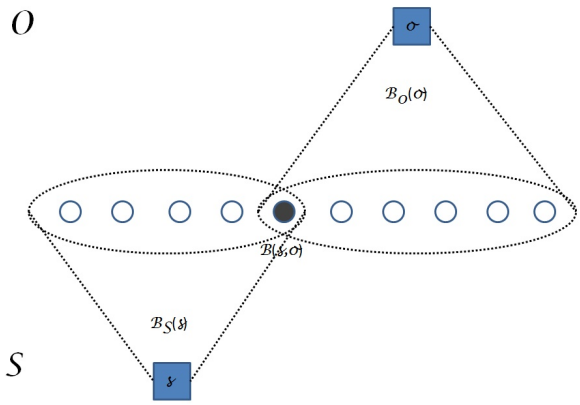


Figure 1: Balls of facilities

To deal with capacities, we classify the facilities in $\mathcal{S}$ based on the number of clients served by them. A facility $s \in \mathcal{S}$ is said to be *heavy* if it serves more than $U/2$ clients in $\mathcal{S}$, else it is said to be *light*. Note that the number of heavy facilities can be at most $2k$. Let $\mathcal{S}_{\mathcal{L}}$ denote the set of light facilities in $\mathcal{S}$. Since $|S| = 3k$, $|\mathcal{S}_{\mathcal{L}}| \geq k$.

Let $\mathcal{B}^L_{\mathcal{O}}(o)$ be the set of clients served by o in optimal and by light facilities in $\mathcal{S}$ and $\mathcal{M}_o = |\mathcal{B}^L_{\mathcal{O}}(o)|$. We say that a facility $s \in \mathcal{S}_{\mathcal{L}}$ *dominates* o, if it serves more than half the clients served by light facilities in $\mathcal{S}$ and by $o \in \mathcal{O}$, *i.e.* $\mathcal{B}(s, o) > \mathcal{M}_o/2$. A facility belonging to $\mathcal{S}_{\mathcal{L}}$ is called *bad* if it dominates more than one facilities in $\mathcal{O}$, it is called *good* if it dominates exactly one facility in $\mathcal{O}$, else it is called *nice*

We now devise a $1-1$ and onto mapping $\tau : \mathcal{B}^L_{\mathcal{O}}(o) \to \mathcal{B}^L_{\mathcal{O}}(o)$. Order the clients in $\mathcal{B}^L_{\mathcal{O}}(o)$ as $j_0, j_1, ..., j_{\mathcal{M}_o - 1}$ such that for every $s \in S$ with a nonempty $\mathcal{B}(s, o)$, the clients in $\mathcal{B}(s, o)$ are consecutive; that is, there exists r, s, $0 \leq r \leq s \leq \mathcal{M}_o - 1$, such that $\mathcal{B}(s, o) = \{j_r, ..., j_s\}$. Define $\tau(j_p) = (j_q)$, where $q = (p + \lfloor \mathcal{M}_o/2 \rfloor)$ *modulo* $\mathcal{M}_o$. Consider Figure 2a which shows the set $\mathcal{B}_{\mathcal{O}}(o)$. The corresponding mapping is shown in Figure 2b.

The following claim holds on mapping:

Claim 1. *If $s \in \mathcal{S}_{\mathcal{L}}$ does not dominate o, then $\tau(\mathcal{B}(s, o)) \cap \mathcal{B}(s, o) = \phi$.*

Proof. For contradiction, assume that both j_p, $\tau(j_p) = j_q \in \mathcal{B}(s, o)$ for some s, where $|\mathcal{B}(s, o)| \leq \mathcal{M}_o/2$. If $q = p + \lfloor \mathcal{M}_o/2 \rfloor$, then $|\mathcal{B}(s, o)| \geq q - p + 1 = \lfloor \mathcal{M}_o/2 \rfloor + 1 > \mathcal{M}_o/2$. If $q = p + \lfloor \mathcal{M}_o/2 \rfloor - \mathcal{M}_o$, then $|\mathcal{B}(s, o)| \geq p - q + 1 = \mathcal{M}_o - \lfloor \mathcal{M}_o/2 \rfloor + 1 > \mathcal{M}_o/2$. In either case, we have a contradiction, and hence mapping τ satisfies the claim. □

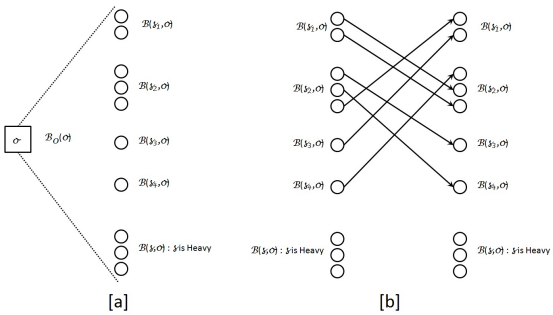


Figure 2: Mapping

The notion of dominate can be used to construct a bipartite graph $H = (\mathcal{S}, \mathcal{O}, E)$. For each facility in $\mathcal{S}_{\mathcal{L}}$, we have a vertex on the $\mathcal{S}$-side and for each facility in $\mathcal{O}$, we have a vertex on the $\mathcal{O}$-side. We add an edge between $s \in \mathcal{S}_{\mathcal{L}}$ and $o \in \mathcal{O}$ if s dominates o. Note that the degree of each vertex on $\mathcal{O}$-side is at most one while the vertices on the $\mathcal{S}$-side can have degree up to k.

We now consider all k swaps, one for each facility in $\mathcal{O}$. If $s \in \mathcal{S}_{\mathcal{L}}$ is good, then we consider the $swap(s, o)$, where o is the facility in $\mathcal{O}$ dominated by s. Let λ be the number of facilities in $\mathcal{O}$ that did not participate in the above swaps. Then the total number of bad and nice facilities in $\mathcal{S}_{\mathcal{L}}$ is at least λ and at least $\lambda/2$ of them must be nice. The remaining λ facilities in $\mathcal{O}$ get swapped with the nice facilities in $\mathcal{S}_{\mathcal{L}}$ such that each nice facility is considered in at most two swaps. The bad facilities are not considered for swapping. The swaps considered above satisfy the following properties:

1. Each $o \in \mathcal{O}$ is considered in exactly one swap.
2. Facilities in $\mathcal{S} \setminus \mathcal{S}_{\mathcal{L}}$ are not considered in any swap operation.
3. Bad facilities in $\mathcal{S}_{\mathcal{L}}$ are not considered in any swap operation.
4. Each nice facility $s \in \mathcal{S}_{\mathcal{L}}$ is considered in at most two swap operations.
5. If $swap(s, o)$ is considered then s does not dominate any facility $o' \neq o : o' \in \mathcal{O}$.

Lemma 1. *Let $cost(\mathcal{S})$ denote the cost of the local optimal solution $\mathcal{S}$ and, $cost(\mathcal{O})$ denote the cost of the global optimal solution $\mathcal{O}$. Then, $cost(\mathcal{S}) \leq 5cost(\mathcal{O})$.*

Proof. Consider $swap(s, o)$. Let $j \in \mathcal{B}_{\mathcal{S}}(s)$. We first reassign the clients in $\mathcal{B}_{\mathcal{S}}(s)$.

1. If $j \in \mathcal{B}_{\mathcal{O}}(o)$, assign j to o.

2. If $j \notin \mathcal{B}_{\mathcal{O}}(o)$, assign j to $s' \in \mathcal{S}_{\mathcal{L}}$ such that $\tau(j) = j'$ and $j' \in \mathcal{B}_{\mathcal{S}}(s')$.

In case 1, the change in cost is given by $(O_j - S_j)$. In case 2, the change in cost is $(c(j,\ s') - S_j)$. Let $j \in \mathcal{B}_{\mathcal{O}}(o')$. From triangle inequality, we get $c(j,\ s') \leq c(j,\ o') + c(o',\ \tau(j)) + c(\tau(j),\ s') = O_j + O_{\tau(j)} + S_{\tau(j)}$.
As $\mathcal{S}$ is a locally optimal solution, we have

$$\sum_{j \in \mathcal{B}_{\mathcal{S}}(s) \cap \mathcal{B}_{\mathcal{O}}(o)} (O_j - S_j) + \sum_{j \in \mathcal{B}_{\mathcal{S}}(s) \setminus \mathcal{B}_{\mathcal{O}}(o)} (O_j + O_{\tau(j)} + S_{\tau(j)} - S_j) > 0 \quad (1)$$

Each facility $o \in \mathcal{O}$ is considered in exactly one swap operation. Thus the first term of inequality when added over all k swaps gives exactly $cost(\mathcal{O}) - cost(\mathcal{S})$. Each $s \in \mathcal{S}$ is considered in at most two swaps. The second term of inequality when added over all k swaps is no greater than $2(O_j + O_{\tau(j)} + S_{\tau(j)} - S_j)$. As τ is a $1-1$ and onto mapping, $\sum_{j \in \mathcal{C}} O_j = \sum_{j \in \mathcal{C}} O_{\tau(j)}$ and $\sum_{j \in \mathcal{C}} (S_{\tau(j)} - S_j) = 0$. Thus, $2(O_j + O_{\tau(j)} + S_{\tau(j)} - S_j) = 4cost(\mathcal{O})$. Combining the two terms, we get $cost(\mathcal{O}) - cost(\mathcal{S}) + 4cost(\mathcal{O}) \geq 0$. Thus, $cost(\mathcal{S}) \leq 5cost(\mathcal{O})$. □

In the algorithm presented so far, we move to a new solution if it gives some improvement in the cost, however small that improvement may be. This may lead to an algorithm taking lot of time. To ensure that the algorithm terminates in polynomial time, a local search step is performed only when the cost of the current solution $\mathcal{S}$ is reduced by at least $\frac{cost(\mathcal{S})}{p(n,\epsilon)}$, where n is the size of the problem instance and $p(n, \epsilon)$ is an appropriate polynomial in n and $1/\epsilon$ for a fixed $\epsilon > 0$. This modification in the algorithm incurs a cost of additive ϵ in the approximation factor.

It is easy to see that if we have $3.5k$ facilities then the total number of bad and nice facilities in $\mathcal{S}_{\mathcal{L}}$ is at least $\lambda + k/2$ and at least $(\lambda + k)/2 \geq \lambda$ of them must be nice. The remaining λ facilities in $\mathcal{O}$ get swapped with the nice facilities in $\mathcal{S}_{\mathcal{L}}$ such that each nice facility is considered in at most one swap. This saves us factor 2 coming from the second term of equation (1). Thus, we get $(3 + \epsilon, 3.5)$ algorithm. Also, using p-swaps of Arya *et al*. [2], we can get $(3 + 2/p, 3)$ algorithm.

4 Conclusion and future work

We gave a $(5 + \epsilon)$ factor approximation algorithm for hard uniform capacitated k median problem using local search technique, violating cardinality by a factor of 3. It improves upon the existing results known for the problem using local search, with respect to cardinality violation. It would be interesting to obtain a constant factor algorithm reducing the cardinality violation to $(1 + \epsilon)$. Though such a result is known using LP-techniques, it would be interesting to obtain similar result using local search. Another direction to extend the work would be to consider the non-uniform capacitated version of the problem using local search.

References

[1] Karen Aardal, Pieter L. van den Berg, Dion Gijswijt, and Shanfei Li. Approximation algorithms for hard capacitated k-facility location problems. *European Journal of Operational Research*, 242(2):358–368, 2015. https://doi.org/10.1016/j.ejor.2014.10.011

[2] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristic for k-median and facility location problems. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, Heraklion, Crete, Greece*, pages 21–29, 2001. https://doi.org/10.1145/380752.380755

[3] Manisha Bansal. Approximation algorithms for facility location problems- https://drive.google.com/file/d/0bxmghjb2ede3nkvka2rzcfnethm/view. In *PhD Thesis, October, 2013*.

[4] Jaroslaw Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated *k*-median problems. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 722–736, 2015. https://doi.org/10.1137/1.9781611973730.49

[5] Jaroslaw Byrka, Bartosz Rybicki, and Sumedha Uniyal. An approximation algorithm for uniform capacitated k-median problem with (1 + ϵ) capacity violation. In *Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liège, Belgium, June 1-3, 2016, Proceedings*, pages 262–274, 2016. https://doi.org/10.1007/978-3-319-33461-5_22

[6] Moses Charikar and Sudipto Guha. Improved combinatorial algorithms for the facility location and k-median problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS), New York, NY, USA*, pages 378–388, 1999. https://doi.org/10.1137/s0097539701398594

[7] Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k-median problem (extended abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 1–10, 1999. https://doi.org/10.1145/301250.301257

[8] Moses Charikar and Shi Li. A dependent lp-rounding approach for the k-median problem. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, pages 194–205, 2012. https://doi.org/10.1007/978-3-642-31594-7_17

[9] H. Gökalp Demirci and Shi Li. Constant approximation for capacitated k-median with $(1 + \epsilon)$ - capacity violation. In *Proceedings of the 43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016.

[10] Madhukar R. Korupolu, C. Greg Plaxton, and Rajmohan Rajaraman. Analysis of a local search heuristic for facility location problems. *Journal of Algorithms*, 37(1):146–188, 2000. https://doi.org/10.1006/jagm.2000.1100

[11] Shanfei Li. An Improved Approximation Algorithm for the Hard Uniform Capacitated k-median Problem. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014), Germany*, pages 325–338. https://doi.org/10.1016/j.ejor.2014.10.011

[12] Shi Li. On uniform capacitated k-median beyond the natural LP relaxation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 696–707, 2015. https://doi.org/10.1137/1.9781611973730.47

[13] Shi Li. Approximating capacitated k-median with $(1 + \epsilon)k$ open facilities. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 786–796, 2016. https://doi.org/10.1137/1.9781611974331.ch56

[14] Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. In *Proceedings of Symposium on Theory of Computing Conference (STOC), Palo Alto, CA, USA*, pages 901–910, 2013. https://doi.org/10.1145/2488608.2488723

Efficient Trajectory Data Privacy Protection Scheme Based on Laplace's Differential Privacy

Ke Gu[†,‡,±], Lihao Yang[†], Yongzhi Liu[†], Bo Yin[†]
[†] School of Computer & Communication Engineering
Changsha University of Science & Technology, Changsha 410114, China
[‡] Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation
Changsha University of Science & Technology, Changsha 410114, China
[±] School of Information Science and Engineering, Central South University, Changsha 410083, China

Keywords: trajectory data, polygon, centroid, Laplace's differential privacy

Received: May 25, 2017

Now many applications of trajectory (location) data have facilitated people's daily life. However, publishing trajectory data may divulge individual sensitive information so as to influence people's normal life. On the other hand, if we cannot mine and share trajectory data information, trajectory data will lose its value to serve our society. Currently, because the records of trajectory data are discrete in database, some existing privacy protection schemes are difficult to protect trajectory data. In this paper, we propose a trajectory data privacy protection scheme based on Laplace's differential privacy mechanism. In the proposed scheme, the algorithm first selects the protected points from the user's trajectory data; secondly, the algorithm builds the polygons according to the protected points and the adjacent and high frequent accessed points selected from the accessed point database, then the algorithm calculates the polygon centroids; finally, the noises are added to the polygon centroids by the Laplace's differential privacy method, and the new polygon centroids are used to replace the protected points, and then the algorithm constructs and issues the new trajectory data. The experiments show that the running time of the proposed algorithms is fast, the privacy protection of the scheme is effective and the data usability of the scheme is higher.

Povzetek: Predlagana je metoda za učinkovito varovanje podatkov o poteh na osnovi Laplacove diferenčne privatnosti.

1 Introduction

1.1 Background

With the rapid development of computer and network, data mining and analysis plays an increasingly important role in our social life. The huge amounts of data (such as big data) can bring many application services to our society, such as trajectory (location) data, health and food data, traffic safety data, etc. Trajectory data is a kind of position information with large scale, fast changing and generally accepted characteristics, which mainly comes from vehicle networks, mobile devices, social networks and so on. Now many applications of trajectory data have facilitated people's daily life, thus trajectory data service is called as a kind of new mobile computing service. Currently, it is the key of developing trajectory data services that we must be able to learn and understand position information [1]. However, trajectory data is mainly collected and disseminated by mobile equipments, but many mobile devices and mobile communication technologies must integrate geographical data and individual information into trajectory data, such as individual information may contain individual privacy data, personal health status, social status and behavior habits, etc, thus mining and publishing trajectory data may divulge individual sensitive information so as to influence people's normal life [2,3,4].

Now it is the key of trajectory data privacy protection that how to protect sensitive trajectory data while providing trajectory information service on data mining. For example, if mined data is not processed and protected on fully open status, mined data may reveal user's privacy so as to affect user's normal life. Thus, it is double-edged sword that how to mine and use trajectory data. Namely we must find a compromising approach between service and protection. However, many existing privacy protection schemes cannot provide the balance of utility and protection. For example, the generalization method [5] cannot availably protect data, and the anonymous grouping method [6] is not efficient enough. Furthermore, because the records of trajectory data are discrete in database[1], some existing privacy protection schemes are difficult to protect trajectory data. Therefore, we focus on finding an efficient privacy protection scheme for trajectory data in this paper.

[1]In real world, trajectory data may not be discrete. In this paper, our focus is the combination of location data and accessed frequency, thus we consider that the records of trajectory data are discrete.

1.2 Our contributions

In this paper, we propose a trajectory data privacy protection scheme based on Laplace's differential privacy mechanism. In the proposed scheme, the algorithm first selects the protected points from the user's trajectory data; secondly, the algorithm builds the polygons according to the protected points and the adjacent and high frequent accessed points selected from the accessed point database, then the algorithm calculates the polygon centroids; finally, the noises are added to the polygon centroids by the Laplace' differential privacy method, and the new polygon centroids are used to replace the protected points, and then the algorithm constructs and issues the new trajectory data. The experiments show that the running time of the proposed algorithms is fast, the privacy protection of the scheme is effective and the data usability of the scheme is higher.

1.3 Outline

The rest of this paper is organized as follows. In Section 2, we discuss the related works about trajectory data privacy protection. In Section 3, we review the related definitions and theorems on which we employ. In Section 4, we propose an efficient trajectory data privacy protection scheme, which is based on the Laplace's differential privacy mechanism. In Section 5, we analyze and show the efficiency of the proposed scheme by the experiments. Finally, we draw our conclusions in Section 6.

2 Related work

Currently many privacy protection schemes are being widely used in many fields, such as secure communication, social network, data mining and so on. The works [5,6] first proposed the k-anonymity model to protect social network, whose anonymity protection methods mainly include generalization [7,8], compression, decomposition [9], replacement [10] and interference. Based on the works of [5,6], many other k-anonymous protection methods [11-21] were also proposed. However, the works [20,21,22] proved that some anonymous protection methods cannot protect sensitive data very well. Additionally, Cristofaro et al. [23] proposed a privacy-encrypted protection scheme. Although their scheme can ensure data security, data utility is decreased. Current location data privacy protection methods [1,24] are mainly classified to three categories: the heuristic privacy-measure methods, the probability-based privacy inference methods and the privacy information retrieval's methods. The heuristic privacy-measure methods [25,26,27,28] are mainly to provide the privacy protection measure for some no-high required users, such as k-anonymity [25], t-closing [26], m-invariability [27] and l-diversity [28]. Also, although the information retrieval's privacy protection methods can achieve perfect privacy protection, there are more or less privacy information in the released data, so these methods may result in that no data can be released, and these methods have high overhead. Additionally, the probability-based privacy inference methods can protect data and achieve better data utility under certain conditions, but the effectiveness of the methods depends on original data availability. Further, the three kinds of methods are based on a unified attack model [1], which depends on certain background knowledge to protect location data. However, with the increase of background knowledge got by the attackers, these methods could not always effectively protect location data. The works [5,6,11-19] showed the shortages of the relationship-privacy protection methods. Ting et al. [29] analyzed a variety of privacy threat models and tried to optimize the effectiveness of the data obtained while preventing different types of reasoning attacks. Bugra et al. [30] proposed the first effective location-privacy preserving mechanism (LPPM) that enables a designer to find the optimal LPPM for a LBS (location-based service) given user's service quality constraints against an adversary implementing the optimal inference algorithm. Such LPPM is the one that maximizes the expected distortion (error) that the optimal adversary incurs in reconstructing the actual location of a user, while fulfilling the user's service-quality requirement. Presently, it is the key of protecting location data to provide a privacy protection method not sensitively to background knowledge. Based on the requirement, differential privacy protection technology can exactly satisfy it. Differential privacy is a kind of strong privacy protection method, which is not sensitive to background knowledge. However, because location data has the characteristics of sparsity and farrago, many differential privacy protection methods are not enough efficient. He et al. [31] proposed a synthetic system based on GPS path, which can provide strong differential privacy protection mechanism. The proposed system gets different speed trajectory by using a hierarchical reference method to isolate the original trajectory, and then protects the speed trajectory. Chatzikokolakis et al. [32] proposed a predictive differentially-private mechanism for location privacy, which can offer substantial improvements over the independently applied noise. Their works showed that correlations in the trace can be in fact exploited in terms of a prediction function that tries to guess the new location based on the previously reported locations. Additionally, their works tested the quality of the predicted location using a private test; in case of success the prediction is reported otherwise the location is sanitized with new noise. Chatzikokolakis et al. [33] also showed a formal notion of privacy that protects the user's exact location–"geo-indistinguishability", and then proposed two mechanisms to protect the privacy of user when dealing with location-based services. Also they extended their mechanisms to the case of location traces, and provided a method to limit the degradation of the privacy guarantees due to the correlation between the points. Li et al. [34] proposed a compressive mechanism for differential privacy, which is based on compressed sensing theory. Their mechanism is to consider

every data as a single individual, so it undermines the relationship of data so as to be not suitable to protect location data. Jia et al. [1] proposed a differential privacy-based transaction data publishing scheme. Their method establishes the relationship of transaction data items by a query tree and adds noises to the query tree based on the compressive mechanism and the Laplace's mechanism. However, it is difficult to measure the effectiveness of their method on privacy protection. Zhang et al. [35] proposed an accurate method for mining top-k frequent data records under differential privacy. In their scheme, the exponential mechanism is used to sample top-k frequent data records, and then the Laplace's mechanism is utilized to generate noises to distort original data. Although the effectiveness of their method may accurately be measured on privacy protection, their method neglects the relationship of transaction data items.

3 Differential privacy

Differential privacy protection can achieve privacy protection target by making data distortion, where the common approach is to add noises into queried results. The purpose of differential privacy protection is to minimize privacy leakage and to maximize data utility [36,37]. Currently differential privacy protection has two main methods [38,39]—the Laplace's mechanism and the exponential mechanism.

DWork et al. [39] proposed a protection method for the sensitivity of private data, which is based on the Laplace's mechanism. Their method distorts the sensitive data by adding the Laplace's distribution noises to the original data. Their method may be described as follows: the algorithm M is the privacy protection algorithm based on the Laplace's mechanism, the set S is the noise output set of the algorithm M, and the input parameters are the data set D, the function Q, the function sensitivity ΔQ and the privacy parameter ε, where the set S approximately subjects to the Laplace's distribution ($\frac{\Delta Q}{\varepsilon}$) and the mean (zero), as shown in the formula (1):

$$\Pr\left[M(Q,D)=S\right] \propto \exp\left(\frac{\varepsilon}{\Delta Q}\cdot \mid S-Q(D)\mid_1\right) \qquad (1)$$

Also, in their method, the probability density function of added noise subjecting to the Laplace's distribution is as the formula (2):

$$\Pr(x,\lambda)=\frac{1}{2\cdot\lambda}\cdot e^{\frac{-|x|}{\lambda}} \qquad (2)$$

where $\lambda=\frac{\Delta Q}{\varepsilon}$, namely the added noise is independent from the data set, and is only related to the function sensitivity and the privacy parameter. The main idea of their method adds the noises subjecting to the Laplace's distribution into the output result so as to distort the sensitive data to achieve data protection target. For example, in their method, let $Q(D)$ be the querying function of top-k accessing count, then the output of the algorithm M can be represented by the following formula (3):

$$M(Q,D)= Q(D)+\left(Lap_1(\tfrac{\Delta Q}{\varepsilon}),\ Lap_2(\tfrac{\Delta Q}{\varepsilon}),...,\ Lap_k(\tfrac{\Delta Q}{\varepsilon})\right) \qquad (3)$$

where $Lap_i(\frac{\Delta Q}{\varepsilon})(1\le i\le k)$ is each round of the independent noise subjecting to the Laplace's distribution, and the noise is proportional to ΔQ and inversely proportional to ε.

Definition 3.1 $\varepsilon-$**Differential Privacy**: Given two adjacent data sets D and D' where at most a data record is different between D and D' ($|D\neq D'|=1$), for any algorithm M, whose range is $Range(M)$, if the result S outputted by the algorithm M satisfies the following formula (4) on the two adjacent data sets D and D' ($S\in Range(M)$), then the algorithm M satisfies $\varepsilon-$differential privacy.

$$\Pr\left[M(D)\in S\right]\le e^{\varepsilon}\cdot\Pr\left[M(D')\in S\right] \qquad (4)$$

Pr represents the randomicity of the algorithm M on D and D', namely denotes the risk probability of privacy disclosure. ε represents the privacy protection level, where if ε is bigger, then privacy protection degree is lower; on the contrary, if ε is smaller, then privacy protection degree is higher.

Definition 3.2 Data Sensitivity[2]: Data sensitivity is divided to global sensitivity and local sensitivity, we set Q as query function, then the global sensitivity of the function Q is defined as follows:

$$\Delta Q=\max_{D,D'}\{\mid Q(D)-Q(D')\mid_1\} \qquad (5)$$

where D and D' represent the adjacent data sets, $Q(D)$ represents the output of the function Q on the data set D, ΔQ is the sensitivity and represents the maximum of the outputs' difference.

Additionally, because the ε-differential privacy protection scheme may be used many times in the different stages of processing data, the ε-differential privacy protection scheme also needs to satisfy the following theorems:

Theorem 3.1 for the same data set, the whole privacy protection process is divided to the different privacy protection algorithms $(M_1,M_2,...,M_n)$, whose privacy protection levels are ε_1, ε_2,...,ε_n, so the privacy protection level $\sum_{i=1}^{n}\varepsilon_i$ of the whole process needs to satisfy differential privacy protection.

Theorem 3.2 for the disjoint data set, the whole privacy protection process is divided to the different privacy protection algorithms $(M_1,M_2,...,M_n)$, whose privacy protection levels are ε_1, ε_2,...,ε_n, so the privacy protection level $\max\{\varepsilon_i\}$ of the whole process needs to satisfy differential privacy protection.

[2]Differential privacy protection is to add noises to protect data, if data sensitivity is small, then it can effectively protect data while a small quantity of noises are added into original data; on the contrary, if data sensitivity is big, then a lot of noises need to be added into original data.

4 Trajectory data privacy protection scheme

In the section, we propose a trajectory data privacy protection scheme, which employs the Laplace's differential privacy method to protect the user's trajectory data. In the proposed scheme, the algorithm first selects the protected points from the user's trajectory data; secondly, the algorithm builds the polygons according to the protected points and the adjacent and high frequent accessed points selected from the accessed point database, then the algorithm calculates the polygon centroids; finally, the noises are added to the polygon centroids by the Laplace's differential privacy method, and the polygon centroids are used to replace the protected points, and then the algorithm constructs and issues the new trajectory data. The procedure of the proposed scheme is described as follows:

(1) Input the trajectory data I, the related and historic point data set D[3], the radius r and the differential privacy protection parameters ε and min_count[4];

(2) Select the protected point set A from the trajectory data I, then select the point data $f \in A$ and its corresponding adjacent points from D, where the adjacent points belong to the range of a circle that f is the center of the circle and r is the corresponding radius, and the frequent accessed counts of the adjacent points are no less than min_count, finally form the point set B;

(3) Traverse the set B, and build the corresponding polygons according to the points f and its corresponding adjacent points from B, where only one point in every polygon belongs to the trajectory data I, and then calculate the corresponding polygon centroids, and form the polygon centroid set J, where $j_i(x, y) \in J$ is the polygon centroid (see Section 4.2 for more details);

(4) Use the Laplace's mechanism to add the noises $Lap(\frac{k \cdot \Delta Q}{\varepsilon})$ into the set J, where the noises are added into the polygon centroids, and then generate the set G (see Section 4.3 for more details);

(5) Use the modified polygon centroids from G to replace the correspondingly protected points $f \in A$, and then issue the new trajectory data I'.

4.1 Processing trajectory data

The section describes how to select the related data from the trajectory data I and the related and historic point data set D. The proposed algorithm selects the protected point data $f \in A$ and its adjacent points from D. Figure 1 shows the procedure of selecting the related data. In Figure 1,

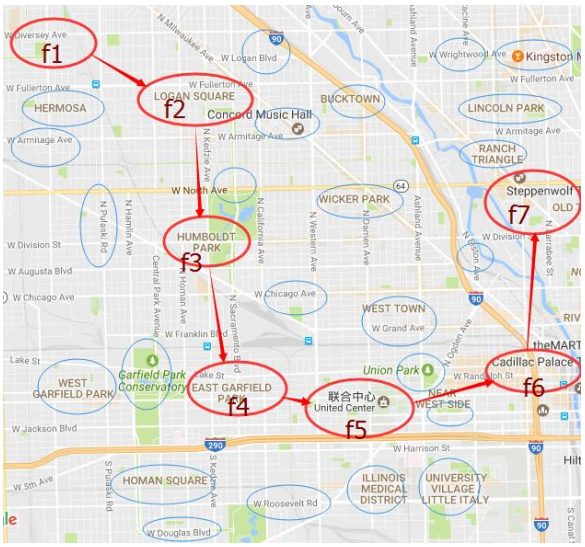


Figure 1 Processing Trajectory Data

a random trajectory of one user is shown, where the red circles and the red arrows are used to show the trajectory, and the green circles denote the accessed historic location points[5], which build the related and historic point data[6] set D. According to the Figure 1, the procedure of selecting the related data may be described as follows:

- The proposed algorithm inputs the trajectory data I of one user, the related and historic point data set D and the related privacy protection parameters r, ε and min_count;
- The algorithm selects the protected point set A from the trajectory data I;
- The proposed algorithm forms the point set B according to the point data $f_i \in A$ and its corresponding adjacent points from D, where the adjacent points belong to the range of a circle that f is the center of the circle and r is the corresponding radius, and the frequent accessed counts of the adjacent points are no less than min_count.

4.2 Building polygon model

The section describes how to build the polygon model to compute the polygon centroid. The proposed algorithm builds the polygons according to the protected points $f \in A$ and the corresponding adjacent points from D. Figure 2 shows the procedure of building polygon.

In Figure 2, the trajectory of one user is $f_1, f_2, ...f_5 \in I$, and the points $h_1, h_2, ...h_{13}$ with accessed counts come from D, where $f_2, f_4 \in A$ are the protected points.

[3]The related and historic point data include the historic location points accessed by people and the corresponding accessed counts. To the trajectory data, we may save the historic trajectory data and the related information (including accessed time and accessed count) to the database, and then the data may be classified to statistically form the set D.

[4]Our proposed scheme focuses on highly frequent accessed location data so as to distort attacker's target. So, the setting of min_count is to improve the efficiency of the proposed scheme.

[5]The adjacent point data may be related to other users.

[6]The historic duration is within one month.

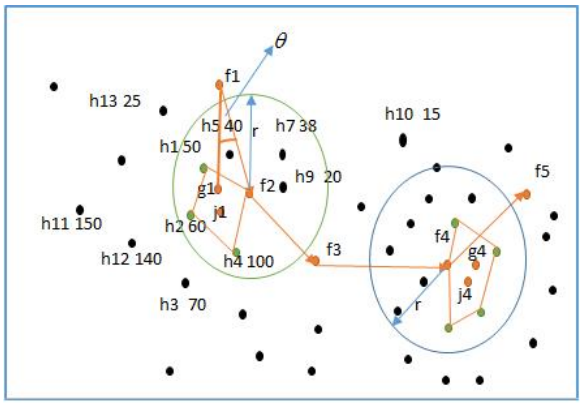


Figure 2 Building Polygon Model

In the green circle that f_2 is the center of the circle and r is the corresponding radius, the points h_1, h_2 and h_4 ($\in D$ and their accessed counts ≥ 50) and the point f_2 are used to form a polygon. Then the proposed algorithm computes the polygon centroid j_1 (noises are added to j_1 to generate a new point g_1). Similarly, the algorithm may traverse the set B to build the polygons. We need to remark that the points h_1, h_2 and h_4 is nearby the point f_2, thus the points may be used to build the polygon so as to maintain the usability of the modified trajectory, and that we set min_count is 50, thus some points whose accessed counts are less than 50 are not used to build the polygon in the green circle, such may distort the attacker's target and improve the efficiency of the proposed scheme. The procedure of building polygon model may be described as follows:

- The algorithm traverses the set B, and then selects the relevant and max-sized points to build the polygons according to the distance. For example, to a potential polygon, the algorithm selects N points as vertices from B whose coordinates are $P(x_i, y_i)$ with $i = 1, 2, 3.....N$, where one of the N points is in the original trajectory, and the other points are nearby the point;

- The algorithm computes the polygon centroids according to the vertices of the formed polygons. The formulas is described as follows:

$$j_i.x = \frac{\sum_{k=1}^{|P_i|} P_i.x_k}{n},\ j_i.y = \frac{\sum_{k=1}^{|P_i|} P_i.y_k}{n}.$$

  where $P_i(x_k, y_k)$ is the coordinate of the k_th vertices of the i_th polygon, $|P_i|$ is the vertices number of the i_th polygon, and $j_i(x, y)$ is the coordinate of the i_th polygon centroid.

- The polygon centroids are formed to the set J, where $j_i(x, y) \in J$.

4.3 Adding noises based on the Laplace's mechanism

In the section, we show how to use the Laplace's mechanism to add the noises $Lap(\frac{k \cdot \Delta Q}{\varepsilon})$[7] into the set J. The main steps of the algorithm are described as follows:

- Input the privacy protection level ε and the polygon centroid set J, and then generate the noise $Lap(\frac{k \cdot \Delta Q}{\varepsilon})$ satisfying the probability $\Pr(j(x, y), \lambda)$, where

$$\Pr(j(x, y), \lambda) = \frac{1}{2 \cdot \lambda} \cdot e^{\frac{-|j(x,y)|}{\lambda}}.$$

  In the above formula, the variant $j(x, y)$ denotes the corresponding coordinate of the polygon centroid and $\lambda = \frac{k \cdot \Delta Q}{\varepsilon}$.

- Add the noises $Lap(\frac{k \cdot \Delta Q}{\varepsilon})$ into the set J so as to disturb the polygon centroids[8]:

$$j_i.x = j_i.x \pm Lap(\frac{k \cdot \Delta Q}{\varepsilon}),$$
$$j_i.y = j_i.y \pm Lap(\frac{k \cdot \Delta Q}{\varepsilon}),$$

  where $j_i \in J$, $j_i(x, y)$ denotes the coordinate of the i_th polygon centroid, and $Lap(\frac{k \cdot \Delta Q}{\varepsilon})$ is each round of the independent noise subjecting to the probability $\Pr(j(x, y), \lambda)$. Finally, the algorithm generates the set G.

- Use the modified polygon centroids from G to replace the correspondingly protected points $f \in A$, and then issue the new trajectory data I'. For example, as the Figure 2 shown, the noise is added to j_1 to generate a new point g_1, and then g_1 is used to replace the point f_2, thus the original trajectory $f_1 \Rightarrow f_2 \Rightarrow f_3$ changes to $f_1 \Rightarrow g_1 \Rightarrow f_3$.

5 Experiment and efficiency analysis of the proposed scheme

In the section, our experiments are mainly from two aspects to evaluate the efficiency of the proposed scheme: the first one is the running time of the proposed algorithms, namely the time of extracting the available data; the second one is the effectiveness of the proposed algorithms, whose indexes include the trajectory deviation rate and the trajectory accurate rate. The test original data set comes from the simulation on the Baidu map[9], which is similar to the Gowalla

[7] ΔQ is the sensitivity of the query function Q, where we set $\Delta Q = \max\{\sqrt{(P_i.x_k - j_i.x)^2 + (P_i.y_k - j_i.y)^2}\}$ with $i = 1, 2, ...|N_P|$ and $k = 1, 2, ...|P_i|$, $|N_P|$ is the number of the polygons and $|P_i|$ is the number of the vertices of every polygon.

[8] If the formed polygon is on the left of the protected point from the trajectory data I, then the operation " $+$ " is used; otherwise, the formed polygon is on the right of the protected point from the trajectory data I, then the operation " $-$ " is used.

[9] Baidu is a network company in China. The baidu map is one of the network services provided by the company, which provides a lot of APIs for programmers to develop their applications on the map.

data set[10]. The test original data set contains user_id, accessed time, longitude and latitude and so on. The period of the test original data set is about one month. All proposed algorithms are coded by C++ and codeblocks[11]. The related parameters for the test are set as Table 1.

Table 1: Parameter Value

Parameter	Value (unit: 5 meter)
r	40,50,60,70,80,90,100,110
ε	1,2,3,4,5,6,7,8,9,10,11,12

5.1 Running time analysis

In the section, we test the running time of the proposed algorithms mainly through the time of extracting the available data, namely we test the effectiveness of computing all the polygon centroids from the available data. In the tests, when we set r=70 and ε=1,2,3,4,5,6,7,8,9,10,11,12 respectively, the time of extracting the available data is described as Table 2.

From the Table 2, we may know the time of extracting the available data is very fast, and the efficiency of computing all the polygon centroids from the available data is always increasing with the increasing of ε in a certain range.

5.2 Protection effectiveness analysis

In the section, we test the protection effectiveness of the proposed algorithms mainly through the trajectory deviation rate and the trajectory accurate rate, where the trajectory deviation rate is the angle θ formed by the modified polygon centroid and the original trajectory points, shown as Figure 3, and if the trajectory deviation rate is bigger in a certain range, then the protection effectiveness is higher; the trajectory accurate rate is used to test the protection effectiveness and usability of the noise-added data, and if the trajectory accurate rate is smaller in a certain range, then the usability is higher.

In the test, we compute the trajectory accurate rate through the following methods: 1) set the coordinate (a_i, b_i) of the polygon centroid; 2) compute the hypotenuse $c_i = \sqrt{a_i^2 + b_i^2}$; 3) compute the accurate rate $Z = \mid 1 - \frac{c_i'}{c_i} \mid$, where c_i is the original hypotenuse and c_i' is the noise-added hypotenuse. The trajectory deviation rate is bigger in a certain range, the protection effectiveness is higher; the trajectory accurate rate is smaller in a certain range, the usability is higher. So, when we set $\varepsilon = 5, 10, 15$ and $r = 40, 50, 60, 70, 80, 90, 100, 110$ respectively, Table 3,4,5 show the deviation rate and accurate rate of the trajectory data.

[10]Gowalla is a location-based social networking website where users share their locations by checking-in.

[11]The test environment is under Win10 OS, Intel i5 CPU 2.3Ghz and 8G RAM.

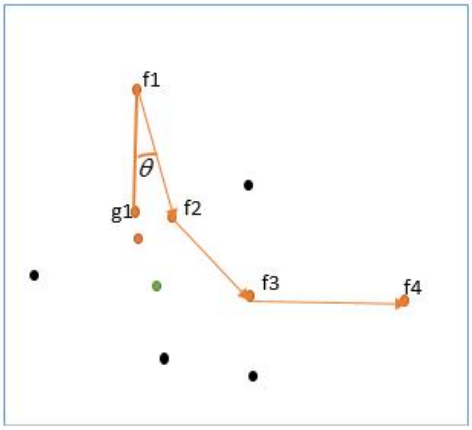


Figure 3 Trajectory Deviation Angle

From the Table 3, when $\varepsilon = 5$ and $r < 90$, we may know that the polygon centroid is not changed with the increasing of r , thus the deviation rate θ and the accurate rate Z are also not changed. Such shows that in the range of $r < 90$, the new points are not selected to build the new polygon, thus the polygon is not modified. when $r >= 90$, the new points are selected to build the new polygon, thus the polygon centroid is recomputed, thus the deviation rate θ and the accurate rate Z are changed. Such shows that the deviation rate θ could become big with the increasing of r, and the data usability becomes small. Also, from the Table 4 and the Table 5, when $\varepsilon = 10, 15$, we may get the similar results as that of the Table 3. Additionally, when we fixedly set $r = 70$ and $\varepsilon = 1, 2, 3, 4, ...15$ respectively, Table 6 shows the deviation rate and accurate rate of the trajectory data. From the Table 6, we may know that the deviation rate θ and the accurate rate Z are always increasing with the increasing of ε. That is because the constraint condition becomes small with the increasing of ε in the differential privacy mechanism. However, such also shows that the deviation rate θ becomes big so that the data usability becomes small.

6 Conclusions

Currently, because the records of trajectory data are discrete in database, some existing privacy protection schemes are difficult to protect trajectory data. In this paper, we propose a trajectory data privacy protection scheme based on Laplace's differential privacy mechanism. In the proposed scheme, the algorithm first selects the protected points from the user's trajectory data; secondly, the algorithm builds the polygons according to the protected points and the adjacent and high frequent accessed points selected from the accessed point database, then the algorithm calculates the polygon centroids; finally, the noises are added to the polygon centroids by the differential privacy method, and the new polygon centroids are used to replace the protected points, and then the algorithm constructs and issues the

Table 2: The Efficiency of Extracting Available Data

ε	1	2	3	4	5	6	7	8	9	10	11	12
Time (ms)	4	4	3	3	3	4	3	3	3	3	3	2

Table 3: Trajectory Deviation Rate And Accurate Rate ($\varepsilon = 5$)

r	c_i	c'_i	Z	θ
40	645.264	613.125	0.049807	23.2510
50	645.264	613.125	0.049807	23.2510
60	645.264	613.125	0.049807	23.2510
70	645.264	613.125	0.049807	23.2510
80	645.264	613.125	0.049807	23.2510
90	608.511	572.839	0.058621	24.7920
100	608.511	572.839	0.058621	24.7920
110	608.511	572.839	0.058621	24.7920

Table 4: Trajectory Deviation Rate And Accurate Rate ($\varepsilon = 10$)

r	c_i	c'_i	Z	θ
40	645.264	613.096	0.049852	23.2532
50	645.264	613.096	0.049852	23.2532
60	645.264	613.096	0.049852	23.2532
70	645.264	613.096	0.049852	23.2532
80	645.264	613.096	0.049852	23.2532
90	608.511	572.809	0.05867	24.7941
100	608.511	572.809	0.05867	24.7941
110	608.511	572.809	0.05867	24.7941

Table 5: Trajectory Deviation Rate And Accurate Rate ($\varepsilon = 15$)

r	c_i	c'_i	Z	θ
40	645.264	612.964	0.050057	23.2584
50	645.264	612.964	0.050057	23.2584
60	645.264	612.964	0.050057	23.2584
70	645.264	612.964	0.050057	23.2584
80	645.264	612.964	0.050057	23.2584
90	608.511	572.665	0.058908	24.7996
100	608.511	572.665	0.058908	24.7996
110	608.511	572.665	0.058908	24.7996

Table 6: Trajectory Deviation Rate And Accurate Rate ($r = 70$)

ε	c_i	c'_i	Z	θ
1	645.264	613.126	0.049806	23.25090
2	645.264	613.126	0.049806	23.25090
3	645.264	613.126	0.049806	23.25090
4	645.264	613.126	0.049806	23.25090
5	645.264	613.125	0.049807	23.2510
6	645.264	613.125	0.049807	23.2510
7	645.264	613.122	0.049812	23.2514
8	645.264	613.117	0.049819	23.2518
9	645.264	613.109	0.049833	23.2524
10	645.264	613.096	0.049852	23.2532
11	645.264	613.079	0.049879	23.2541
12	645.264	613.057	0.049913	23.2551
13	645.264	613.030	0.049954	23.2562
14	645.264	612.999	0.050003	23.2573
15	645.264	612.964	0.050057	23.2584

new trajectory data. The experiments show that the running time of the proposed algorithms is fast, the privacy protection of the scheme is effective and the data usability of the scheme is higher.

Acknowledgement

This study was funded by the National Natural Science Foundation of China (No.61402055,No.61504013), the Natural Science Foundation of Hunan Province (No.2016JJ3012).

References

[1] Ouyang Jia, Yin Jian, Liu Shaopeng, Liu Yuba. An Effective Differential Privacy Transaction Data Publication Strategy. Journal of Computer Research& development, 2014, 51(10):2195-2205. https://doi.org/10.7544/issn1000-1239.2014.20130824

[2] Loki. Available at: http://loki.com/.

[3] FireEagle. Available at: http://info.yahoo.com/privacy/us/yahoo/fireeagle/.

[4] Google latitude. Available at: http://www.google.com/latitude/apps/badge.

[5] Samarati P, Sweeney L. Generalizing data to provide anonymity when disclosing information. Proc. of the 7th ACM SIGACTSIGMOD-SIGART Symp.

on Principles of Database Systems, 1998, 188-202. https://doi.org/10.1145/275487.275508

[6] Samarati P. Protecting Respondents Identities in Microdata Release. IEEE Transactions on Knowledge and Data Engineering, 13(6):1010-1027. https://doi.org/10.1109/69.971193

[7] Fung BC, Wang K, Yu PS. Anonymizing classification data for privacy preservation. IEEE Trans on Knowledge and Data Engineering(TKDE), 2007,19(5):711-725. https://doi.org/10.1109/tkde.2007.1015

[8] Sweeney L. Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2012, 10(5):571-588. https://doi.org/10.1142/S021848850200165X

[9] Xiao X, Tao Y. Anatomy: Simple and effective privacy preservation. Proc of VLDB 2006. New York: ACM, 2006, 139-150.

[10] Zhang Q, Koudas N, Srivastava D, et al. Aggregate query answering on anonymized tables. Proc of the 23rd Int Conf on Data Engineering(ICDE). Piscataway, NJ: IEEE, 2007, 116-125. https://doi.org/10.1109/icde.2007.367857

[11] Wong RCW, Li J, Fu AWC, Wang K. (a, k)-Anonymity: An enhanced k-anonymity model for privacy-preserving data publishing. In: Proc.of the ACM 12th SIGKDD Int'l Conf on Knowledge Discovery and Data Mining, 2006, 754-759. https://doi.org/10.1145/1150402.1150499

[12] LeFevre K, DeWitt DJ, Ramakrishnan R. Mondrian multidimensional k-anonymity. Proc.of the 22nd Int'l Conf. on Data Engineering, 2006, 6(3): 25-35. https://doi.org/10.1109/ICDE.2006.101

[13] Machanavajjhala A, Gehrke J, Kifer D, Venkitasubramaniam M. L-Diversity: Privacy beyond k-anonymity. Proc.of the 22nd IEEE Int'l Conf. on Data Engineering, 2006. https://doi.org/10.1109/ICDE.2006.1

[14] Xiao X, Tao Y. Personalized privacy preservation. Proc.of the 2006 ACM SIGMOD Int'l Conf. on Management of Data, 2006, 229-240. https://doi.org/10.1145/1142473.1142500

[15] Xu J, Wang W, Pei J, Wang X, Shi B, Fu AWC. Utility-Based anonymization using local recoding. Proc.of the 12th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, 2006, 785-790. https://doi.org/10.1145/1150402.1150504

[16] Li N, Li T, Venkatasubramanian S. t-closeness: Privacy beyond k-anonymity and l-diversity. Proc.of the 23rd IEEE Int'l Conf. on Data Engineering, 2007, 106-115. https://doi.org/10.1109/icde.2007.367856

[17] Wong RCW, Fu AWC, Wang K, Pei J. Minimality attack in privacy preserving data publishing. Proc.of the 33nd Int'l Conf. on Very Large Databases, 2007, 543-554.

[18] Tao Y, Xiao X, Li J, Zhang D. On anti-corruption privacy preserving publication. Proc.of the 24th Int'l Conf. on Data Engineering, 2008, 725-734. https://doi.org/10.1109/ICDE.2008.4497481

[19] Backstrom L, Dwork C, Kleinberg J. Wherefore are thouR3579X?: Anonymized social networks, hiddern patterns and structural steganography. Proc.of the 16th Int'l Conf. on World Wide Web, 2007, 181-190. https://doi.org/10.1145/1242572.1242598

[20] Zheleva E, Getoor L. Preserving the privacy of sensitive relationships in graph data. Proc.of the 1st ACM SIGKDD Workshop on Privacy, Security, and Trust in KDD, 2007, 153-171. https://doi.org/10.1007/978-3-540-78478-4_9

[21] Korolova A, Motwani R, Nabar SU, Xu Y. Link privacy in social networks. Proc. of the 24th Int'l Conf. on Data Engineering, 2008, 1355-1357. https://doi.org/10.1109/icde.2008.4497554

[22] Cristofaro E, Soriente C, Tsudik G, et al. Hummingbird: Privacy at the time of twitter. IEEE Symposium on Security and Privacy -S&P, 2012, 285-299. https://doi.org/10.1109/sp.2012.26

[23] Beresford AR, Rice A, Skehin N, Sohan R. MockDroid: Trading privacy for application functionality on smartphones. Proc. of the 12th Workshop on Mobile Computing Systems and Applications, ACM Press, 2011, 49-54. https://doi.org/10.1145/2184489.2184500

[24] Huo Z, Meng XF. A survey of trajectory privacy-preserving techniques. Chinese Journal of Computers, 2011, 34(10):1820-1830. https://doi.org/10.3724/sp.j.1016.2011.01820

[25] Bamba B, Liu L, Pesti P, Wang T. Supporting anonymous location queries in mobile environments with privacy grid. Proc. of the 17th Int'l Conf. on World Wide Web. New York: ACM Press, 2008, 237-246. https://doi.org/10.1145/1367497.1367531

[26] Liu L. From data privacy to location privacy: Models and algorithms. Proc. of the 33rd Int'l Conf. on Very Large Data Bases. New York: ACM Press, 2007, 1429-1430.

[27] Liu F, Hua KA, Cai Y. Query l-diversity in location-based services. Proc. of the 10th Int'l Conf. on Mobile Data Management. Taipei, 2009, 436-442. https://doi.org/10.1109/mdm.2009.72

[28] Ting W, Ling L. From Data Privacy to Location Privacy. Machine Learning in Cyber Trust, Berlin: Springer, 14 March 2009, pp.217-246. https://doi.org/10.1007/978-0-387-88735-7_9

[29] Bugra G, Ling L. Protecting Location Privacy. IEEE Transactions on Mobile Computing, 2008, 7(1):1-18. https://doi.org/10.1109/TMC.2007.1062

[30] He X, Cormode G, Machanavajjhala A, Procopiuc CM, Srivastava D. Differentially Private Trajectory Synthesis Using Hierarchical Reference Systems. VLDB Journal, 2015, 8(11):1154-1165. https://doi.org/10.14778/2809974.2809978

[31] Chatzikokolakis K, Palamidessi C, Stronati M. A Predictive Differentially-Private Mechanism for Location Privacy. Proc. of the 14th International Symposium on Privacy Enhancing Technologies, Berlin: Springer, 2014, LNCS 8555, pp.21-41. https://doi.org/10.1007/978-3-319-08506-7_2

[32] Chatzikokolakis K, Palamidessi C, Stronati M. Geo-indistinguishability: A Principled Approach to Location Privacy. ICDCIT 2015, Berlin: Springer, 2015, LNCS 8956, pp.49-72. https://doi.org/10.1007/978-3-319-14977-6_4

[33] Li YD, Zhang Z, Winslett M, et al. Compressive mechanism: Utilizing sparse representation in differential privacy. Proc. of the 10th Annual ACM Workshop on Privacy in the Electronic Society. New York: ACM, 2011, pp.177-182. https://doi.org/10.1145/2046556.2046581

[34] Zhang XJ, Wang M, and Meng XF. An Accurate Method for Mining top-k Frequent Pattern Under Differential Privacy. Journal of Computer Research and Development, 2014, 51(1):104-114. https://doi.org/10.7544/issn1000-1239.2014.20130685

[35] Dwork C. The promise of differential privacy: A tutorial on algorithmic techniques. Proc. of the Foundations of Computer Science (FOCS). Piscataway, NJ: IEEE, 2011, pp.1-2. https://doi.org/10.1109/focs.2011.88

[36] Dwork C. A firm foundation for private data analysis. Communications of the ACM, 2011, 54(1):86-95. https://doi.org/10.1145/1866739.1866758

[37] Mcsherry F, Talwar K. Mechanism design via differential privacy. Proc. of the 48th Annual IEEE Symp. on Foundations of Computer Science (FOCS), Piscataway, NJ: IEEE, 2007, pp.94-103. https://doi.org/10.1109/focs.2007.4389483

[38] DWork C, McSherry F, Smith A. Calibrating noise to sensitivity in private data analysis. Proc. of the 3th Theory of Cryptography Conf (TCC06), Berlin: Springer, 2006, pp.265-284. https://doi.org/10.1007/11681878_14

https://doi.org/10.31449/inf.v42i3.1497 Informatica **42** (2018) 417–438 **417**

A Hybrid Particle Swarm Optimization and Differential Evolution Based Test Data Generation Algorithm for Data-Flow Coverage Using Neighbourhood Search Strategy

Sapna Varshney and Monica Mehrotra
Department of Computer Science, Jamia Millia Islamia, India
E-mail: sapna_varsh@yahoo.com, drmehrotra2000@gmail.com

Keywords: search based software testing, particle swarm optimization, differential evolution, data flow testing, dominance tree

Received: January 15, 2017

Meta-heuristic search techniques, mainly Genetic Algorithm (GA), have been widely applied for automated test data generation according to a structural test adequacy criterion. However, it remains a challenging task for more robust adequacy criterion such as data-flow coverage of a program. Now, focus is on the use of other highly-adaptive meta-heuristic search techniques such as Particle Swarm Optimization (PSO) and Differential Evolution (DE). In this paper, a hybrid (adaptive PSO and DE) algorithm is proposed to generate test data for data-flow dependencies of a program with a neighbourhood search strategy to improve the search capability of the hybrid algorithm. The fitness function is based on the concepts of dominance relations and branch distance. The measures considered are mean number of generations and mean percentage coverage. The performance of the hybrid algorithm is compared with that of DE, PSO, GA, and random search. Over several experiments on a set of benchmark programs, it is shown that the hybrid algorithm performed significantly better than DE, PSO, GA and random search in data-flow test data generation with respect to the measures collected.

Povzetek: Razvit je nov algoritem kot kombinacija hibridnega roja delcev in diferenčne evolucije z uporabo sosednje iskalne strategije.

1 Introduction

Software testing aims at assessing the quality and reliability of software product by detecting as many defects as possible. The cost of software testing increases exponentially with the size of input search space, thereby making manual testing a difficult and tedious task. There are software testing tools available with capture and playback features to automate the execution of test scripts. However, the test cases are manually selected by the human tester and may not be optimal. It is therefore desirable to generate optimal test data that reveals as many errors as possible according to a test adequacy criterion [1]. *Structural (white-box) testing* tests software for its structure and has the inherent capability to expose faults. The structural test adequacy criteria can be *statement coverage, branch coverage,* or *path coverage* that aim at executing every statement, branch or path respectively at least once. *Data-flow coverage*, an effective and robust test adequacy criterion, focuses on the definition and usage of variables in a program. Data-flow testing, therefore, could lead to more efficient and targeted test suites.

The attempts to reduce the cost of software testing by automating the process of software test data generation have been constrained by the ever increasing size and complexity of software. In the early period of automated test data generation, gradient descent and meta-heuristic search (MHS) algorithms such as Tabu Search, Hill Climbing and Simulated Annealing [2, 3, 4]. In the past two decades, evolutionary search-based algorithms such as Genetic Algorithm (GA) have been widely employed for test data generation as an effective alternative [5, 6, 7, 8, 9]. A search-based approach captures the test adequacy criteria as a fitness function that is used to guide the search. Due to an extensive application of search-based algorithms to test data generation problem, the approach has come to be known as Search Based Software Testing (SBST, coined by Harman and Jones). Recently, the focus is on the use of other highly adaptive search-based techniques such as Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) and Differential Evolution (DE). It has been observed that GA and ACO have slow convergence towards the optimal solution. PSO and DE are conceptually very simple and the knowledge of previous good solutions is retained by all the members of the current population by means of constructive cooperation among them. PSO and DE have been found to be robust in solving optimization problems; however, the performance depends on control parameters. PSO has been shown to be well suited for test data generation with better performance than GA [10, 11, 12, 13, 14]. Hybridization of search-based algorithms for test data generation has also been reported in literature. GA with a local search algorithm [15] and more recently, GA with

PSO has been applied for test data generation in some studies [16, 17, 18, 19, 20, 21].

In this study, we propose a hybrid global search algorithm by combining an adaptive PSO with DE mutation operator to automatically generate test data for data-flow dependencies of a program. In the proposed hybrid algorithm, a new term based on DE differential operator is included for velocity update in PSO for some additional exploration capability. The greedy selection scheme of DE is used wherein position of a particle is updated only if it yields a better fitness value. This results in movement of particles only to better locations in the input search space. A local neighborhood strategy is also included in the proposed hybrid algorithm to explore more promising candidate solutions and overcome the problem of boundary constraints. Design of the fitness function [22] is based on dominance concepts and branch distance that is used to guide the search for optimal test data for data-flow dependencies of a program. The performance of the proposed hybrid algorithm is compared with that of DE, PSO, GA and random search. It is demonstrated that the proposed hybrid algorithm outperformed DE, PSO, GA and random search in terms of mean percentage coverage achieved, and mean number of generations to produce the final test suite for data-flow coverage of a program.

The rest of the paper is organized as follows: Section 2 provides a brief description of automated software test data generation process and related work. Section 3 provides an overview of data-flow analysis. Sections 4 and 5 provide a brief description of PSO and DE algorithms. Section 6 describes the proposed hybrid algorithm. Section 7 gives the experimental results. Section 8 provides the discussion and the detailed statistical analysis of the experimental results. Section 9 deals with threats to validity and limitations of the proposed hybrid algorithm. Finally, section 10 gives the conclusion.

2 Related work

This section presents the methods to generate test data for software structural testing and the related literature. ***Symbolic execution,*** a static method, has been employed for test data generation [2]; however, the performance is constrained by programming constructs such as pointers, loop conditions with input variables, array subscripts and procedure calls [23]. Dynamic methods that have been employed for test data generation can be classified as *random*, *path-oriented* and *goal-oriented* techniques [9, 23]. A ***random test data generator*** arbitrarily selects test data from the input domain. Though easy to implement, it may fail to find optimal test data. ***Path-oriented test data generator*** [5] uses control flow information to identify a set of independent paths to generate test data. However, it does not work well with infeasible paths or paths that contain loops. A ***goal-oriented test data generator*** [9, 23, 24] generates test data for a selected goal such as a statement or a branch, irrespective of the path taken.

The meta-heuristic search techniques guided by a *fitness function* have been adopted to generate optimal test data mainly according to a structural test adequacy criterion. From the literature on structural test data generation, it can be inferred that branch coverage and path coverage are the most often used and well-understood measures [25]. For branch coverage, fitness values are calculated by finding approximation level and branch distance for a target branch from control flow graph [8, 26]. Data-flow coverage criterion has not been used much [27] due to difficulty in writing test cases that satisfy data-flow dependencies of a program. Wegener et al. [28] defined different types of fitness functions for structural testing; data-flow test criteria being classified as node-node-oriented methods. Recently only there has been more work on search based test data generation for data-flow coverage using GA as the algorithm of choice [6, 7, 22, 24, 29, 30]. Now, other highly adaptive search-based techniques such as PSO [14, 18] and ACO [31] are also being applied to generate test data for data-flow coverage due to simplicity and faster convergence. ACO [32] and Harmony Search [33] has also been applied to generate structural test data for branch coverage.

Vivanti et al. [30] have proposed a GA-based technique for data-flow coverage evaluated on open source Java applications. The results have indicated the scalability and applicability of data-flow criteria for test data generation.

In our previous work [22], an elitist GA-based approach is proposed to generate test data for data-flow dependencies of a program using dominance concepts and branch distance. The fitness function is derived from the work by Ghiduk et al. [6]; it is augmented with branch distance to produce a smoother landscape for guiding the search and also takes into account that a definition may be killed by another definition before the associated use is reached. The performance of the proposed approach is compared with random search and earlier studies on test data generation for data-flow dependencies of a program by Girgis [7], Ghiduk et al. [6] and Girgis et al. [21]. The proposed GA-based approach guided by the novel fitness function outperformed random search and the earlier studies [6, 7, 21] to generate test data for data-flow coverage of a program.

Windisch et al. [10] applied PSO to artificial and complex industrial test objects to generate test data for branch coverage. Their results showed efficiency and efficacy of PSO over GA for most code elements to be covered.

Agarwal et al. [11] applied binary PSO, Agarwal and Srivastava [12] applied discrete quantum PSO and Mao [13] applied standard PSO to generate test data for branch coverage test adequacy criterion.

Nayak and Mohapatra [14] proposed an algorithm to generate test cases using PSO for data flow coverage. This technique cannot rank test cases because the fitness function, as simply taken from Girgis [7], assigns the same fitness value to all the test cases that cover the same number of test requirements and a fitness value of 0 to all the test cases that do not cover any test requirement or

cover a partial aim. Here, the fitness function is unable to guide the search.

Application of hybrid algorithms have also been studied for test data generation problem. Zhang et al. [16] proposed a hybrid algorithm (GA and PSO) to generate test data for path coverage. GA and PSO operations are applied to two population sets. Triangle classification problem is taken as the case study and the hybrid algorithm is compared with GA and PSO. The hybrid algorithm is shown to be better than GA and PSO with respect to number of iterations. The average time taken is found to be more than PSO but less than GA. Their hybrid technique is complicated and may generate redundant test cases for automatic test data generation.

Li et al. [17] also proposed a hybrid algorithm (GA and PSO) to generate test data for path coverage. PSO equations to update particle's velocity and position distance are used instead of mutation operator of GA. The algorithm is applied only to the triangle benchmark problem.

Singla et al. [18] applied a hybrid algorithm (GA and PSO) to generate test data for data-flow coverage. The fitness function used is same as in [6]; it does not take into account the traversal of killing nodes as well as closeness of test data in case if only partial aim is covered. The strategy is tested only on some simple programs.

Kaur and Bhatt [19] proposed a hybrid algorithm (GA and PSO) to prioritize test data in regression testing. The algorithm has been tested on few simple programs.

Girgis et al. [21] proposed a hybrid Genetical Swarm Optimization (GSO) Technique to generate a set of test paths that cover the all-uses criterion for data-flow coverage. The authors have claimed that the set of paths generated by the proposed GSO can be passed to a test data generation tool to find program inputs that will execute them to complete the data flow paths testing of the program under test. The fitness function used is same as in [7]; it is not able to guide the search and results in loss of valuable information in case if only partial aim is covered.

Chawla et al. [20] proposed a hybrid PSO and GA algorithm for automatic generation of test suites with branch coverage as the test adequacy criterion. The experiments are performed with ten Java container classes. The algorithm is shown to perform better than GA, PSO and existing hybrid strategies based on GA and PSO.

Each optimization algorithm has its own advantages and disadvantages. Also, one optimization algorithm will not work well for all the optimization problems. DE, a meta-heuristic search-based algorithm, has been applied to several optimization problems [34, 35] to demonstrate its potential. Das et al. [36] has explored hybridization of PSO with DE applied to the design of digital filters. However, DE has not been applied for test data generation and optimization problem [25, 27, 37].

The proposed study will focus on the application of a hybrid adaptive PSO-DE algorithm to generate test data for data-flow dependencies of a program. The proposed hybrid global search algorithm combines the evolution scheme of both PSO and DE incorporating the best of both the algorithms in the context of test data generation. A new term based on DE differential operator is included for velocity update in PSO. The greedy selection scheme of DE is also used wherein position of a member is updated only if it yields a better fitness value. The hybridization scheme has resulted in movement of particles only to better locations in the input search space. The design of fitness function [22] is based on the dominance relations between the nodes of a program's control flow graph augmented with branch distance which produces a smoother landscape for guiding the search. This leads to faster and better convergence of test data to achieve the desired coverage. A neighborhood search strategy is also incorporated into the proposed hybrid algorithm that further helps in overcoming the problem of boundary constraints and local optima by exploring more promising candidate solutions. This is the main contribution of this paper. The proposed hybrid algorithm generates test data for one test requirement at a time; other test requirements are also checked for coverage thereby reducing the overall number of fitness evaluations.

3 Data flow analysis

In this study, data-flow coverage is used as the test adequacy criteria. Data-flow analysis [38] augments the control-flow testing criteria; the emphasis is on the definition and use of the variables in a program. The control flow of a program is represented by a directed graph G (V, E) also known as control flow graph (CFG), where V is the set of all the nodes and E is the set of all the edges in the graph. Each node corresponds to a program statement or group of sequential program statements and an edge represents flow of control from one node to another. There are two distinct nodes: an entry node n_0 and an exit node n_{end}. Node n dominates node m (dominance relationship) if every path from entry node n_0 to m contains n. By applying dominance relationship to all the nodes of CFG, a tree can be obtained that is rooted at n_0. This tree is called the dominator tree [39]. For each node m in the CFG, Dom (m) is the set of all the nodes that dominate node m. Figure 2 gives the CFG of the example program as given in Figure 1. The dominator tree is shown in Figure 3. For example, Dom (12) = {1, 2, 6, 7, 12}.

In a program, the definition and use occurrences of each variable are identified. A variable is said to be defined in a program statement (def-node) if a value is associated with the variable. A variable is said to be used in a program statement if its value is referenced for computational use (c-use node) or a predicate use (p-use node). Data-flow testing should cause the traversal of def-clear sub-paths from the variable definition to either some or all of the p-uses, c-uses, or their combination. Empirically, the all-uses criterion has been shown to be most effective compared to the other data-flow criteria [40]. A def-clear path does not include any intermediate nodes containing other definitions of that variable (killing nodes). A def-clear path can be further

```
#include<stdio.h>
#include<conio.h>

1   1  void main() {
2   1  int a, b, c, valid;
3   1  printf("\nEnter the value of three sides: ");
4   1  scanf("%d %d %d", &a, &b, &c);
5   1  valid=0;
6   2  if((a>=0)&&(a<=100)&&(b>=0)&&(b<=100)&&(c>=0)
         &&(c<=100)) {
7   3                 if(((a+b)>c)&&((c+a)>b)&&((b+c)>a)) {
8   4                           valid=1;
9   5                 }
10  5     }
11  6     if (valid==1) {
12  7                 if ((a==b)&&(b==c))
13  8                       printf("\nEquilateral triangle.");
14  9                 else if ((a==b)||(b==c)||(c==a))
15  10                      printf("\nIsosceles triangle.");
16  11                else
17  11                      printf(("\nScalene triangle.");
18  12    } else {
19  13                printf("\n Invalid input ").;
20  14    }
21  15 }
```

Figure 1: Triangle classification program.

Table 1: List of variables and def-use occurrences in the example program

Variable	def Node	c-use Node	p-use Edge
a b c	1	None	2-3 2-6 3-4 3-5 7-8 7-9 9-10 9-11
valid	1,4	None	6-7 6-13

Table 2: List of def-use paths for the example program.

Path No.	def-use Path (Terminates with -1 for c-use)	Killing Node(s)
1	1-2-3	None
2	1-2-6	None
3	1-3-4	None
4	1-3-5	None
5	1-7-8	None
6	1-7-9	None
7	1-9-10	None
8	1-9-11	None
9	1-6-7	4
10	1-6-13	4
11	4-6-7	None
12	4-6-13	None

categorized as a dcu-path (c-use of the variable) or a dpu-path (p-use of the variable). For the example program, Table 1 provides definition and use nodes for each variable, Table 2 provides the list of all-def-use paths and

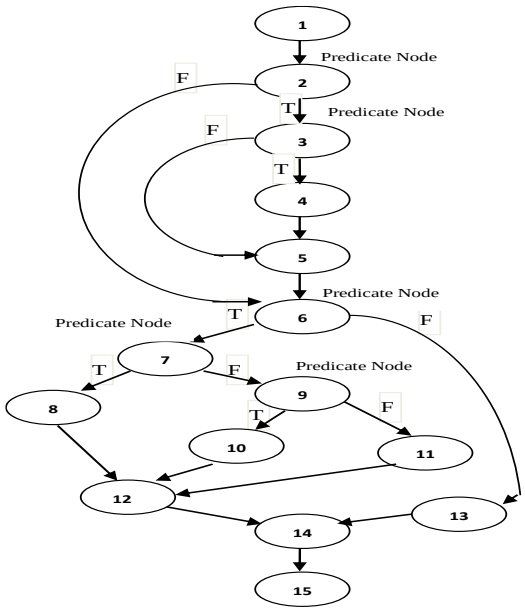


Figure 2: CFG of the example program.

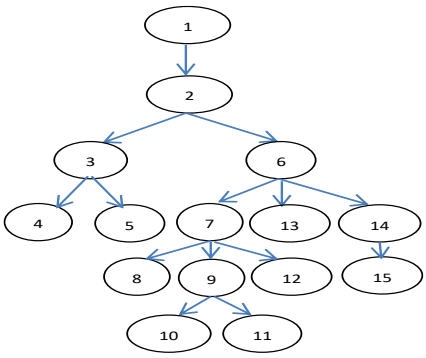


Figure 3: Dominator tree for the example

Table 3: Dominance paths for the nodes of the CFG.

Node No.	Dominance Path
1	1
2	1-2
3	1-2-3
4	1-2-3-4
5	1-2-3-5
6	1-2-6
7	1-2-6-7
8	1-2-6-7-8
9	1-2-6-7-9
10	1-2-6-7-9-10
11	1-2-6-7-9-11
12	1-2-6-7-12
13	1-2-6-13
14	1-2-6-14
15	1-2-6-14-15

Table 3 provides the dominance paths for the nodes of the program flow graph.

4 Particle swarm optimization

In 1995, Kennedy and Eberhart [41] introduced Particle Swarm Optimization algorithm, a population-based search algorithm based on the social and cognitive behavior of different swarms such as flock of birds, herd of animals or school of fishes. The application of PSO for solving many continuous space problems in the field of Computer Science and Engineering has demonstrated its potential. Unlike GA, PSO does not use evolution operators such as crossover and mutation. Instead, each member of the swarm (called particle) attains optimal solution by learning from its own experience and the experience of other members of the swarm. Each particle maintains its current position, current velocity and the best position it has achieved so far, called pbest. The global best position of the swarm is called gbest. Both pbest and gbest are used by the particle in determining its next best position in the swarm. Thus, the knowledge of previous good solutions is retained by all the particles resulting in a faster convergence towards the optimal solution.

Consider a swarm of *n* particles denoted as (p_1, p_2... p_n). Position of the ith particle in the d-dimensional search space is denoted as X_i = (Xi^1, X_i^2…X_i^d) and the associated velocity is denoted as V_i = (V_i^1, V_i^2…V_i^d). The personal best position of the ith particle in dimension *d* is denoted as $pbest_i^d$. The position of the best particle of the entire swarm in dimension *d* is denoted as $gbest^d$. The velocity and position of the ith particle in dimension *d* can be updated by Equations 1 and 2 as given below.

$$V_i^d = w \times V_i^d + c_1 \times r_1 \times (pbest_i^d - X_i^d) + c_2 \times r_2 \times (gbest^d - X_i^d) \quad (1)$$

$$X_i^d = X_i^d + V_i^d \quad (2)$$

where, c_1 and c_2 are positive learning constants called cognitive and social scaling parameters chosen in such a way that their sum never exceeds 4, and r_1 and r_2 are two random numbers in the range [0,1]. The inertia weight *w* controls the impact of the previous history on the new velocity of the ith particle. A particle's velocity in each dimension is clamped to a maximum magnitude V_{max}. The position and velocity of each particle in the swarm are continuously updated until an optimal solution is achieved.

4.1 Adaptive inertia weight

In PSO algorithm, a large value of inertia weight facilitates *exploration* (global search) of the input search space and a small value of inertia weight facilitates *exploitation* (local search) of the input search space for the optimal solution. Various inertia weighting strategies used in the literature have been categorized into *constant*, *random*, *time varying* and *adaptive* inertia weight strategies [42]. In constant and random inertia weight strategies, value of inertia weight is either constant or is chosen randomly during the search. In time varying inertia weight strategies, inertia weight is defined as a function of time or iteration number. Here, value of inertia weight is independent of the state of the particles in the search space. In adaptive inertia weight strategies, state of the particles in the search space (feedback mechanism) is used to adjust the value of the inertia weight.

In this study, fitness value of the particles is used to adjust the inertia weight. Ratio $\boldsymbol{\alpha}$ of the particle's fitness to the average fitness of the swarm is calculated as shown in Equation 3 below:

$$\alpha = f_i / f_{max} \quad (3)$$

Here, f_i=fitness of ith particle and f_{max} is the maximum fitness achieved by the particles in the swarm.

The range of α is [0, 1]. For lower values of α, increasing inertia weight can strengthen the particle's search capability. For values of α that are closer to 1, smaller inertia weight should be used. The inertia weight w_i for the ith particle is therefore defined as a linear function of α and is calculated as follows:

$$w_i = 0.5 \times (1-\alpha) + 0.5 \quad (4)$$

The range of the inertia weight is [0.5, 1].

PSO is computationally inexpensive. The ability of PSO to balance between local exploitation and global exploration of the search space enhances searching ability and avoids premature convergence towards the optimal solution.

5 Differential evolution

Differential Evolution (DE) algorithm was given by Storn and Price [43] in 1995. It is a stochastic population-based global optimization algorithm that uses an evolutionary differential operator to create new offspring from parent chromosomes. Unlike GA, DE works upon real-valued chromosomes. The differential operator of DE replaces the classical crossover and mutation operators of GA.

Let's say, the initial population consists of *n* vectors denoted as (p_1, p_2... p_n). Position of the ith vector in the d-dimensional space is denoted as X_i = (Xi^1, X_i^2…X_i^d). These vectors are referred as chromosomes in DE. To change each chromosome *(target vector)*, a *difference vector* V_i is created. In the literature, there are various mutation schemes to create this vector. In this paper, DE/Rand/1 scheme is used. In this scheme, for each ith member X_i of the current population, three other members (say r_1, r_2 and r_3) are randomly chosen from the current population. Next, the scaled difference (mutation scaling factor F) of any two of the three vectors is added to the third one to obtain the difference vector V_i. The jth component of the difference vector is as given below:

$$v_{i,j} = x_{r1,j} + F \times (x_{r2,j} - x_{r3,j}) \quad (5)$$

To increase the population diversity, a 'crossover scheme' is applied. The difference vector exchanges its components with the target vector X_i to obtain the offspring/trial vector U_i. The most common crossover in DE is 'uniform crossover' as given below:

$$u_{i,j} = v_{i,j} \quad \text{if } rand(0,1) < CR$$
$$= x_{i,j} \quad \text{else} \quad (6)$$

CR is called the crossover constant.

The final step in DE algorithm is the fitness-based selection of either target vector or trial vector in the next generation. F and CR are the control parameters of DE. The performance of DE depends on the manipulation of *target vector* and *difference vector* in order to obtain a *trial vector*.

6 Proposed hybrid algorithm

In the proposed study, an adaptive PSO algorithm is hybridized with the DE algorithm incorporating local neighborhood search strategy. The synergy between PSO and DE algorithms has resulted in a more powerful global search algorithm. The local neighborhood search strategy helps in exploring more promising candidate solutions to overcome the problem of local optima.

In the proposed hybrid (adaptive PSO and DE) algorithm, a differential velocity term inspired by the DE mutation scheme is computed by taking the difference of the position vectors of any two distinct particles randomly chosen from the swarm. A random number r is generated between 0 and 1. If r is less than DE crossover probability, Equation 7 (given below) is used to update the velocity of a particle. In Equation 7, the cognitive term (second term) in Equation 1 is replaced by the differential term scaled by DE mutation scaling factor.

$$V_i^d = w \times V_i^d + F \times (x_j^d - x_k^d) + c_2 \times r_2 \times (gbest^d - X_i^d) \qquad (7)$$

Here, x_j and x_k denote the position of particles j and k respectively ($i \neq j \neq k$) that are randomly chosen from the swarm. A survival of the fittest mechanism is also followed by incorporating the greedy selection scheme of DE as given by Equation 6. Therefore, the particle either moves to a better location or remains at its previous position in the input search space. The current position of a particle will always be its best position.

The steps of the proposed hybrid (adaptive PSO and DE) algorithm are given in Figure 5. The flowchart is given in Figure 6. Inputs to the algorithm are an instrumented program, dominator tree of the program, list of def-use paths to be traversed and the killing nodes if any, number of input variables, domain range of each input variable, and the algorithmic parameters: population size, PSO acceleration parameters, PSO maximum velocity, DE mutation scaling factor and DE crossover probability. Adaptive inertia weight is used as given by Equations 3 and 4. For data-flow coverage criterion, the design of fitness function is explained in Section 6.2 below. Initial value of *pbest* and *gbest* is 0. The algorithm is run once for each uncovered def-use path. If the selected path is not covered by any member of the current population, fitness value is computed for each member. Accordingly, for each particle, the personal best position *pbest* and the global best position *gbest* can be updated. During the evolution process, particle's position and velocity is adjusted according to Equations 2 and 7 respectively. If the updated position of the particle is out of input domain range, a local neighbourhood strategy is applied. Then, the greedy selection scheme of DE is used to generate the new population. The evolution process continues until the termination criteria is met. The other uncovered paths are also checked for coverage. The output is an optimal test suite and a list of def-use paths marked as covered or uncovered, if any.

A tool is developed for instrumenting programs and to generate def-use paths. Dominator tree is generated manually. Infeasible paths, if any, are determined by careful analysis of the program.

6.1 Neighbourhood search strategy

Every meta-heuristic search algorithm suffers with the problem of local optima. Another issue related to meta-heuristic search algorithms is boundary constraints. There are no set mechanisms to deal with such problems. Hence, in this study, an effort is also made to handle the problems of local optima and boundary constraints and to improve the exploitation ability of the algorithm. A neighbourhood search strategy (Figure 4) is introduced to sample more promising candidate solutions to overcome these problems. It is summarized as follows:

Step 1: For each particle, Euclidean distance is calculated from the other particles in the input search space using the position of particles. Accordingly, other particles within a threshold Euclidean distance (determined by preliminary study to fine-tune the algorithmic parameters) form the neighbourhood. Euclidean distance between two particles X_i and X_j in the n-dimensional search space is given by the following equation:

$$d_{ij} = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \qquad (8)$$

Step 2: If a particle's new position is out of range, other particles in the neighbourhood are evaluated.

Step 3: The position of the particle is then replaced with that of the best particle in the neighbourhood instead of a random value.

This helps in exploring more promising candidate solutions.

6.2 Design of Fitness Function

Def-use associations can be represented as node-node fitness functions [28]. Def-use associations specify the node of definition and the node of use for the program variables in the CFG without specifying a concrete path between the nodes. This implies that the first objective to reach is the definition node and then the use node, without however, specifying a path through the CFG. The distance to a node is represented by the standard minimizing metric given below:

$$\text{node distance} = \text{approach level} + v(\text{branch distance}) \qquad (9)$$

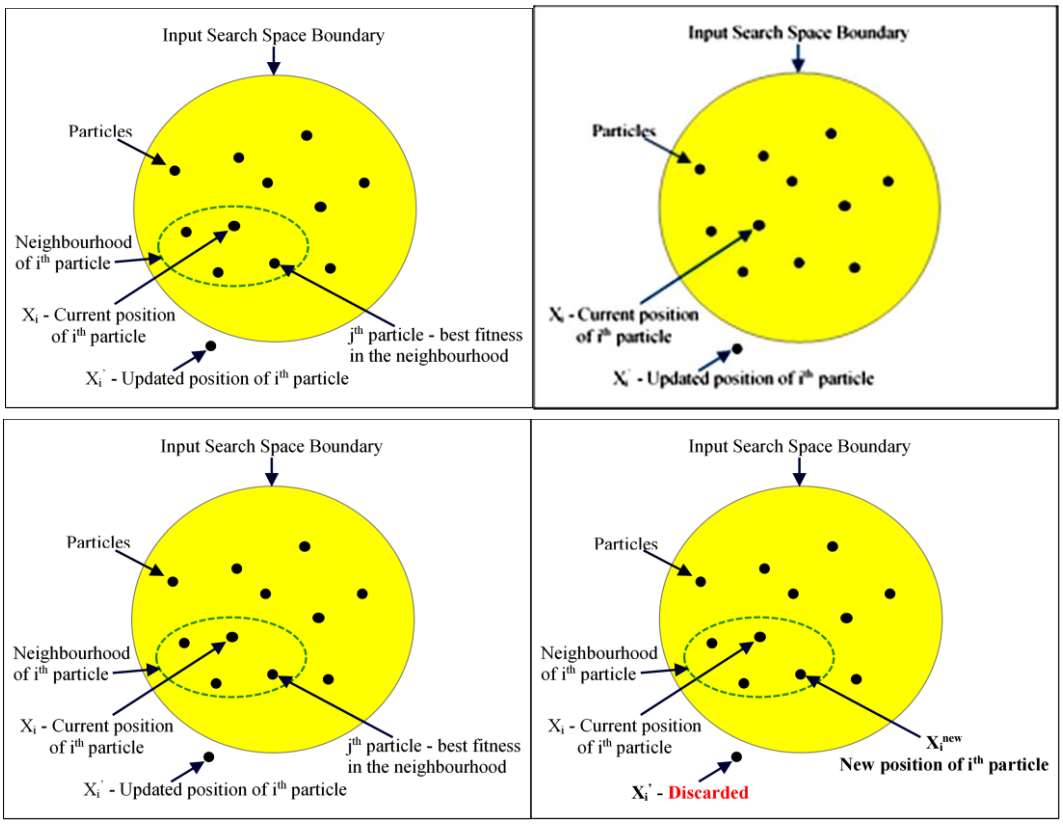


Figure 4: Local Neighbourhood Strategy.

It evaluates to 0 if the target is covered. Approach level is the closest point (a node) of a given execution to the target node. A branch is said to be *critical* if it leads the program execution away from the target node in a path through the program structure [44]; *branch distance* is calculated at that particular predicate node using values of the variables according to the formulae given in Table 4 [3] below.

Table 4: Branch distance measure for relational and logical predicates.

S. No.	Predicate (C)	Branch Distance Formulae: f(C)
1	Boolean	if true then 0 else K
2	x = y	if (x-y)=0 then 0 else abs(x-y)+K
3	x ≠ y	if abs(x-y)≠0 then 0 else K
4	x > y	if (y–x)<0 then 0 else (y-x)+K
5	x ≥ y	If (y–x)≤0 then 0 else (y-x)+K
6	x < y	if (x–y)<0 then 0 else (x-y)+K
7	x ≤ y	if (x–y)≤0 then 0 else (x-y)+K
8	C1 && C2	f(C1) + f(C2)
9	C1 || C2	min(f(C1), f(C2))

K is a failure constant that is added to branch distance if predicate is false

Branch distance provides a measure of how close the program execution was to traverse the alternate edge of the critical branch. Branch distance is normalized in the range [0, 1] using a normalization function *v*, such that the approach level always dominates the branch distance.

In our previous study [22], a novel maximizing fitness function is proposed for data-flow coverage adequacy criterion based on the standard metric (Equation 9) and dominator tree. Dominance relations between the nodes of the CFG are used to obtain path-cover for the nodes of the selected def-use path. The fitness function considers each def-use path as two objectives. For a dcu-path, the first objective is to cover the dominance path of the *definition* node and then to cover the dominance path of the *use* node. For a dpu-path, the first objective is to cover the dominance path of the *definition* node and then to cover the dominance paths of the nodes of the p-use *edge* (u_1, u_2). A dpu-path is formed for both the branches (T/F) of the predicate node. A test case is evaluated with respect to the selected def-use path by executing the program under test with it as an input and recording the nodes that are covered. If a killing node is traversed between the source node and the use node, a fitness value of 0 is assigned to the test case and it is discarded. The fitness value is 1 if all the nodes of the dominance paths of both the objectives are covered; otherwise closeness of the test case to the missed objective (*branch distance*) is computed.

In this work, for fitness maximization, branch distance $bch(x, t_i)$ at the critical branch for test case t_i and target node x is the reciprocal of the value returned by an appropriate formula from Table 4 i.e. the closer a test case is to cover the required branch, higher is its fitness value. The fitness function uses control-flow information (dominance relations between the nodes of the CFG) augmented with branch distance if a partial aim is achieved. This provides a smoother landscape/guidance to the search process towards the optimal solution. Branch distance is computed using Equation 10 and the

fitness functions are given by Equations 11 and 12 as explained below.

Branch distance *bch (x, t_i)* for test case t_i *(i=1...p)* and target node *x*, for fitness maximization, is calculated as follows:

$$\mathrm{bch}(x, t_i) = \begin{cases} 1 & \text{if the test case } t_i \text{ leads to the target node } x \\ \frac{1}{f(C)} & \text{otherwise, using an appropriate formula from Table 4 for the predicate C at the critical branch} \end{cases} \quad (10)$$

The fitness function to evaluate the fitness of a test case t_i (i=1...p) w.r.t. a dcu-path (d, u, v), where d is the *definition node* and u is the *c-use node* of a variable v, is given below:

$$\mathrm{ft}(d, u, t_i) = \frac{1}{2} \times \left(\frac{|\mathrm{cdom}(d, t_i)|}{|\mathrm{dom}(d)|} \times \mathrm{bch}(d, t_i) + \frac{|\mathrm{cdom}(u, t_i)|}{|\mathrm{dom}(u)|} \times \mathrm{bch}(u, t_i) \right) \quad (11)$$

Similarly, the fitness function to evaluate the fitness of a test case t_i (i=1...p) w.r.t. a dpu-path (d, (u_1, u_2), v), where d is the *definition node* and (u_1, u_2) is the *p-use edge* of a variable v, is given below:

$$\mathrm{ft}(d, (u_1, u_2), t_i) = \frac{1}{3} \times \left(\begin{array}{l} \frac{|\mathrm{cdom}(d, t_i)|}{|\mathrm{dom}(d)|} \times \mathrm{bch}(d, t_i) + \frac{|\mathrm{cdom}(u_1, t_i)|}{|\mathrm{dom}(u_1)|} \\ \times \mathrm{bch}(u_1, t_i) + \frac{|\mathrm{cdom}(u_2, t_i)|}{|\mathrm{dom}(u_2)|} \times \mathrm{bch}(u_2, t_i) \end{array} \right) \quad (12)$$

In general,

- dom(x): set of nodes in the dominance path of the target node *x*
- cdom(x, t_i): set of nodes in dom(x) that are covered by test case t_i *(i=1...p)*
- bch(x, t_i): branch distance for test case t_i *(i=1...p)* and target node x using Equation 9

If a killing node is traversed, a fitness value of 0 is assigned to the test case t_i and it is discarded; otherwise Equation 11 or Equation 12 is used to compute the fitness value. Test case t_i is said to be optimal if its fitness value is 1 i.e. the target is covered.

Consider the def-use path# 5 (1, 7, 8) for coverage from Table 2. This is a dpu-path that tests for 'Equilateral triangle' condition. Node 1 (source) and the p-use edge (7, 8) (target) form the two objectives - their dominance paths to be covered by an input test case. There are three cases - if the dominance paths of both the nodes are covered, fitness value of the input test case is 1 and it is optimal. However, if a partial aim is covered (one of the two nodes) or none of the nodes is covered, fitness value of the input test case is computed using Equations 3.2 and 3.4.

From Table 3, the dominance paths of the nodes are as given below:

dom(d) = dom(1) = {1}

dom(u1) = dom(7) = {1, 2, 6, 7}

dom(u2) = dom(8) = {1, 2, 6, 7, 8}

Case 1: Input test case t_1 <2, 2, 2>

Path traversed {1, 2, 3, 4, 5, 6, 7, 8, 12, 15}

Dominance path of the definition node (node 1) is covered.

Dominance path of the first node of the p-use edge (node 7) is covered.

Dominance path of the second node of the p-use edge (node 8) is covered.

As the dominance paths of both the objectives are covered, the fitness value of the input test case using Equation 3.4 is 1; the input test case t_1 is therefore optimal.

Case 2: Input test case t_2 <2, 2, 1>

Path traversed {1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 15}

Dominance path of the definition node (node 1) is covered.

Dominance path of the first node of the p-use edge (node 7) is covered.

Dominance path of the second node of the p-use edge (node 8) is not covered; the critical node is node 7. The branch distance at node 7 using Equation 3.2 is bch (8, t_2) = 0.91

The fitness value of the input test case using Equation 3.4 is ft (1, (7, 8), t_2) = 0.91

Case 3: Input test case t_3 <1, 2, 4>

Path traversed {1, 2, 3, 5, 6, 12, 13, 14, 15}

Dominance path of the definition node (node 1) is covered.

Dominance path of the first node of the p-use edge (node 7) is not covered; the critical node is node 6. The branch distance at node 6 using Equation 3.2 is bch (7, t_3) = 0.91

Dominance path of the second node of the p-use edge (node 8) is not covered; the critical node is node 7. The branch distance at node 6 using Equation 3.2 is bch (8, t_3) = 0.91

The fitness value of the input test case using Equation 3.4 is ft (1, (7, 8), t_3) = 0.74

This case study shows that the input test case t_1 covers the selected def-use path# 5. The input test case t_2 covers the def node and the first node of the selected def-use path# 5 (partial aim). The input test case t_3 does not cover any of the two objectives for the selected def-use path# 5. Accordingly, ft (1, (7, 8), t_1) > ft (1, (7, 8), t_2) > ft (1, (7, 8), t_3). Thus, the input test cases are also ranked according to their fitness values.

7 Experimental setup

In this section, research questions, algorithmic parameters settings, details of the subject programs, and experimental results are provided. DE, PSO, GA and random search techniques are also implemented for comparison with the proposed hybrid (adaptive PSO and DE) algorithm.

7.1 Research questions

The following research questions are formulated to evaluate the performance of the proposed hybrid algorithm:

RQ1: How effective is the proposed hybrid (adaptive PSO and DE) algorithm for optimal test data generation to achieve 100% data-flow coverage of a program?

```
Algorithm ATDG_Hybrid_PSO_DE
Input:
    P            : Instrumented version of the program under test
    arg = (a1,a2,…,ad): Argument list of P encoded into a d-dimension position vector
    DT           : Dominator tree for the program P
    Paths        : List of test requirements i.e. def-use paths
    Popinit      : Initial random population of n particles Xi = [Xi1, Xi2…Xid] and their velocities V = [Vi1, Vi2…Vid] for i=1, 2…n
    c1, c2, Vmax : Algorithmic parameters of Particle Swarm Optimization (PSO) algorithm
    F, CR        : Algorithmic parameters of Differential Evolution (DE) algorithm
Output:
    TestSuite    : Set of optimal test cases
    Pathstat     : List of test requirements marked as 'covered' and 'could not be covered' (if any)
Begin
1.  Popold = Popinit
2.  Popcur = Popinit
3.  while some pathi in Paths is not marked {
4.      while (termination criterion is not met) {      //Either pathi is covered or MaxAttempts
5.          for each particle i of Popcur {
6.              Decode position vector Xi into a test case ti
7.              if pathi is not marked {
8.              Check pathi for coverage w.r.t. ti and calculate fitness value using Eq. 10 or Eq. 11
9.                  if pathi is covered {
10.                     Mark pathi as 'covered' (update Pathstat)
11.                     Add ti to TestSuite
12.                 }
13.             }
14.             for each pathj of TestReq other than pathi that is not marked {
15.                 Check pathj for coverage with respect to ti
16.                 if pathj is covered
17.                     Mark pathj as 'covered' (update Pathstat)
18.             }
19.         }
20.         if pathi is covered
21.             Go to line 3
22.         else {
23.             Update gbestij
24.             for each particle i of Popcur {          //Generate a new population Popnew
25.                 Calculate inertia weight w using Equations 3 and 4
26.                 Randomly choose two distinct particles k and l from Popcur (i≠k≠l)
27.                 for each dimension j (1≤j≤d) of particle i{
28.                     Update pbestij
29.                     Randomly generate r between 0 and 1
30.                     if r<CR{
31.                         Calculate the difference between the jth components of the position vectors of particle k and particle l
32.                         Update velocity Vij of particle i in dimension j using Eq. 7
33.                         Clamp velocity Vij within the range [-Vmax, Vmax]
34.                     }
35.                     Update position Xij of particle i in dimension j using Eq. 2          //Offspring
36.                     if new position Xij of particle i in dimension j is out of range {
37.                         Apply neighbourhood strategy to particle i - according to Euclidean distance (Eq.8)
38.                         New position Xij of particle i in dimension j is the position of the best particle in the neighbourhood
39.                     }
40.                 }
41.                 Calculate fitness value of Offspring using Eq. 10 or Eq. 11
42.                 if Offspring is better than the parent Xi
43.                     Include Offspring in new population Popnew
44.                 else
45.                     Include parent Xi in new population Popnew
46.             }
47.             Popold = Popcur
48.             Popcur = Popnew
49.         }
50.     }
51.     if selected pathi could not be covered
52.         Mark pathi as 'could not be covered'
53. }
54. Return TestSuite, Pathstat
End
```

Figure 5: Proposed hybrid (adaptive PSO and DE) test data generation algorithm.

RQ2: How effective is the proposed hybrid (adaptive PSO and DE) algorithm for optimal test data generation with respect to the convergence speed (mean number of generations) at termination?

7.2 Parameters tuning

A preliminary study was carried out to determine the appropriate value of the algorithmic parameters and threshold value for Euclidean distance. Population sizes

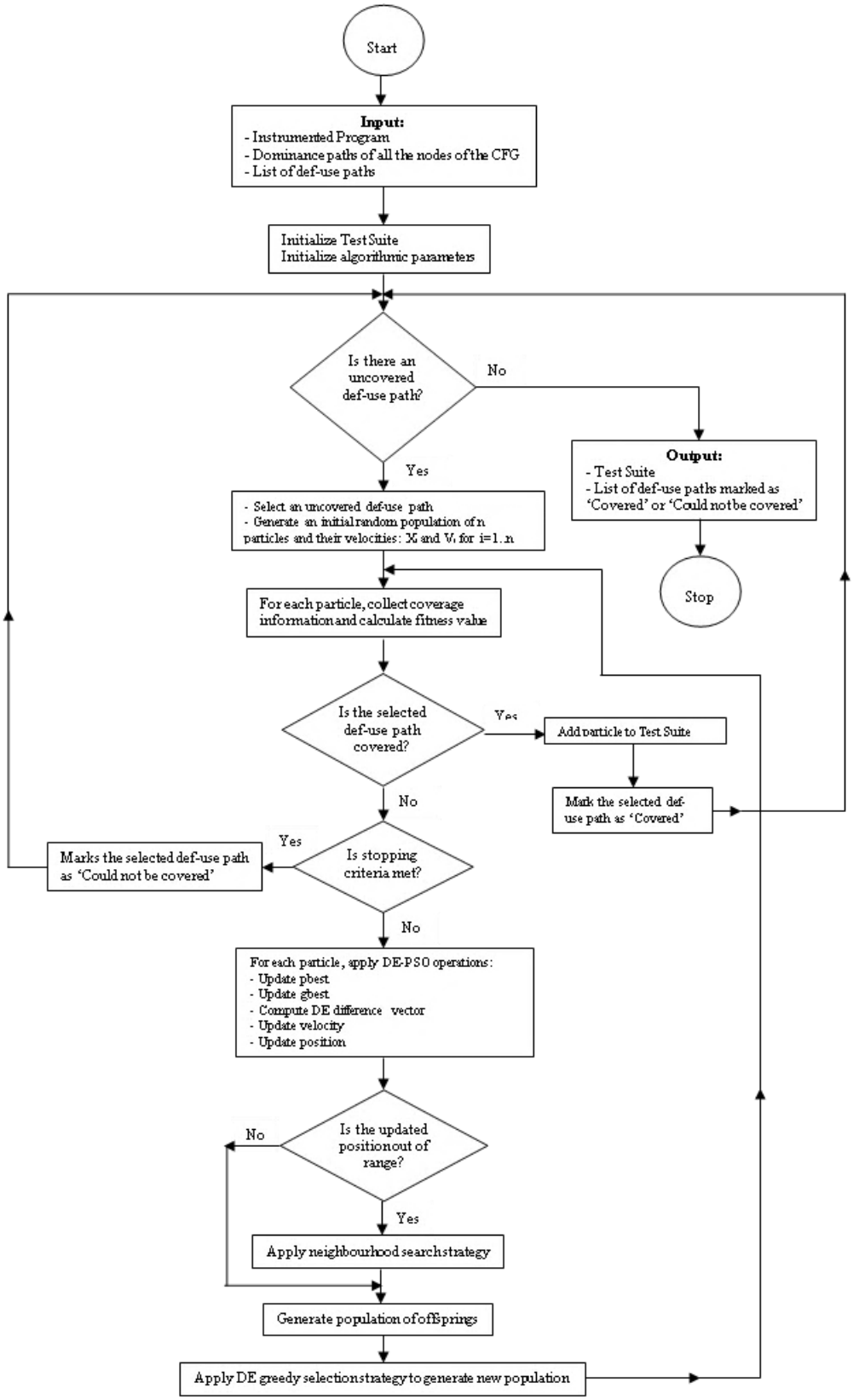


Figure 6: Flowchart of the proposed hybrid (adaptive PSO and DE) test data generation algorithm.

Table 5: Algorithmic parameter settings

Algorithm	Parameters	Value
Common Parameters	Population Size	10, 15, 20, 25
	Maximum number of generations	10^3
	Number of experiments for each program	100
	Fitness Function	As given by Eq. 11 and Eq. 12
	Threshold Euclidean distance	10
DE	Mutation Scaling Factor: F	1
	Crossover Constant: CR	0.9
PSO	Inertia weight	Adaptive as given by Eq. 3 and Eq. 4
	Acceleration constants: c1 and c2	c1=c2=2.0
	Maximum velocity: Vmax	Varies according to the program
GA	Chromosome encoding	Gray encoding
	Parent selection strategy	Roulette Wheel
	Probability of crossover	0.8
	Probability of mutation	0.15

considered are 10, 15, 20 and 25. 'Triangle Classifier' program is used as the pilot benchmark program and 100 experiments were carried out. Accordingly, in the main experiments, the following parameters settings have been used for adaptive PSO, DE and GA:

7.3 Subject programs

For this study, various benchmark programs have been selected from other researchers' work [6, 7, 13, 26] in the area of SBST. Experiments are also performed on programs taken from the SIR repository [45]. Source code of the academic programs is taken from standard reference books [38, 46, 47, 48]. The programs, as given in Table 6 below, have diverse structural elements such as loops, equality conditions, logically connected and nested predicates. A tool has also been developed for the instrumentation of programs and for listing of def-use paths.

7.4 Study results

This section presents the experimental results for various subject programs. For each subject program and each testing approach, 100 experiments were carried out. The measures collected are as follows:

- Mean number of generations: Sum of the number of generations at termination for each experiment over the total number of experiments gives the mean number of generations for a particular subject program.

  Here, termination criteria is either 100% data-flow coverage or 10^3 generations, whichever occurs first. Maximum number of generations is set to 10^3. For more complex programs, the maximum number of generations may be increased. Mean number of generations, however, is not indicative of full data-flow coverage.

Table 6: Subject programs

Program	#def-use Paths	Description	Type
1. Triangle Classifier	12	Finds the type of a triangle	Academic
2. Quadratic Equation	20	Finds the roots of a quadratic equation	Academic
3. Previous Date	66	Finds the previous date of a given date	Academic
4. Day of the Calendar	80	Finds the day on a given date	Academic
5. Marks Processing	19	Finds the final grade and average marks	Academic
6. Banking Transaction System	77	Banking transactions	Industrial
7. Sort	15	Sorting an array	Repository
8. Vector	26	Vector operations	Repository
9. Stack	20	Stack operations	Repository
10. Linked List	35	Linked list operations	Repository

- Mean percentage coverage: Sum of the data-flow coverage achieved for each experiment over the total number of experiments gives the mean percentage coverage achieved for a particular subject program. A def-use path is marked as covered the first time it is traversed and is not checked subsequently. The overall number of fitness evaluations is therefore reduced as stated in Section 2.

  If a path is infeasible, then some c-uses and p-uses that require this path to be traversed might also be infeasible [38]. For each program, infeasible uses, if any, were excluded while measuring data-flow coverage.

7.4.1 Effect of varying population size on the performance of the proposed hybrid (adaptive PSO and DE) algorithm

In this section, the effect of varying population size on the performance of the proposed hybrid algorithm with adaptive inertia weight and neighbourhood search strategy is analyzed. The performance is also compared with other meta-heuristic techniques and random search. The proposed hybrid algorithm, DE, PSO, GA (all guided by the same fitness function) and random search is applied to the various subject programs and experimental results are collected for the different measures. Population sizes that are considered are 10, 15, 20 and 25. Detailed experimental results are presented in Figures 7-16 below.

7.4.2 Overall comparison

In this section, overall performance of the proposed hybrid (adaptive PSO and DE) algorithm is compared with DE, PSO, GA and random search with respect to the measures collected. Tables 7 - 10, as given below, summarize the results of applying the various testing approaches to the set of chosen subject programs for

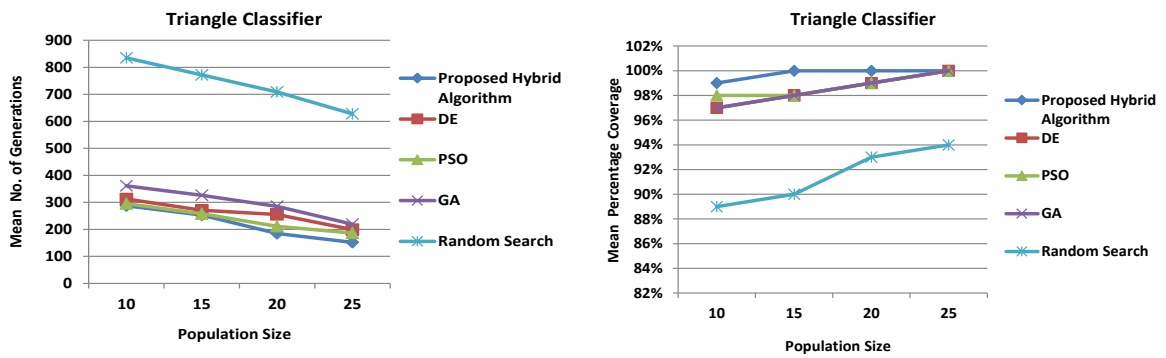


Figure 7: Graphs for 'Triangle Classifier' program.

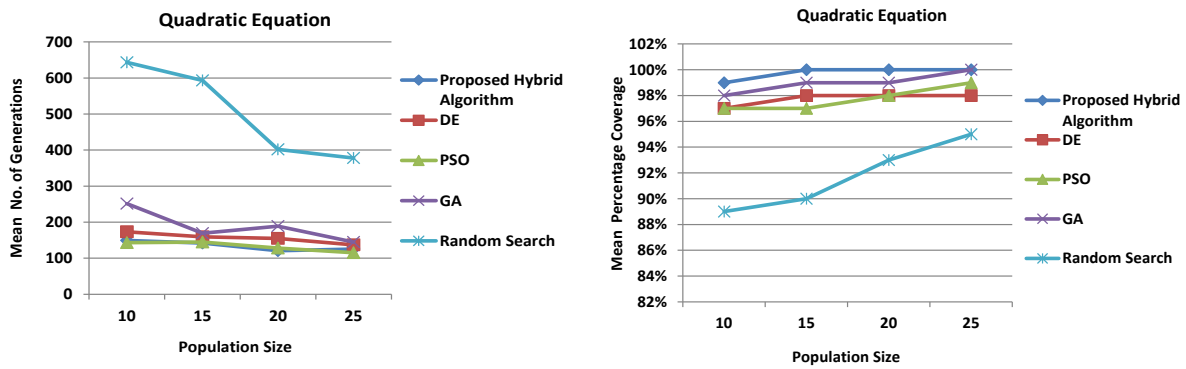


Figure 8: Graphs for 'Quadratic Equation' program.

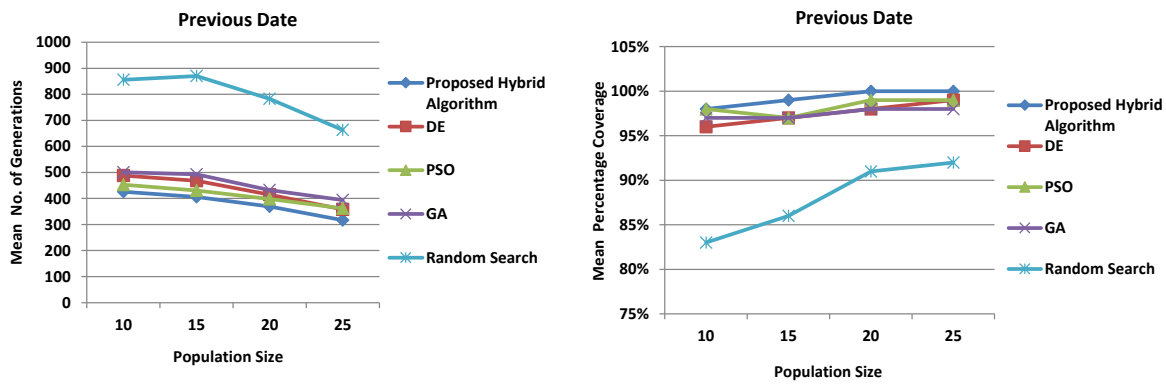


Figure 9: Graphs for 'Previous Date' program.

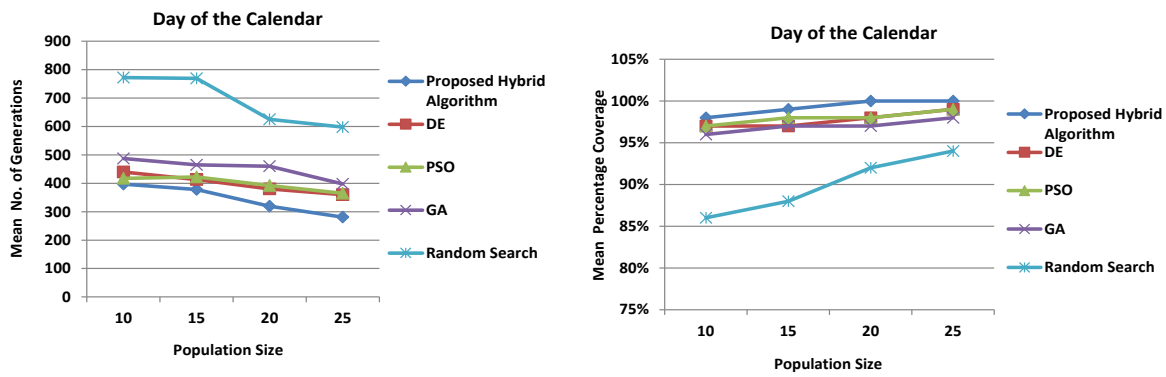


Figure 10: Graphs for 'Day of the Calendar' program.

different population sizes (10, 15, 20, 25). Range of the input integer variables is taken to be 0-100; range is different for variables of Program# 3, 4, and 7 as per the requirement of each program. The results are further discussed in the next section.

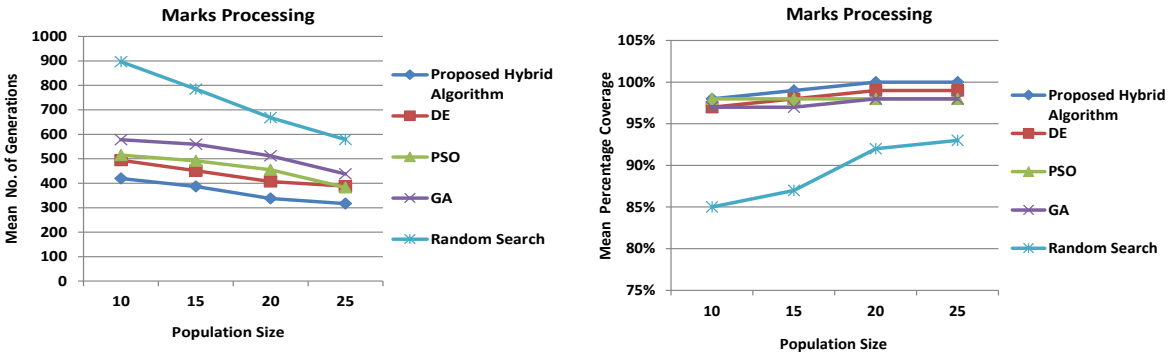


Figure 11: Graphs for 'Marks Processing' program.

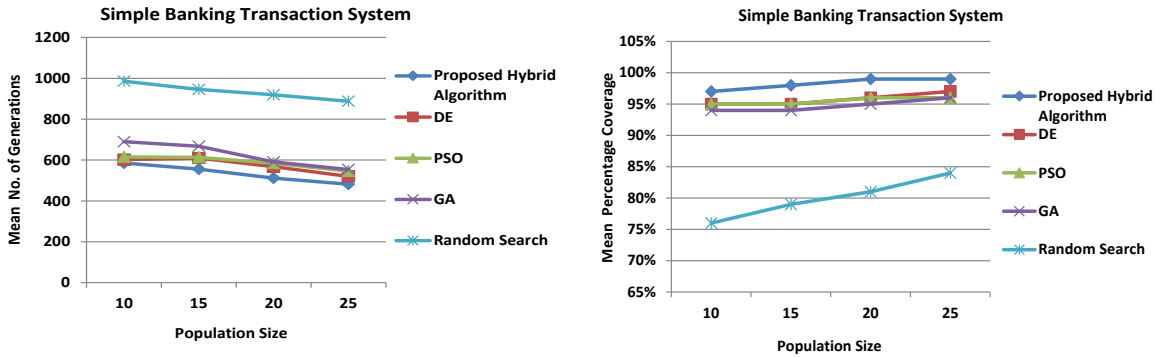


Figure 12: Graphs for 'Simple Banking Transaction System' program.

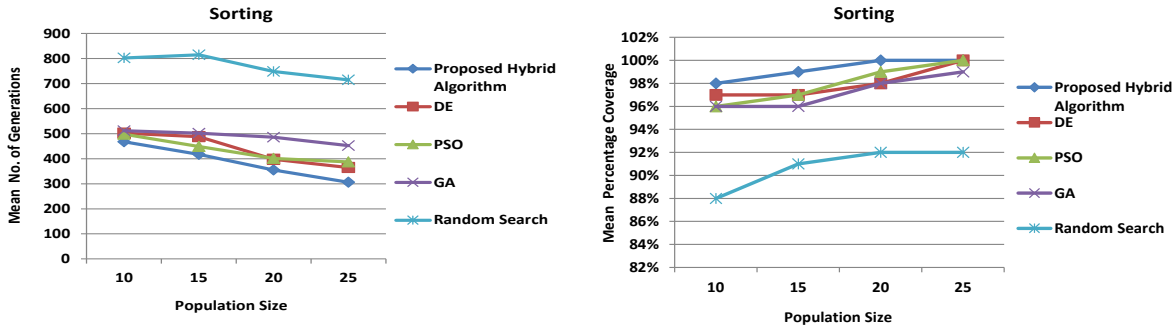


Figure 13: Graphs for 'Sort' program.

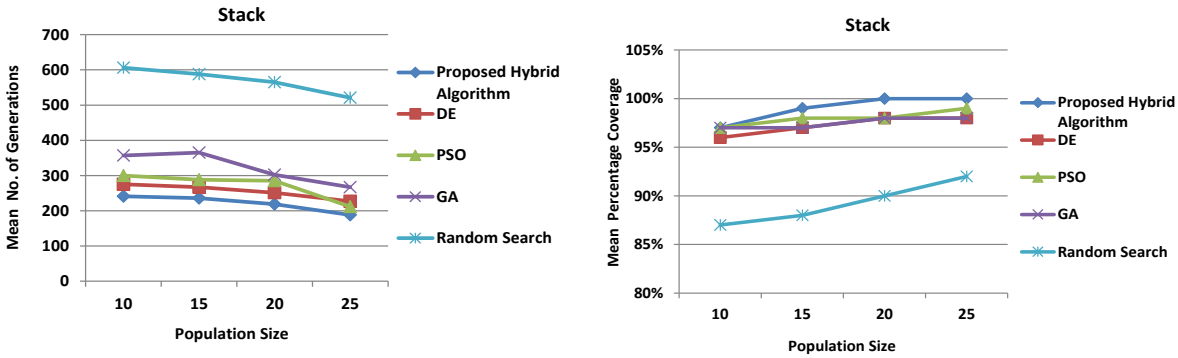


Figure 14: Graphs for 'Stack' program.

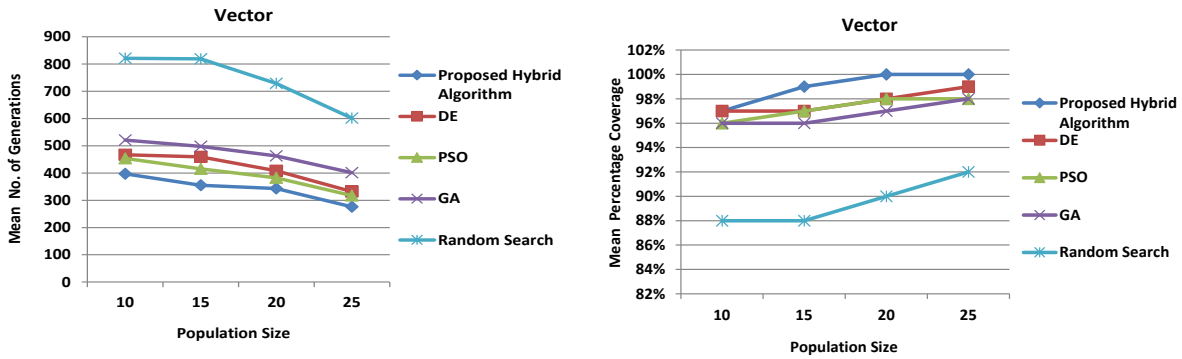


Figure 15: Graphs for 'Vector' program.

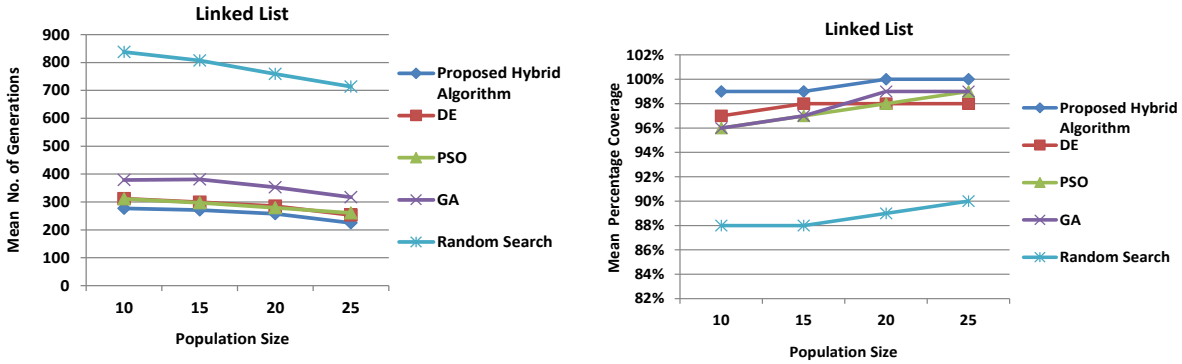


Figure 16: Graphs for 'Linked List' program.

Table 7: Experimental results for **Population Size 10**: Mean number of generations and mean percentage coverage.

Program	Measure									
	Mean Number of Generations					Mean Percentage Coverage				
	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search
Triangle Classifier	287	312	295	361	835	99%	97%	98%	97%	89%
Quadratic Equation	289	320	316	353	743	99%	97%	98%	98%	90%
Previous Date	426	488	453	501	856	98%	97%	98%	97%	85%
Day of the Calendar	397	440	417	487	772	98%	97%	97%	96%	86%
Marks Processing	419	494	515	578	897	98%	97%	98%	97%	85%
Simple Banking Transaction System	585	602	615	690	986	97%	95%	95%	94%	76%
Sort	468	502	498	512	802	98%	97%	96%	96%	88%
Vector	397	467	454	521	821	97%	97%	96%	96%	88%
Stack	241	275	300	357	606	97%	96%	97%	97%	87%
Linked List	277	312	311	379	838	99%	97%	96%	96%	88%

Table 8: Experimental results for **Population Size 15**:
Mean number of generations and mean percentage coverage.

Program	Measure									
	Mean Number of Generations					Mean Percentage Coverage				
	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search
Triangle Classifier	253	271	258	326	772	99%	97%	98%	97%	89%
Quadratic Equation	280	297	288	329	699	99%	98%	99%	99%	91%
Previous Date	406	468	431	493	870	99%	97%	98%	98%	87%
Day of the Calendar	378	413	422	465	769	99%	97%	98%	97%	88%
Marks Processing	387	451	492	559	785	99%	98%	98%	97%	87%
Simple Banking Transaction System	555	610	613	667	945	98%	95%	95%	94%	79%
Sort	417	488	449	502	815	99%	97%	97%	96%	91%
Vector	355	459	415	498	819	99%	97%	97%	96%	88%
Stack	236	267	288	365	588	99%	97%	98%	97%	88%
Linked List	271	299	297	381	807	99%	98%	97%	97%	88%

Table 9: Experimental results for **Population Size 20**:
Mean number of generations and mean percentage coverage.

Program	Measure									
	Mean Number of Generations					Mean Percentage Coverage				
	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search
Triangle Classifier	185	255	211	285	709	100%	99%	99%	99%	93%
Quadratic Equation	246	255	241	289	502	99%	99%	99%	99%	93%
Previous Date	369	415	398	432	783	100%	98%	99%	98%	91%
Day of the Calendar	319	380	392	460	625	100%	98%	98%	97%	92%
Marks Processing	338	407	455	512	668	100%	99%	98%	98%	92%
Simple Banking Transaction System	512	568	584	591	919	99%	96%	96%	95%	81%
Sort	355	398	401	486	748	100%	98%	99%	98%	92%
Vector	343	408	382	463	729	100%	98%	98%	97%	90%
Stack	219	251	285	302	565	100%	98%	98%	98%	92%
Linked List	258	285	279	353	759	100%	98%	99%	99%	90%

Table 10: Experimental results for **Population Size 25**:
Mean number of generations and mean percentage coverage.

Program	Measure									
	Mean Number of Generations					Mean Percentage Coverage				
	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search	Proposed Hybrid Algorithm	DE	PSO	GA	Random Search
Triangle Classifier	152	199	186	219	628	100%	100%	100%	100%	94%
Quadratic Equation	221	262	205	245	478	100%	99%	100%	100%	96%
Previous Date	317	358	362	394	663	100%	99%	99%	99%	93%
Day of the Calendar	281	360	365	398	598	100%	99%	99%	98%	94%
Marks Processing	317	388	383	438	579	100%	99%	98%	98%	93%
Simple Banking Transaction System	482	520	546	554	888	99%	97%	96%	96%	84%
Sort	306	365	387	452	715	100%	100%	100%	99%	92%
Vector	276	332	317	401	601	100%	99%	98%	98%	92%
Stack	188	227	212	267	521	100%	98%	99%	98%	92%
Linked List	225	253	261	317	714	100%	98%	99%	99%	90%

8 Discussion

The experimental results have been presented above in Tables 7-10 and Figures 7-16. In context of the research questions formulated for this study, the experimental results are analysed and discussed in this section.

RQ1: How effective is the proposed hybrid (adaptive PSO and DE) algorithm for optimal test data generation to achieve 100% data-flow coverage of a program?

From the experimental results as shown in Tables 7-10, it can be seen that the proposed hybrid algorithm with adaptive inertia weight and neighbourhood search strategy, achieved highest mean percentage coverage for all the subject programs and for all population sizes that are considered. Only the proposed hybrid algorithm achieved 100% data-flow coverage for all the subject programs for population size 20 (except for Program# 2 and Program# 6) and for population size 25 (except for Program# 6). For population size 10 and 15 also, the mean percentage coverage is 97%-99% with the proposed hybrid algorithm. For each program, infeasible uses, if any, were not considered while measuring data-flow coverage. Infeasible uses, if any, are determined by careful manual analysis as it is not possible to write an algorithm for analyzing a given program to determine if a given element in the coverage domain is feasible or not [38]. This, in addition to the novel fitness function, adaptive inertia weight and neighbourhood search strategy has resulted in full data-flow coverage as the population size is increased from 10 to 25.

For the other meta-heuristic search techniques (DE, PSO and GA), all guided by the same fitness function, mean percentage coverage is between 94%-99% for all the subject programs and for all population sizes that are considered. DE achieved 100% data-flow coverage only for Program# 1 and Program# 7 for population size 25. PSO achieved 100% data-flow coverage only for Program# 1, Program# 2, and Program# 7 for population size 25. GA achieved 100% data-flow coverage only for Program# 1 and Program# 2 for population size 25. However, the proposed hybrid algorithm outperformed DE, PSO and GA with respect to the convergence speed in all the cases. Performance of random search is worst; mean percentage coverage achieved is minimum for all the subject programs for all population sizes that are considered. This provides an explanation for high mean number of generations when percentage coverage is less than 100% as then the algorithm terminates only after 10^3 generations.

RQ2: How effective is the proposed hybrid (adaptive PSO and DE) algorithm for optimal test data generation with respect to the convergence speed (mean number of generations) at termination?

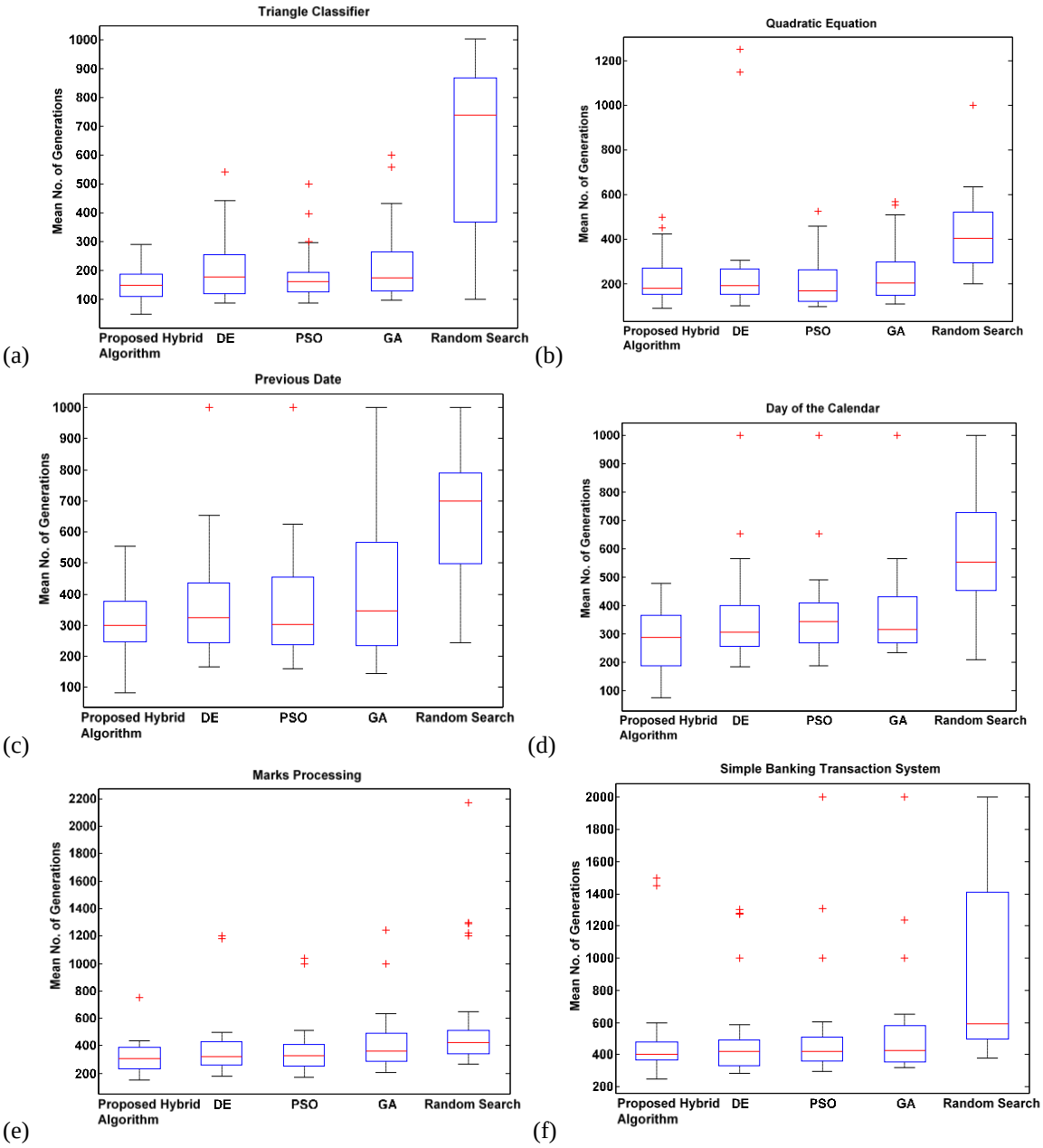
From the experimental results as shown in Tables 7-10, it can be seen that the mean number of generations is least with the proposed hybrid algorithm for all the subject programs and for all population sizes that are considered. There is a substantial reduction in mean number of generations with the proposed hybrid algorithm for benchmark programs such as ‘Triangle Classifier’, ‘Quadratic Equation’, and ‘Previous Date’

Table 11: Statistical results of Friedman Aligned and post hoc test (level of confidence α = 0.05)

Program	Testing Approach	Average Rank	Friedman Aligned Statistic	p-value by Friedman Aligned Test	p-value by applying Post Hoc Methods	Holm's Procedure α/i	Hypothesis
Triangle Classifier	**Proposed Hybrid Algorithm**	**34.5**	24.2877	6.994E-05	-	-	-
	DE	56.23			0.048691	0.05	**Rejected**
	PSO	63.43			0.0099	0.025	**Rejected**
	GA	89.23			0.000001	0.016667	**Rejected**
	Random Search	134.1			0	0.0125	**Rejected**
Quadratic Equation	**Proposed Hybrid Algorithm**	**52.03**	24.2141	7.236E-05	0.799444	0.05	**Not Rejected**
	DE	68.95			0.078049	0.025	**Not Rejected**
	PSO	49.18			-	-	-
	GA	80.03			0.005957	0.016667	**Rejected**
	Random Search	127.3			0	0.0125	**Rejected**
Previous Date	**Proposed Hybrid Algorithm**	**31.78**	24.183394	7.339 E-05	-	-	-
	DE	52.57			0.063918	0.05	**Not Rejected**
	PSO	63.02			0.005364	0.025	**Rejected**
	GA	97			0	0.016667	**Rejected**
	Random Search	133.13			0	0.0125	**Rejected**
Day of the Calendar	**Proposed Hybrid Algorithm**	**35.1**	24.151826	7.447E-05	-	-	-
	DE	58.15			0. 039897	0.05	**Rejected**
	PSO	66.42			0. 005242	0.025	**Rejected**
	GA	85.22			0. 000008	0.016667	**Rejected**
	Random Search	132.62			0	0.0125	**Rejected**
Simple Banking Transaction System	**Proposed Hybrid Algorithm**	**33.2**	23.557598	9.795 E-05	-	-	-
	DE	55.08			0.0470	0.05	**Rejected**
	PSO	59.8			0.017726	0.025	**Rejected**
	GA	96.32			0	0.016667	**Rejected**
	Random Search	133.1			0	0.0125	**Rejected**
Marks Processing	**Proposed Hybrid Algorithm**	**25.68**	23.903359	8.352E-05	-	-	-
	DE	49.28			0.035392	0.05	**Rejected**
	PSO	66.58			0.000266	0.025	**Rejected**
	GA	101.38			0	0.016667	**Rejected**
	Random Search	134.57			0	0.0125	**Rejected**
Sort	**Proposed Hybrid Algorithm**	**34.61**	24.028397	7.883E-05	-	-	-
	DE	57.88			0.038067	0.05	**Rejected**
	PSO	65.55			0.005823	0.025	**Rejected**
	GA	90.38			0.000001	0.016667	**Rejected**
	Random Search	129.07			0	0.0125	**Rejected**
Vector	**Proposed Hybrid Algorithm**	**34.7**	24.235764	7.164E-05	-	-	-
	DE	61.73			0.015956	0.025	**Rejected**
	PSO	56.83			0.048484	0.05	**Rejected**
	GA	99.33			0	0.016667	**Rejected**
	Random Search	124.9			0	0.0125	**Rejected**
Stack	**Proposed Hybrid Algorithm**	**32.52**	23.829629	8.641E-05	-	-	-
	DE	53.25			0.06456	0.05	**Not Rejected**
	PSO	70.13			0.000798	0.025	**Rejected**
	GA	92.77			0	0.016667	**Rejected**
	Random Search	128.83			0	0.0125	**Rejected**
Linked List	**Proposed Hybrid Algorithm**	**34.78**	23.551908	9.821E-05	-	-	-
	DE	60.65			0.021116	0.025	**Rejected**
	PSO	59.48			0.027672	0.05	**Rejected**
	GA	91.53			0	0.016667	**Rejected**
	Random Search	131.05			0	0.0125	**Rejected**

that have multiple and nested conditions along with equality conditions. This is also true for other programs taken from the repository [45] such as 'Sort', 'Stack', 'Vector', and 'Linked List'. As expected, the mean number of generations decreases as the population size increases due to a wider search space.

The performance of random search is worst with respect to the mean number of generations to achieve same data-flow coverage for smaller population sizes and



for programs with multiple and nested conditions. Random search did not achieve full data-flow coverage for any of the subject program. This has resulted in higher values for the measure 'mean number of generations' at termination.

It can be inferred that the proposed hybrid algorithm with adaptive inertia weight and neighbourhood search strategy is the best performing approach for all the subject programs and for all population sizes that are considered with respect to the measures collected. The proposed hybrid algorithm and the other meta-heuristic search techniques (DE, PSO and GA) are all guided by the same novel fitness function; the better performance of the proposed hybrid algorithm can be attributed to the inclusion of adaptive inertia weight and neighbourhood search strategy.

8.1 Statistical analysis on repeated trials

Statistical analysis is performed to validate the effectiveness and efficiency of the proposed hybrid (adaptive PSO and DE) algorithm with adaptive inertia weight and neighbourhood search strategy over other meta-heuristic search techniques (DE, PSO and GA) and random search applied for test data generation in accordance to data-flow coverage criterion. The experiment on each subject program was repeated 100 times. From the experimental results as presented in Section 7.4, it can be seen that the proposed hybrid algorithm as well as the other meta-heuristic search techniques (DE, PSO and GA), all guided by the same fitness function, have comparable results with respect to the measure 'mean percentage coverage' for population size 10 and 15. The proposed hybrid algorithm achieved 100% data-flow coverage for all the subject programs for

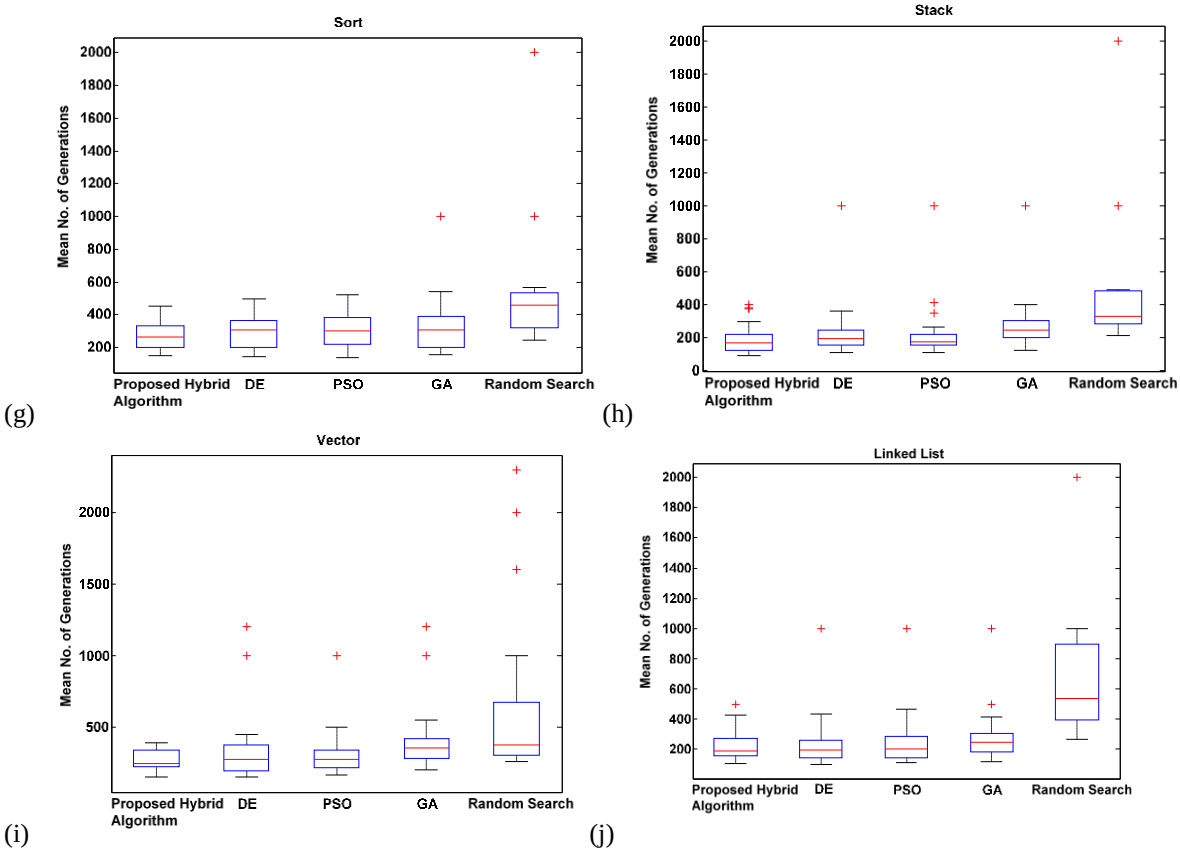


Figure 17: Stability analysis for the measure 'mean number of generations'.

population size 20 (except for Program# 2 and 6) and for population size 25 (except for Program# 6). For population size 25, DE achieved 100% data-flow coverage only for Program# 1 and 7; PSO achieved 100% data-flow coverage only for Program# 1, 2, and 7; GA achieved 100% data-flow coverage only for Program# 1 and 2. Therefore, the convergence speed (mean number of generations) information for population size 25 (best performance for all the approaches) is used for statistical difference test.

In the first step, Friedman Aligned 1xN test, a non-parametric multiple comparison statistical test [49], is applied to check for significant differences between the performance of the proposed hybrid algorithm and the other algorithms. Average rankings of all the algorithms are obtained that provide a fair comparison of the algorithms; a low value indicates higher rank. The unadjusted p-value is also computed through normal approximations; the smaller the p-value, the stronger the evidence against the null hypothesis. The value of α (level of confidence) is set to 0.05. In the second step, if the null hypothesis of equivalence of rankings is rejected, a post hoc test (Holm's procedure) is applied to report adjusted p-values by adjusting the value of α in a step-down manner to compensate for multiple comparisons. Here, the proposed hybrid algorithm acts as the control algorithm and its performance is compared with the rest of the algorithms used for comparison.

Results of the statistical analysis are summarized in Table 11 - average ranking of each algorithm, Friedman Aligned statistic, p-value computed by Friedman Aligned test and p-values obtained in by applying post hoc methods. It can be observed that the rank of the proposed hybrid algorithm is minimum (best performing algorithm) for all the subject programs except for 'Quadratic Equation' program. In case of 'Quadratic Equation' program, PSO is the best performing algorithm; however, PSO did not achieve full data-flow coverage and the proposed hybrid algorithm achieved full data-flow coverage as can be seen in Table 10. Random search gets the worst rank among all the algorithms as expected. The p-values computed by Friedman Aligned test are ≤ α (level of confidence) for all the subject programs, so the null hypothesis of equivalence of rankings can be rejected.

Further, p-values at the level of confidence α are reported by applying Holm's procedure to compensate for multiple comparisons. Holm's procedure rejects those hypotheses that have an unadjusted p-value ≤ α. As can be seen, all the null hypotheses are rejected in all the cases for all the subject programs except for 'Quadratic Equation', 'Previous Date' and 'Stack' programs. The null hypothesis is not rejected for DE in case of 'Quadratic Equation' (for proposed hybrid algorithm also), 'Previous Date' and 'Stack' programs. However, as can be seen from Tables 7 - 10, there is significant difference among the performance of all the algorithms being compared with respect to the measures collected. Thus, it be claimed that there is significant difference between the performances of the proposed hybrid algorithm and the other algorithms being compared.

For further analysis, box plots are drawn as shown in Figure 17 to compare the distribution of the measure *mean number of generations* over 100 trials for all the subject programs (population size 25). It can be observed that the median value of the measure 'mean number of generations' (in 100 trials) for the proposed hybrid algorithm is always less than the corresponding values for DE, PSO, GA and random search for all the subject programs except for 'Quadratic Equation' program. The median value is comparable with that of PSO for the 'Quadratic Equation' program. For all the approaches, the difference between the first quartiles as well as the difference between the third quartiles is quite visible.

It can therefore be concluded that the proposed hybrid (adaptive PSO and DE) algorithm is the best performing algorithm and is significantly different from the other algorithms (DE, PSO, GA and random search) being compared. The proposed hybrid (adaptive PSO and DE) algorithm has stronger ability to generate test data with higher data-flow coverage as well as convergence speed as compared to DE, PSO, GA and random search techniques.

9 Threats to validity and limitations

This section presents the possible validity threats [50] for the proposed study. Threats to internal validity are considered in the context of SBST. The choice of algorithmic parameters such as population size, inertia weight, acceleration constants, maximum velocity, mutation scaling factor, crossover constant affects the performance of the meta-heuristic search algorithms. Preliminary experiments were carried out to determine the appropriate values for the various algorithmic parameters for the proposed hybrid (adaptive PSO and DE) algorithm.

Threats to construct validity may arise from the fact that the performance of the proposed hybrid (adaptive PSO and DE) algorithm is evaluated with respect to the measures 'mean number of generations' and 'mean percentage coverage' for a particular subject program. Other measures such as total number of fitness evaluations and average search time may have also been used for evaluation.

Statistical analysis is performed to establish conclusion validity i.e. to validate the effectiveness and efficiency of the proposed hybrid (adaptive PSO and DE) algorithm over other techniques that have been considered for comparison. It is shown that the proposed hybrid (adaptive PSO and DE) algorithm is significantly different to DE, PSO, GA and random search that are considered for comparison; all except random search have been guided by the same fitness function. Adaptive inertia weight and neighbourhood search strategy have improved the performance of the proposed hybrid (adaptive PSO and DE) algorithm with respect to the measures collected. Threats to conclusion validity may arise from the fact that the infeasible uses / infeasible data-flow paths are identified and eliminated by manual analysis. Also, results for the proposed hybrid algorithm and other techniques have been compiled with respect to the experimental setup used for the present study.

The main external threat to validity is the choice of subject programs that may limit the generalization of results of the proposed study to real and more complex programs. Also, a different population size apart from those considered may produce different coverage results.

However, subject programs that are considered have many of the same programming constructs as large programs. The proposed approach should therefore be able to handle real and more complex programs. The claim is, however, a matter of further investigation.

10 Conclusion

Automated test data generation is still an open problem in spite of decades of research. In the field of SBST, GA has been the algorithm of choice for control-flow coverage criteria. Very recently only, other highly adaptive search-based techniques such as PSO have been employed for structural test data generation. DE is another simple to implement and highly adaptive search-based technique that has been not yet applied for automated test data generation. Among the structural test adequacy criteria, data-flow coverage test adequacy criterion has received relatively little attention. This paper presents a hybrid (adaptive PSO and DE) algorithm with neighbourhood search strategy for optimal test data generation in accordance to the all-uses data-flow coverage test adequacy criterion.

The performance of the proposed hybrid (adaptive PSO and DE) algorithm has been experimentally evaluated and compared with that of DE, PSO, GA and random search for data-flow coverage. It is shown that the proposed hybrid (adaptive PSO and DE) algorithm outperformed DE, PSO, GA and random search with respect to the measure 'mean number of generations' for all the population sizes that are considered. For the measure 'mean percentage coverage', performance of the proposed hybrid (adaptive PSO and DE) algorithm is comparable to that of DE, PSO and GA for smaller population sizes (10 and 15); however, only the proposed hybrid algorithm achieved full data-flow coverage as the population size is increased to 20 and 25 for complex subject programs. Performance of random search is worst. Here, we have explored a promising hybrid optimization algorithm for test data generation. In future, we intend to fine tune the algorithmic parameters and work upon more complex subject programs.

11 References

[1] H. Zhu, P.A.V. Hall, J.H.R. May (1997). Software unit test coverage and adequacy. *Computing Surveys*, ACM, Vol. 29, No. 4, pp. 366-427. https://doi.org/10.1145/267580.267590

[2] R.A. DeMillo and A.J. Offutt (1991). Constraint-based automatic test data generation. *IEEE Transactions on Software Engineering*, Vol. 17, No. 9, pp. 900-910. https://doi.org/10.1109/32.92910

[3] N. Tracey (2010). A Search-Based Automated Test-Data Generation Framework for Safety-Critical Software. *Doctoral Thesis*, University of York.

[4] X.S. Yang (2010). *Engineering Optimization: An Introduction with Metaheuristic Applications*. Wiley, New Jersey.

[5] M.A. Ahmed and I. Hermadi (2007). GA-based multiple paths test data generator. *Computers and Operations Research*, Elsevier, Vol. 35, No. 10, pp. 3107-3124. https://doi.org/10.1016/j.cor.2007.01.012

[6] A.S. Ghiduk, M.J. Harrold, and M.R. Girgis (2007). Using Genetic Algorithms to Aid Test-Data Generation for Data-Flow Coverage. *Proceedings of the 14th Asia-Pacific Software Engineering Conference*, IEEE, pp. 41-48. https://doi.org/10.1109/ASPEC.2007.73

[7] M.R Girgis (2005). Test Data Generation for Data Flow Testing Using a Genetic Algorithm. *Journal of Universal Computer Science*, Vol. 11, No. 6, pp. 898-915. http://dx.doi.org/10.3217/jucs-011-06-0898

[8] P. McMinn (2004). Search-Based Software Test Data Generation: A Survey. *Journal of Software Testing, Verification and Reliability*, Wiley, Vol. 14, No. 2, pp. 105-156. https://doi.org/10.1002/stvr.294

[9] R.P. Pargas, M.J. Harrold, and R. Peck (1999). Test-Data Generation Using Genetic Algorithms. *Journal of Software Testing, Verification and Reliability*, Wiley, Vol. 9, No. 4, pp. 263-282. https://doi.org/10.1002/(SICI)1099-1689(199912)9:4%3c263::AID-STVR190%3e3.0.CO;2-Y

[10] A. Windisch, S. Wappler, and J. Wegener (2007). Applying Particle Swarm Optimization to Software Testing. *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO'07)*, pp. 1121-1128. https://doi.org/10.1145/1276958.1277178

[11] K. Agarwal, A. Pachauri, and Gursaran (2008). Towards Software Test Data Generation using Binary Particle Swarm Optimization. *Proceedings of the XXXII National Systems Conference*, pp. 339-343.

[12] K. Agarwal and G. Srivastava (2010). Towards Software Test Data Generation using Discrete Quantum Particle Swarm Optimization. *Proceedings of the ISEC'10*, pp. 65-68. https://doi.org/10.1145/1730874.1730888

[13] C. Mao (2014). Generating Test Data for Software Structural Testing Based on Particle Swarm Optimization. *Arabian Journal of Science and Engineering*, Springer, Vol. 39, No. 10, pp. 4593-4607. https:/doi.org/10.1007/s13369-014-1074-y

[14] N. Nayak and D.P. Mohapatra (2010). Automatic Test Data Generation for Data Flow Testing Using Particle Swarm Optimization. *Springer-Verlag Heidelberg*, pp. 1-12.

[15] A. Arcuri and X. Yao (2008). Search based software testing of object-oriented containers. *Information Sciences*, Elsevier, Vol. 178, No. 15, pp. 3075-3095. https://doi.org/10.1016/j.ins.2007.11.024

[16] S. Zhang, Y. Zhang, H. Zhou, and Q. He (2010). Automatic path test data generation based on GA-PSO. *Proceedings of the International Conference on Intelligent Computing and Intelligent Systems (ICIS'10)*, IEEE, pp. 142-146. https://doi.org/10.1109/ICICISYS.2010.5658735

[17] K. Li, Z. Zhang, and J. Kou (2010). Breeding Software Test Data with Genetic-Particle Swarm Mixed Algorithm. *Journal of Computers*, Vol. 5, No. 2, pp. 258-265. https://doi.org/10.4304/jcp.5.2.258-265

[18] S. Singla, D. Kumar, H.M. Rai, and P. Singla (2011). A hybrid PSO approach to automate test data generation for data flow coverage with dominance concepts. *International Journal of Advanced Science and Technology*, Vol. 37, pp. 15-26.

[19] A. Kaur and D. Bhatt (2011). Hybrid particle swarm optimization for regression testing. *International Journal on Computer Science and Engineering*, Vol. 3, No. 5, pp. 1815-1824.

[20] M.R. Girgis, A.S. Ghiduk, and E.H. Abd-Elkawy (2015). Automatic Data Flow Test Paths Generation using the Genetical Swarm Optimization Technique. *International Journal of Computer Applications*, Vol. 116, No. 22, pp. 25-33.

[21] P. Chawla, I. Chana, and A. Rana (2015). A novel strategy for automatic test data generation using soft computing technique. *Frontier Computer Science*, Springer, Vol. 9, No. 3, pp. 346-363. https://doi.org/10.1007/s11704-014-3496-9

[22] S. Varshney and M. Mehrotra (2016). Search-based Test Data Generator for Data-Flow Dependencies using Dominance Concepts, Branch Distance and Elitism. *Arabian Journal of Science and Engineering*, Springer, Vol. 41, No. 3, pp. 853-881. https://doi.org/10.1007/s13369-015-1921-5

[23] B. Korel (1990). Automated Software Test Data Generation. *IEEE Transactions on Software Engineering*, Vol. 16, No. 8, pp. 870-879. https://doi.org/10.1109/32.57624

[24] P. G. Frankl and S. N. Weiss (1993). An Experimental Comparison of the Effectiveness of Branch Testing and Data Flow Testing. *IEEE Transactions on Software Engineering*, Vol. 19, No. 8, pp. 774-787. http://doi.ieeecomputersociety.org/10.1109/32.238581

[25] S. Ali, L.C. Briand, H. Hemmati, and R.K.P. Walawege (2010). A Systematic Review of the Application and Empirical Investigation of Search-Based Test Case Generation. *IEEE Transactions on Software Engineering*, Vol. 36, No. 6, pp. 742-762. https://doi.org/10.1109/TSE.2009.52

[26] A. Pachauri and G. Srivastava (2013). Automated test data generation for branch testing using genetic algorithm: An improved approach using branch ordering, memory and elitism. *Journal of Systems*

and Software, Elsevier, Vol. 86, No. 5, pp. 1191-1208. https://doi.org/10.1016/j.jss.2012.11.045

[27] S. Varshney and M. Mehrotra (2013). Search Based Software Test Data Generation for Structural Testing: A Perspective. *ACM SIGSOFT Software Engineering Notes*, Vol. 38, No. 4. https://doi.org/10.1145/2492248.2492277

[28] J. Wegener, A. Baresel, and H. Sthamer (2001). Evolutionary test environment for automatic structural testing. *Information and Software Technology*, Elsevier, Vol. 43, No. 14, pp. 841-854. https://doi.org/10.1016/S0950-5849(01)00190-2

[29] K. Liaskos, M. Roper, and M. Wood (2007). Investigating Data-Flow Coverage of Classes Using Evolutionary Algorithms. *Proceedings of the 9th annual conference on Genetic and Evolutionary Computation (GECCO'07)*, pp. 33-53. https://doi.org/10.1145/1276958.1277183

[30] M. Vivanti, A. Mis, A. Gorla, and G. Fraser (2013). Search-based Data-Flow Test Generation. *International Symposium on Software Reliability Engineering (ISSRE)*, IEEE, pp. 370-379. https://doi.org/10.1109/ISSRE.2013.6698890

[31] A. Ghiduk (2010). A New Software Data-Flow Testing Approach via Ant Colony Algorithms. *Universal Journal of Computer Science and Engineering Technology*, Vol. 1, No. 1, pp. 64-72.

[32] C. Mao, L. Xiao, X. Yu, and J. Chen (2015). Adapting ant colony optimization to generate test data for software structural testing. *Swarm and Evolutionary Computation*, Elsevier, Vol. 20, pp. 23-36. http://dx.doi.org/10.1016/j.swevo.2014.10.003

[33] C. Mao (2014). Harmony search-based test data generation for branch coverage in software structural testing. *Neural Computing and Applications*, Springer, Vol. 25, No. 1, pp. 199-216. https://doi.org/10.1007/s00521-013-1474-z

[34] A. Shamekhi (2013). An Improved Differential Evolution Optimization Algorithm. *International Journal of Research and Reviews in Applied Sciences*, Vol. 15, No. 2, pp. 132-145.

[35] H. Sharma, P. Shrivastava, and J.C. Bansal (2014). Fitness Based Self Adaptive Differential Evolution. *Nature Inspired Cooperative Strategies for Optimization (NICSO 2013)*, pp. 71-84.

[36] S. Das, A. Abraham, and A. Konar (2008). Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives. *Studies in Computational Intelligence*, Springer, Vol. 116, pp. 1-38. https://doi.org/10.1007/978-3-540-78297-1_1

[37] R. Malhotra and M. Khari (2013). Heuristic search-based approach for automated test data generation: a survey. *International Journal of Bio-Inspired Computation*, Inderscience, Vol. 5, No. 1. https://doi.org/10.1504/IJBIC.2013.053045

[38] A.P. Mathur (2008). *Foundations of Software Testing*. Pearson.

[39] T. Lengauer and R.E. Tarjan (1979). A fast algorithm for finding dominators in a flowgraph. *ACM Transactions on Programming Languages and Systems*, Vol. 1, No. 1, pp. 121-141. https://doi.org/10.1145/357062.357071

[40] S. Rapps and E.J. Weyuker (1985). Selecting software test data using data flow information. *IEEE Transactions on Software Engineering*, Vol. 11, No. 4, pp. 367-375. https://doi.org/10.1109/TSE.1985.232226

[41] J. Kennedy and R. C. Eberhart (1995). Particle Swam Optimization. *Proceedings of IEEE International Conference on Neural Networks (ICNN'95)*, IEEE, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[42] A. Nickabadi, M.M. Ebadzadeh, and R. Safabakhsh (2011). A novel particle swarm optimization algorithm with adaptive inertia weight. *Applied Soft Computing*, Elsevier, Vol. 11, No. 4, pp. 3658-3670. https://doi.org/10.1016/j.asoc.2011.01.037

[43] R. Storn and K.V. Price (1995). Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces. *International Computer Science Institute, USA*, TR-95-012.

[44] R. Ferguson and B. Korel (1996). The chaining approach for software test data generation. *ACM Transactions on Software Engineering and Methodology*, Vol. 5, No. 1, pp. 63-86. https://doi.org/10.1145/226155.226158

[45] SIR: Software-artifact Infrastructure Repository. http://sir.unl.edu/portal/index.php

[46] P.C. Jorgenson (2002). *Software Testing: A Craftsman's Approach*. 2nd ed., CRC Press.

[47] G.J. Myers (2006). The *Art of Software Testing*. 2nd ed., Wiley.

[48] K.K Aggarwal and Y. Singh (2007). *Software Engineering*. 3rd ed., New Age International Publishers.

[49] Garcia, A. Fernandez, J. Luengo, and F. Herrera (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, Elsevier, Vol. 180, pp. 2044-2064. https://doi.org/10.1016/j.ins.2009.12.010

[50] R. Feldt and A. Magazinius (2010). Validity threats in empirical software engineering research-an initial survey. *Proceedings of the 22nd International Conference on Software Engineering and Knowledge*

Qualitative and Quantitative Optimization for Dependability Analysis

Leila Boucerredj
University 08 mai1945, Guelma, Faculty of Science and Technology
Department of Electrotechnics and Automatics, BP.401, Guelma 24000, Algeria
Laboratory of Automatic and Signals Annaba (LASA)
Laboratory of Automatics and Informatics Guelma (LAIG)
E-mail: l_boucerredj@yahoo.fr

NasrEddine Debbache
University Badji-Mokhtar, Annaba, Algeria
Faculty of Science of the Engineer, Department of Electronics, Laboratory of Automatic and Signals Annaba (LASA)
E-mail: NDebbache@yahoo.fr

Keywords: system controlled by computer, dependability, optimization qualitative and quantitative, truth table, Karnaugh table, reduced Markov graph

Received: March 31, 2017

Systems that are not dependable and insecure may be rejected by their users. For many systems controlled by computer, the most important system property is the dependability of the system. For this reason in this paper, we propose a complete approach for dependability analysis. The proposed approach is based on optimization qualitative and quantitative for dependability analysis, qualitative optimization is based on causality relations between the events deduced from Truth Table Method combined with Karnaugh Table for deriving minimal feared states, quantitative optimization is based on Reduced Markov Graph this graph is directly composed by a minimal feared state deduced from the qualitative optimization, to avoid the problem of combinatorial explosion in the number of states in the Markov graph modelling. The representation of the Markov graph will be particularly interesting to study dependability.

Povzetek: Razvita je inovativna metoda za kvalitativno in kvantitativno optimizacijo analize odvisnosti programov.

1 Introduction

The migration from analogical to digital components in the systems controlled by computer has increased the complexity of the systems. In this modern system, dependability is the most important aspect of system quality, in order to guarantee their functional behaviour[1]. Most of the critical failures are generated by the interactions between the sub-systems, implemented in different technologies, which is based on the disciplines of mechanical engineering, electrical engineering and information technology....Therefore, the dependability analysis, one of the most important problems for modern systems typically intelligent systems such as system controlled by computers, becomes extremely difficult. The systems having no simple interconnections are called complex and hybrid systems [2], [3], [4].

The dependability of a system reflects the user's degree of trust in that system. Dependability covers the related systems attributes of reliability, availability and security. These are all inter-dependent [3], [5], [6]. Undependable systems may cause information loss with a high consequent recovery cost [5]. The costs of system failure may be very high if the failure leads to economic losses or physical damage.

In the most current papers, the evaluation of dependability methods (evaluation of reliability and availability) are generally reserved for the simple systems (series and parallel systems) or for the components [1], [2], [5]. The dependability analysis is conventionally modeled and analyzed using techniques such as Fault Tree Analysis (FTA) and Reliability Block Diagrams (RBD), consider for example the method of reliability block diagram, is primarily directed towards success analysis and does not deal effectively with complex repair and maintenance strategies or general availability analysis, is in general limited to non-repairable systems. The analysis is limited to single failures and is time-consuming [7].

Fault tree analysis is one of the most important logic and probabilistic techniques used in system reliability and safety assessment [2], [3]. FTA can be simply described as an analytical technique, whereby an undesired state of the system is specified (usually a state that is critical from a safety or reliability standpoint), and the system is then analyzed in the context of its environment and operation to find all realistic ways in which the undesired event (top event) can occur. FTA has limitations with respect to reconfiguration or state-

dependent behaviour of systems [7], so we can not represent reconfiguration [8], [9], [10]. Finally, it is not possible to take into account transient failures [4], [5], [6], [7].

The discrete events methods analyses (automats, Petri net) have their contribution in this field but the use of accessibility graph is quickly confronted to the problem of combinative explosion [4]. But, for the most part of *system controlled by computers*, their components are configured, where the interactions between the components are defined by logical or physical links which complicate the evaluation of the dependability of these kinds of systems.

System controlled by computer performance degradation is a stochastic process, Hence the need to use more appropriate methods for modelling and analysis of modern dynamic systems models such states transitions [6], [8], [9], [10], [11], [12]. These models include state graphs (e.g. Markov graphs).

Markov graph models (MGM) have been used to analyze computer networks [13], [14] and Programmable Electronic Systems (PES) used in industry to protect and control processes [14], [15].

Markov graph model represents the system in terms of the system states and transitions between the states, the representation will be particularly interesting to study dependability, the designer has the ability to view all of the operating modes (nominal and degraded) and the feared states of the system studied, and all failure rates (transitions) components, thereby improving the overall understanding of the behavior and evolution of the system in the presence of failures.

The aim of this work is to propose a dependability evaluation of system controlled by computer using a new approach based on optimization qualitative and quantitative analysis. The qualitative analysis optimization based on *Truth Table method* combined with *Karnaugh Table* used for focus the search of failure on the system study (or parts of the system) that are interesting for dependability analysis, the objective is to determine the causality events between nominal states, degraded state and feared state for deriving Minimal Feared State (MFS). Then we complement our study by a quantitative analysis optimization based on the construction of the Reduced Markov Graph (RMG), this graph is directly constructed by a set of minimal feared state deduced from the results of qualitative analysis. The advantage of Markov graphs lies in their ability to take into account the dependencies between components and the possibility to obtain various measurements from the same database modelling (Reliability, Availability, security...).

Despite their conceptual simplicity and their ability to overcome some shortcomings of the conventional methods of dependability, Markov graph is quickly confronted to the problem of combinative explosion in the number of states if the system is complex [5], [15], [16], [17], because the modelling process involves the enumeration of all possible states and all transitions between these states. To avoid the problem of combinative explosion of the number of states in the Markov graph modelling, it is possible under certain assumptions (Markov assumption) modelling with Truth table (TT), combined with Karnaugh Table (KT), for deriving minimal feared state (qualitative optimization) and subsequently generates the Reduced Markov Graph (quantitative optimization), which greatly facilitates the modelling because it is more structured and more compact. As the information associated with changes of state is stochastic (transition rates), this approach is well suited to describe the failure.

The paper is organized as follows: Section 2 present the quantitative analysis by Markov graph model. Section 3 contains detailed description of proposed approach optimization. We use a case study to illustrate the effectiveness of our approach, the results and summary steps of our proposed approach are provided in Section 4. Section 5 concludes the paper.

2 Quantitative analysis by Markov Graph

This method permits the calculation of reliability or availability of a repairable system or no with failure rates to the constant values [15], [16], [17]. It gives a representation of the causes of failures and their combination that lead to a feared situation.

A Markov process can be represented graphically by a state-of-transition model called a Markov graph. It is an oriented graph composed by a vertices and oriented arcs (lines),

- the vertices represent the states of the process,
- the oriented arcs (lines) connecting the evolution of states. This arcs are labeled by a transition rate T_{ij} (failure rate: λ) from the state S_i to the state S_j, or from the state S_j to the state S_i by a transition rate T_{ji} (repair rate: μ), so we can propose the folowing definition.

2.1 Definition (Markov Graph)

A Markov Graph (MG) is defined as a 5-tuple MG=(N, S, T_{ij}, λ, μ) where:

- N: is a finite number of states of MG,
- S = 2^N: corresponds to all possible states component of system (S_{ij}: nominal state, feared state, degraded state...),
- T_{ij}: continuous transition rates (failure, repair…) from S_i state to S_j state.

λ, μ *:* are the backward and forward transition rates, respectively:

if an arc (line) leads from the operating state **i** to the failure state **j** ($S_i \xrightarrow{T_{ij}} S_j$) is characterized by a constant failure rate λ_{ij} [1/time unit].

if an arc (line) leads from the failure state **j** to the operating state **i** ($S_i \xleftarrow{T_{ji}} S_j$) is characterized by a constant repair rate μ_{ji} [1/time unit].

A Markov graph state (S), is represented by a circle or oval, corresponds to a global system components state, the components may move from the failed state to the working state (S_j to S_i) as well as moving from the

working state to the failed state (S_i to S_j). These possible transitions (λ_{ij}, μ_{ji}) are represented by the transition lines (or arc model) and arrows in the Markov Graph from one to the other states (see Figure 1):

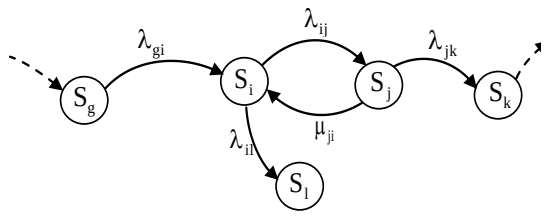


Figure 1: Markov graph representation.

The Markov Graph above (Figure 1) may be translated into a set of linear differential equations which represent the time-dependent behaviour of the state probabilities. These equations are given below.

An *n* state of Markov model leads to a system of *n* coupled differential equations. Let *P(t)* be a vector that gives the probability of being in each state at time *t*. the system of differential equation describing the Markov model is given by:

$$\left[\frac{dP_1(t)}{dt},\frac{dP_2(t)}{dt},...,\frac{dP_n(t)}{dt}\right]=[A]\times[P_1(t),P_2(t),...,P_n(t)] \quad (1)$$

Or:

$$\dot{\underline{P}}=[A]\times\underline{P} \quad (2)$$

Where $\dot{\underline{P}}$ and $\underline{P}$ are n × 1 column verctors, [*A*] is an n×n matrix (matrix of transition rates between states) and n is the number states in the system. The solution of equation 2 is given by equation 3:

$$\underline{P}=e^{At}\times\underline{P}(0) \quad (3)$$

Where e^{At} is an n×n matrix and $\underline{P}(0)$ is the initial probability vector describing the initial state of the system. It can be used for system state probability evaluation at the time t (transient analysis) or in the steady state t→∞ (stationary analysis) [16].

2.2 Example 1

In order to illustrate how the Markov model equations are developed, assume we have an example illustrated by figure 2.

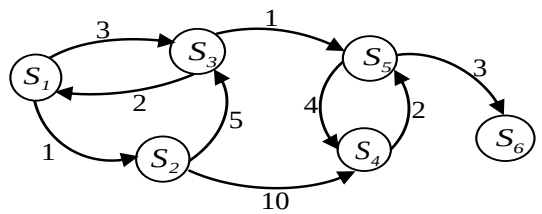


Figure 2: Markov Graph example.

The differential equation describing Markov Graph example (see Figure 2) is given by:

$$\left[\frac{dP_1(t)}{dt},\frac{dP_2(t)}{dt},\frac{dP_3(t)}{dt},\frac{dP_4(t)}{dt},\frac{dP_5(t)}{dt},\frac{dP_6(t)}{dt}\right]=[A]\times[P_1(t),P_2(t),P_3(t),P_4(t),P_5(t),P_6(t)] \quad (4)$$

$$A=\begin{bmatrix}-(1+3) & 1 & 3 & 0 & 0 & 0\\ 0 & -(5+10) & 5 & 10 & 0 & 0\\ 2 & 0 & -(2+1) & 0 & 1 & 0\\ 0 & 0 & 0 & -2 & 2 & 0\\ 0 & 0 & 0 & 4 & -(4+3) & 3\\ 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

$$=\begin{bmatrix}-4 & 1 & 3 & 0 & 0 & 0\\ 0 & -15 & 5 & 10 & 0 & 0\\ 2 & 0 & -3 & 0 & 1 & 0\\ 0 & 0 & 0 & -2 & 2 & 0\\ 0 & 0 & 0 & 4 & -7 & 3\\ 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}$$

The vector that gives the probability of being in each state at time *t* is (see equation 5):

$$\dot{\underline{P}}=[A]\times\underline{P} \quad (5)$$

[*A*] is defined as the state transition matrix.

The solution of equation 5 is:

$$\underline{P}=e^{At}\times\underline{P}(0)$$

If we have chosen the method of state representation of Markov processes to study the dependability of a modern complex system, it is necessary to use complex algorithms for calculate the parameters of dependability (reliability, maintainability...) [13], [16], [17].

In this paper we have choose the Markov Graph Model (MGM) to study the dependability systems. MGM represent the logical behaviour of components of the system study and should contain all possible states and transitions for the state components. In the context of dependability the representation of the Markov graph will be particularly interesting to visualize all the operating modes (nominal, degraded) and the failure state of the system and all failure and repair rates (transitions) of the components, which improves the overall understanding and evolution of the system in the presence of failures.

In the next part we have proposed an algorithm to construct Markov Graph Model (MGM).

2.3 An algorithm to construct MGM

Markov Graphs (MG) is the most frequently used type model for dependability analysis. It can be used to represent hardware, software and their combined interactions in a single model to provide various information. For example, a Markov graph can determine the probability of a system being in a particular state at a particular time and it can provide estimates for both safety and reliability [14].

The first step for building the Markov graphs is to identify the different states (working or failed) that the system can occupy (2^N states). The next step is to investigate how the system moves from one state to another state, by the various transitions between the states, these transitions represent the failure and the repair rates for the various components.

For construct the Markov Graph we have proposed the following algorithm.

Initialization:
Procedure Initialization:
Decompose the system into a component $\mathbf{C_i}$;
Define the number of state components of the system **N**;
Define Markov Graph Elements (MGE) MGE=(N, S_i, T_{ij}, λ, μ);
end procedure
Construction:
Procedure Construction:
for each Component C_i (i = 1 to N) **do**
create all states S_i *(2^N state)* of the system (working, degraded, failed);
draw all possible transitions (T) represented by the transition lines and arrows between states;
if the state of components are reparable
develop all transition failure rate and repair rate for each components;
draw the transition lines from operating state **i** to failure state **j** witch characterized by a constant failure rate $\boldsymbol{\lambda_{ij}}$ **then**
draw the transition lines from failure state **j** to the operating state **i** witch characterized by a constant repair rate $\boldsymbol{\mu_{ji}}$
else
draw the transition lines from operating state **i** to failure state **j** witch characterized by a constant failure rate $\boldsymbol{\lambda_{ij}}$
end
end
end for
end procedure

The major drawback of Markov graph models is that Markov diagrams for large systems are generally exceedingly large and complicated and difficult to construct. For example, the Markov graph associated to a system with **N** redundant components (each with two possible states: working and failed) can contain up to $\mathbf{2^N}$ states. For example, if we assume a system has **11** elements, each of which has two states (good and failed), the total number of possible states becomes $\mathbf{2^N = 2^{11} = 2048}$. We can see that the Markov Graphs for large systems are generally exceedingly large and complicated and difficult to construct [5], [6].

As the size of the Markov Graph increases if the systems are complex such as *intelligent systems*, we need use new approach to avoid the problem of combinatorial explosion in the number of states in the Markov graph modelling [16], [17], it is possible under certain assumptions (Markov assumption) modeling with Truth table method combined with Karnaugh table, to determine the minimal cut sets (Minimal feared state) and subsequently generates the Reduced Markov Graph (RMG), this permits simplifying the representation of MG and reducing the combinatorial explosion of the number states of the MG if the system is complex for quantitative optimization.

The proposed approach optimization for dependability analysis is developed in the next section.

3 Proposed approach optimization

3.1 Basic notation

In this section, we start with defining some basic elements of our proposed approach for dependability analysis.

3.1.1 Feared scenarios definition

A scenario can be defined as a beginning, an end and a history which describes the evolution of a system. In dependability and security study, a feared scenario leads to a catastrophic or dangerous state called feared state. The feared scenario describes how the system leaves from a nominal behavior towards the behavior in case of failure [4].

In this work the definition of a minimal feared scenario is based on the concept of 'Minimal Cut Sets'.

3.1.2 Minimal cut sets and minimal cut vectors

A cut set is a set of components of a system whose simultaneous failure leads into the failure of the system (if the system has been operational). A cut set is minimal, if no component can be removed from it without losing its status as a cut set [18], [19] [20]. A minimum cut sets is a section containing no other cut.

Every (minimal) cut set can be represented by a state vector. This state vector is known as (minimal) cut vector [13], [15], [18], [19] [20].

The Minimal Cut Set (MCS) size is a qualitative ranking of the causal combination, based on Boolean logic [17].

The qualitative analysis proceeds by 'Minimal Cut Sets' is used to optimize resources in assuring system safety.

From the results of qualitative analysis (MCS) we can calculate the occurrence probability of feared state using probability technique (F (t)) [6]:

- Mechanical and hydraulic components are characterized by a Weibull distribution, $F(t)=1-e^{-\left(\frac{t-\gamma}{\eta}\right)^{\beta}}$, where β is the shape parameter, η is the scale parameter and γ is the location parameter.

- Electronic and sensor components are defined by an Exponential distribution, $F(t)=1-e^{-\lambda t}$, where λ is the failure rate.

- Software components can be characterized by, $F(t)=1-e^{-\left(\frac{\eta}{\gamma+\eta}\lambda t\right)}$, where η is the solicitation rate, γ the execution rate and λ the failure rate. These parameters are evaluated by tests or simulations.

3.2 Truth Table method and Minimal Feared State

Based on the Boolean algebra, the Truth Table (TT) method allows identifying all the states (operations and failures) of the system based on binary behaviors [21].
The principle of this method consists of decomposing the system and identifying the failure modes of the different components, each component is characterized by an operating state (1) or by a failure state (0). It is a good tool to help understand the system functioning process and we can pick out the minimal feared scenario and expression for system reliability.

Establishing the TT of a system consists of analyzing the effects of all the vectors of the states components and determining all the malfunctions of the system. From this table, it is easy to deduce the failure combinations and failures leading to an undesirable event [21], [22], [23]. This optimizes system efficiency by minimizing the number of operations that must be performed to accomplish a given task.

Truth table is a picture of boxes 2^N, where N is the number of state components system. Each box represents a combination of state components of the system. From truth table we built the output function system state. It is possible to convert Truth table to the Karnaugh table which can also be directly translated into a Boolean function [24], helps us simplify Boolean expressions of system reliability, and to obtain the minimal feared scenarios (minimal cut sets) for constructing the reduced Markov graph, this allows to optimize the quantitative analysis and to optimize dependability system.

3.3 Karnaugh Table

Karnaugh Table (KT) is a Truth table graph, which aids for simplifying the output expressions of TT into a minimal number of literals form (Minimal Cut Sets).

– Karnaugh Tables are really only good for manual simplification of expressions.

– Compared to the algebraic method, the KT process is a more orderly process requiring fewer steps and always producing a minimum expression (Minimal Cut Sets).

– KT can take on values 1 or 0, in the context of dependability 1 represents the good states of system functioning and 0 represent the failure states. Therefore can be exploited to help simplification of expressions by grouping together adjacent cases containing ones, thus aids for generate the minimum number of feared states (Minimal Cut Sets (MCS) or Minimal Feared Scenario (MFS)) in the TT which will make the system to fail if their failure occurs. To illustrate the use of TT combined with KT to find the MFS we take the following example.

3.4 Example 2 to convert TT into KT for deriving MFS

In Truth table (TT) or Karnaugh Table (KT) anytime you have N components; you will have 2^N possible combinations and 2^N cases.

Consider a system having 4 components (a, b, c and d), at least two must work for the system to work.

If we list all combinations ($2^{N=4}$ = 16 combinations) of operational and failure states (in TT (Table 1.a) or KT (Table 1.b), (1) represents the operational state and (0) represents the failure state), we would have a table as illustrated in Table 1. The cases ''SF'' represent the State Functioning (SF) of the system.

Truth table of system example is shown first (Table 1.a), the converted TT into KT is shown behind (Table 1.b).

a	b	c	d	SF
1	1	1	1	**1**
0	1	1	1	**1**
1	0	1	1	**1**
0	0	1	1	**1**
1	1	0	1	**1**
0	1	0	1	**1**
1	0	0	1	**1**
0	0	0	1	0
1	1	1	0	**1**
0	1	1	0	**1**
1	0	1	0	**1**
0	0	1	0	0
1	1	0	0	**1**
0	1	0	0	0
1	0	0	0	0
0	0	0	0	0

Table 1.a: Truth Table.

cd \ ab	00	01	11	10
00	0	0	**1**	0
01	0	**1**	**1**	**1**
11	**1**	**1**	**1**	**1**
10	0	**1**	**1**	**1**

Not simplified with adjacent cases: represent the initial state of TT: All_OK (all components work).

Table 1.b: Karnaugh Table.

Table 1. Deriving MFS using TT and KT.

Karnaugh table representation (table 1.b) is equivalent to the Truth table (Table 1.a), that is to say that a line of TT corresponds to a square in the KT (see Table 1).

In system example (Table 1), we illustrate the use of KT (Table 1.b) for deriving the simplified output expression associate to the TT (Table 1.a). The principle of our proposed approach for deriving MFS from TT combined with KT; the case contains "All components work", not simplified with adjacent cases.

On inspecting Table 1 (a, b), the simplified expression (*reduced to fewer terms*) deduced from TT combined with KT (by *grouping* together *adjacent cases* containing *ones*), for system example is given by the following expression (see equation 7):

$$MFS = ab\bar{c} + \bar{a}b\bar{c}d + a\bar{b}c\bar{d} + \bar{a}cd + \bar{a}\bar{b}c + bc\bar{d} + abcd \quad (7)$$

Equation 7 represents the minimal cut sets or minimal feared scenario of system example.

If it is necessary to calculate the reliability system using MFS, we can use the probability technique form [6], for calculate the reliability system from its original form components, as shown in section 3.1.2.

So from the minimal feared scenario (equation 7), we can write down the expression of system reliability (equation 8) using probability of etch state of components:

$$R = P_a P_b (1-P_c) + (1-P_a) P_b (1-P_c) P_d + P_a (1-P_b) P_c (1-P_d) + (1-P_a) P_c P_d + (1-P_a)(1-P_b) P_c + P_b P_c (1-P_d) + P_a P_b P_c P_d \ . \quad (8)$$

The corresponding expression for calculate the occurrence probability of feared scenario (unreliability system *F* (t)) from equation 8, *is* given by equation 9:

$$F(t) = 1 - R(t) \quad (9)$$

In order to illustrate the use of TT combined with KT for construct Reduced Markov Graph (RMG) based on MFS let us consider a very simple example in the next section.

3.5 Construction of reduced Markov Graph

In this part of paper we explain the construction of RMG using TT method combined with KT.

3.5.1 Objectives

The objective of the qualitative optimization described previously is to point out the minimal feared states based on the causal events of TT combined with KT, for analyze with precisely the causal events what makes the system leave the normal behavior and goes to the feared state; starting from the initial states ‘’all components work (all_OK)’’ in TT (to begin the analyze) that contain the necessary information to make the qualitative analysis.

The main problem encountered when analyzing critical scenarios by exploring the all states (2^N) in the TT if the system is complex, anytime if you have N components, you will have 2^N possible combinations, and 2^N cases. In order to avoid the explosion combinatorial of states in TT we focus the search of the feared state on the part of the system that are interesting for dependability analysis, precisely is to make the Truth Table of the part of the system that leads to the feared state by exploring the all states that have a causal relation with the occurrence of the feared state, then we convert the TT to the KT for deriving MFS and then construct the Reduced Markov Graph (RMG).

The concept of the proposed approach it will:

Focus the search of feared state on the parts of the system (if the system is complex) that are interesting for dependability analysis,

Define the TT of the parts (or define the TT of the complete system if the system is not complex) of the system functioning that are interesting for dependability analysis, and establish the correspondence logical expressions of each state function,

Convert TT to the KT, and the case contain “All state working” in KT not simplified with adjacent cases containing ones, for deriving MFS.

The following example should clarify the proposed approach.

3.5.2 Illustration example 3

Suppose we have a system having 3 components a, b, c (n = 3) with two components to work for the system to work. The structure function of the system example 3 is defined in Table 2. Witch “1” represents operational state and “0” failure state.

list of combinations (operational and failure) state **($C_{i=3}$) = 2^3** =8				**State functioning (SF)** of system example 3
a	**b**	**c**	**C_i**	
1	1	1	abc	1
0	1	1	$\bar{a}bc$	1
1	0	1	$ab\bar{c}$	1
0	0	1	$\bar{a}\bar{b}c$	0
1	1	0	$ab\bar{c}$	1
0	1	0	$\bar{a}b\bar{c}$	0
1	0	0	$ab\bar{c}$	0
0	0	0	$\bar{a}\bar{b}\bar{c}$	0

Table 2: Truth Table of system example 3.

Table 2 represent the TT of system example 3, the system have **3** components, each of which have two states (good and failed); the total number of possible list of combinations states becomes **2^3 = 8**. This all states are presented in the TT illustrate by table 2, from this table a direct Markov Graph of system example 3 is represent in figure 3, corresponding at to all lists of combinations of operational and failure state of components (**($C_{i=3}$)=2^3 =8**).

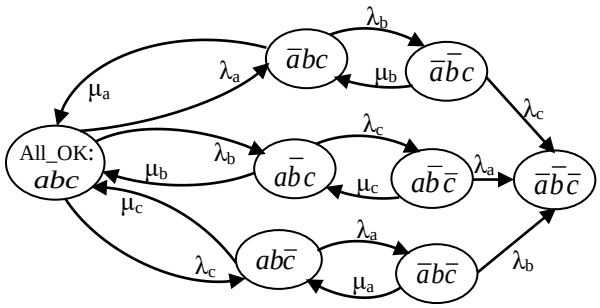


Figure 3: Converted TT to the Markov Graph.

MG of the system example 3 can be reduced using the MFS deduced from the KT as shown in Table 3.

c \ ab	00	01	11	10
0	0	1	1	0
1	0	1	1	1

All components are failed

Represent the initial state of TT: All_OK Not simplified.

$$MFS = \bar{a}bc + ab\bar{c} + a\bar{b}c + abc$$

Table 3: MFS of system example 3 using KT.

So for deriving Minimal Feared Scenario (MFS) using the KT (Table 3), the case represents all

components of system example 3 works, not simplified with adjacent cases containing ones. This case represents the initial state of TT and Markov graph model.

If it is possible to generate the Minimal Feared Scenario or Minimal Cut Vector (MCV) from TT of the system study, it is not necessary to make the Table of Karnaugh. What is necessary is that the Boolean expression should be reduced to its minimal form (MFS), and then draw the Reduced Markov Graph (RMG) from MFS or MCV.

Now, as we have seen in the table 2 and 3, the system example 3 has the following MFS (equation 10):

$$MFS = \bar{a}bc + a\bar{b}c + ab\bar{c} + abc \,. \quad (10)$$

These minimal cut sets (or MFS) can be represented by the following Minimal Cut Vectors (MCV): (0,1,1), (1,0,1), (1,1,0), (1,1,1) (see Table 2 and 3).

The above MG (Figure 3) can be reduced to the one show in figure 4, by using MFS or MCV as illustrated in table 2 and 3**,** respectively.

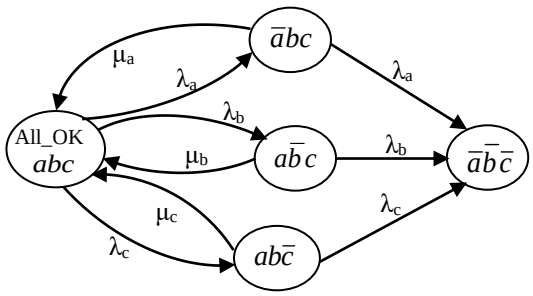


Figure 4: Reduced Markov Graph of system example 3.

In this section we see that the Markov Graph (MG) representation is very easy to construct if we use the TT method. If the system study is complex, we focus the search of feared state on the parts of the system that are interesting for dependability analysis, then create its TT combined with KT for construct the reduced Markov graph by using the concept of MFS or MCV associate to the TT combined with KT, which the case contain "All states working" in the KT not simplified with adjacent cases for deriving MFS as shown in table 3.

The Summary steps of our proposed approach are given in the next section.

4 Steps of our proposed approach

Now we need to enumerate the steps of our proposed approach of the dependability analysis as stated in section 3, the first step of our proposed approach is the: qualitative optimization.

Qualitative optimization steps: this step is based on the output simplified expression (*reduced to fewer terms*) deduced from the causality events of TT combined with KT in order to generate automatically the MFS for construct the RMG for quantitative optimization. The summary steps of qualitative optimization are:

Step 1. Define the number (N) of components (**C_i)** of the system study (***$C_{i = 1 \text{ to } N}$***).

Step 2. Start to build the Truth table of the system study,
If N the number of components you will have 2^N possible combinations

If the system is complex,

Make the TT of the parts of the system that are *interesting for dependability analysis* by identifying the all components of the part that are leads to the feared state (to guide and facilitate the search of feared state).

else

Identify all components of the system study for dependability analysis and develop the TT.

Step 3. In TT begin from the initial state ''All_Ok'' (correspond to all components of the system functioned correctly (if the system is complex: all components of the part of the system functioned correctly), then point out all possible combinations state of components (each components has two states ''working'' or ''failed'' therefore 2^N possible combinations). In the context of dependability put (1) for the good state (working state) and (0) for failure state and generate the state function of the system study.

Step 4. Convert TT to the KT and place 1s and 0s in the squares according to the Truth table.

Step 5. In KT circle groups of cases adjacent that contain 1, but the case represents the initial state All_Ok (all components of the system in good state), *not simplified* with *adjacent cases* containing *ones***.** Groups may be in sizes that are power of 2: 2^0= 1, 2^1=2, 2^2=4, 2^3=8... 2^N.

Step 6. From KT write the simplified output expression**,** by *grouping* together adjacent cases containing *ones*. The simplified output expression (*reduced to fewer terms*) represents the minimal cut sets or minimal feared scenario, this allows for modeling complex systems and to find the dependencies between failures, which are difficult to obtain with conventional dependability methods [10], [25].

Step 7. From the minimal feared scenario we construct the reduced Markov graph (RMG) for quantitative optimization. If it is possible to find the simplified output Boolean expression of the system study, from TT to its minimal form (minimal cut sets), it is not necessary to make the KT, then write the minimal feared scenario (MFS) or minimal cut vector (MCV) associate to the TT, and then construct directly the RMG to study the quantitative dependability optimisation.

Also from the simplified output expression, we can calculate the reliability system using probability propagation techniques [6] as shown in section 3.1.2.

Quantitative Optimization steps: from the results of qualitative optimization (MFS), reduced Markov graph are modelled (quantitative optimisation) based on minimal feared states, the states of RMG represented by circles connected by lines and arrows indicating possible transitions between the states. The transitions are conditioned, as appropriate, by process failure or repair entities down the intensity (failure rate or repair rate). This allows the representation of state dependent behaviour, including different information of components of the system and permits to obtain various

measurements from the same database modelling (Reliability, Probability of feared scenario, security...).

A case study in the next section is presented to illustrate the proposed approach.

4.1 Case study

In recent years, dependability and security is an important design priority in the development and advancement of modern technology and civilization. Figure 5 show the modern automatic control system case study used for controlling and maintaining a fluid at a desired level [V_{min} V_{max}] in a tank controlled by computer it is composed:

- Of a pump,
-Tree electrovalve EV1, EV2 and EV3, these electrovalves have only two operating positions fully open or fully closed.
-A tank controlled (according to order of the user Q_{out}).
-A tank of draining.
-A sensor of level which provides an analogical measurement of the level of fluid in the tank.
-A computer (CP) which decides, according to the value of the volume delivered by the sensor to supply (or not) the tank by feeding (or not) the electrovalve EV1.

The role of the computer is to simulate the volume (V) in the tank in real time, and giving the order of opening or closing to the tree electrovalves (EV1, EV2, and EV3).

The program that automatically the computer commands the tree electrovalves (EV1, EV2 and EV3) is:

```
if V ≤ Vmin
open EV1
if V ≥ Vmax
close EV1
  If EV1 blocked open and
 V > Vmax
   open EV2
   if EV2 blocked close
     open EV3
   end
  end
end
end
```

This system must avoid the overflow of the controlled tank. According to the received information from the sensor, if the volume in the controlled tank over crosses V_{max} ($V > V_{max}$) the computer actuates the electrovalves EV2 or EV3 of the system for draining the controlled tank; if the sensor identify that the volume in the controlled tank oversteps the upper limit V_{max} and if the EV2 (blocked close) is out of service (EV2_HS), the EV3 it can be used to drain the controlled tank in the tank of draining. If EV2_HS and EV3_HS, we consider the overflow of the controlled tank.

In this work we consider that only the electrovalves EV1, EV2, EV3 and computer (CP) can have failures (EV1_HS, EV2_HS, EV3_HS and CP_F (computer failed)) in the case of filing the controlled tank.

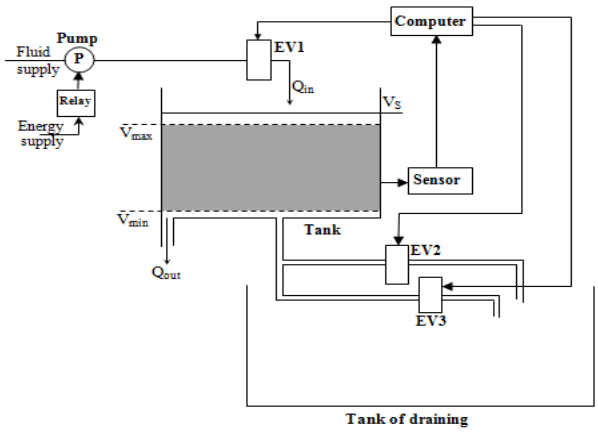


Figure 5: Case study.

4.2 Application of the proposed approach

By applying the method described in section 4, the first step is the qualitative analysis optimization for deriving MFS in order to identify the causal events leading to the overflow of the controlled tank.

4.2.1 Qualitative analysis optimization

The qualitative optimization is based on the simplified output expression (minimal cut sets) obtained from the Boolean reduction of TT method combined with KT as previously described in section 4. Our goal is to search the combinations of component failures causing system failure (overflow of the controlled thank).

For constructing the TT of case study we star with the state of all components (All_OK) in the good condition (EV1 EV2 EV3 CP) = (1111). Then we list all combinations of operational and failure state of tree electrovalves and computer ($2^{N=4}$ = 16 combinations), so we have the following table (Table 4).

EV1	EV2	EV3	CP	State Functioning (SF)
1	1	1	1	1
0	1	1	1	1
1	0	1	1	1
0	0	1	1	1
1	1	0	1	1
0	1	0	1	1
1	0	0	1	1
0	0	0	1	0
1	1	1	0	0
0	1	1	0	0
1	0	1	0	0
0	0	1	0	0
1	1	0	0	0
0	1	0	0	0
1	0	0	0	0
0	0	0	0	0

Table 4: Truth table of case study.

In Truth Table the state (EV1 EV2 EV3 CP) = (0000) represent the overflow of the system (SF = 0).

In this work the aim of the qualitative optimization is to determine the minimal cut sets (Minimal Feared State), by using the KT for generate the minimal number of feared state from TT. This is an efficient method to compute the Minimal Feared State of the system study based on the causality events of TT. So from the TT (Table 4) we construct the KT as shown in table 5.

EV3 CP \ EV1 EV2	00	01	11	10
00	0	0	0	0
01	0	1	1	1
11	1	1	1	1
10	0	0	0	0

This case represents the failure state of EV1, EV2, EV3 and CP (EV1_HS, EV2_HS, EV3_HS and CP_F).

This case represents the initial state All_OK: (EV1_OK, EV2_OK, EV3_OK and CP_OK) not simplified with adjacent cases.

Table 5: Karnaugh table of the case study.

From Karnaugh Table (see Table 5) we deduce the minimized Boolean expression form (see equation 11):

$$\begin{aligned} MFS = {} & EV1\_OK\ EV2\_OK\ EV3\_OK\ CP\_OK + \\ & EV1\_HS\ EV2\_OK\ EV3\_HS\ CP\_OK + \\ & EV1\_OK\ EV3\_HS\ CP\_OK + \\ & EV1\_OK\ EV2\_HS\ CP\_OK + \\ & EV1\_HS\ EV2\_OK\ CP\_OK + \\ & EV1\_HS\ EV2\_HS\ EV3\_OK\ CP\_OK . \end{aligned} \tag{11}$$

The minimal feared scenario (equation 11) deduced from KT is used not only in the qualitative optimization but in all the quantitative evaluations as well. The description of a scenario as given previously (in section 3.5) can be represented by Markov Graph, this allow drawing the reduced Markov Graph for quantitative optimization studied in the next section where the cercal are the events and the lines are the transition.

4.2.2 Quantitative analysis optimization

To study the dependability of the system controlled by computer (case study), it is important, first, to model it. Therefore, the first part of the methodology that we have proposed is the qualitative analysis optimization which will provide us with all the necessary information about the operation and the dysfunction of the system study and the causal events leading to the feared state.

Quantitative evaluations are most easily performed if the minimal feared state is obtained. The aim of this section is to complement our qualitative study by the quantitative analysis based on the construction of Markov Graph, which allows a limitation of the combinatorial explosion [13], [14], [16], [17]. This graph is directly constructed from the minimal feared states (Reduced Markov Graph) obtained from qualitative optimization. It is composed by a set of functional modes and a set of transitions to which statistical information regarding the system dynamics has been added.

This method permits the calculation of reliability or availability of a repairable system or no with failure rates to the constant values. It gives a representation of the causes of failures and their combinations that lead to the feared situation (overflow of the controlled tank), using us here the Software Reliability Workbench [26] for modelling the case study and for studies the quantitative optimization.

Reliability Workbench is Isographs flagship suite of reliability, safety and maintainability software.

So put the tree electrovalves and computer having a repair rates $\mu = 0.2\ h^{-1}$ and a failure rates are respectively: $\lambda = 0.02\ h^{-1}$ for the tree electrovalves EV1, EV2 and EV3; and $\lambda = 0.05\ h^{-1}$ for the computer (CP).

Consequently from the results of qualitative analysis (equation 11), by using the causality events of TT and KT, we directly built the Reduced Markov Graph (RMG) represented in software Reliability Workbench for quantitative analysis as shown in figure 6.

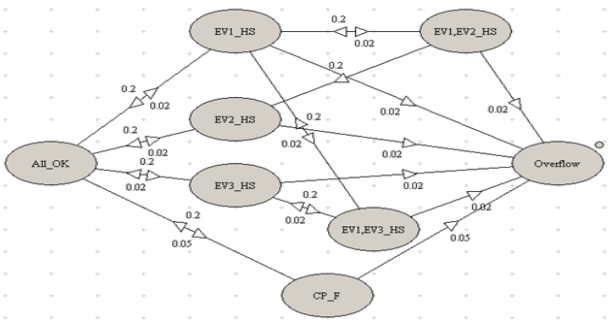


Figure 6: Reduced Markov Graph of case study.

The tops correspond to the states of the system. The lines describe the transitions between these states and a rate of transitions whose value is a constant theirs is associated. The Reduced Markov graph represented in figure (6) shows the event combinations leading to the feared states (Overflow). This graph includes the minimal failure sequences leading to the feared events.

The state **All OK**: all electro-valve and computer are in the good condition.
The **EV1_HS** state: represents the failure of EV1 (EV1_HS) and EV2 and CP in good condition.
The **EV2_HS** state: represents the failure of EV2 (EV2_HS) and EV1, CP in good condition.
The **EV3_HS** state: represents the failure of EV3 (EV3_HS) and EV1, CP in good condition.
The state "**EV1, EV2 HS**": represents the failure of EV1 and EV2, and EV3, CP in good condition.
The state "**EV1 and EV3 HS**" represents the failure of EV1 and EV3, and EV2, CP in good condition.
The state "**CP_F**" represents the failure of computer.
The state **Overflow** corresponds to the failures of EV1, EV2, EV3 and CP ((EV1, EV2, EV3, CP) = (0000)), this sequence represents the overflow of the controlled tank (system state = 0).

We have now defined the Reduced Markov Graph and can now proceed to perform an analysis.

A direct simulation in software Reliability Workbench, with 100 points and a lifetime of 450h, we obtain the following results:

Figure 7 shows the reliability of the controlled tank.

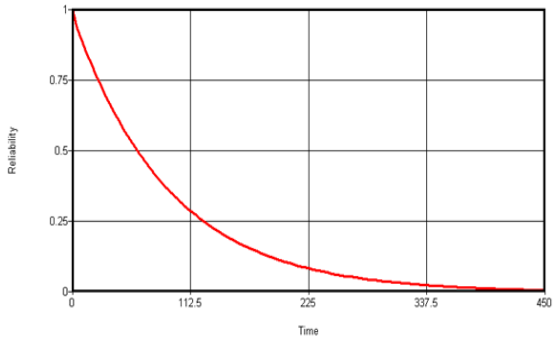


Figure 7: Reliability of the controlled tank.

A simulation shows that at time 200h the reliability of the controlled tank is: 0.11; at time 100h the reliability equal 0.33 and at time 50h the reliability equal 0.57. We can see that the reliability of the system depend on the failure states of components; it decreases rapidly as the number of failure components increases.

Figure 8, shows the Failure Frequency (FF) of overflow of the controlled tank.

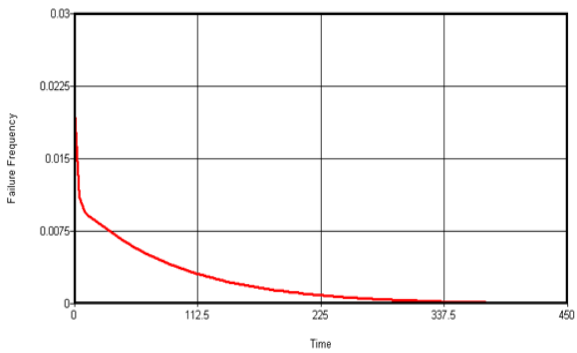


Figure 8: Failure frequency of the controlled tank.

Simulations show that at time 200h, FF of the system is: 0.0012; at time 100h the FF equal 0.0036; and at time 50h the FF equal 0.0062.

Figure 9 shows the evolution of Conditional Failure Intensity (CFI).

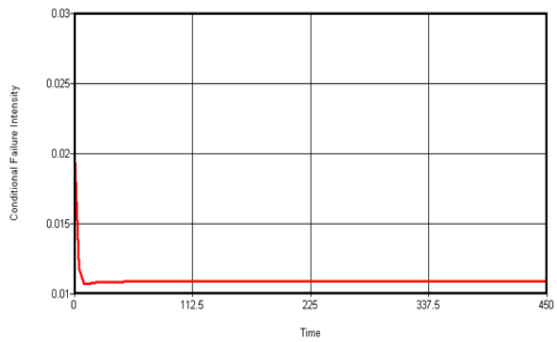


Figure 9: Conditional Failure intensity of the case study.

From figure 9, we can see that at the time instant t = 200h, 100h and 50h respectively the CFI of the system equal: 0.011.

Figure 10 shows the probability of overflow of the controlled tank.

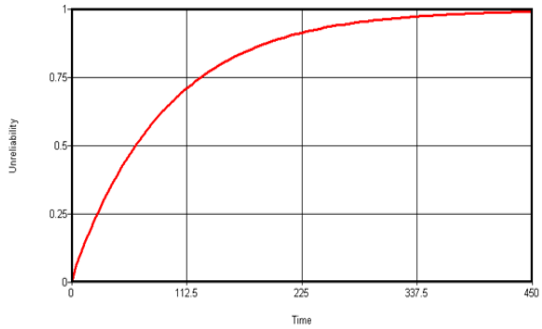


Figure 10: Probability of overflow of the controlled tank.

As Figure 10 shows, the probability of overflow of the controlled tank is: 0.89 at time 200h; 0.67 at time 100h and 0.43 at time 50h.

As confirmed by the results of the simulations we conclude that because the failure states of tree electrovalves (EV1, EV2 and EV3) and computer (CP), the probability of overflow of the controlled tank increases rapidly with time.

5 Conclusion

In this paper we have proposed a new approach for optimizing the qualitative and quantitative analysis used for dependability evaluation of modern intelligent systems such as systems controlled by computer. The first step of our proposed approach is the qualitative analysis optimization, for deriving minimal feared scenario based on causality events of Truth table combined with Karnaugh table. It is a good tool to help understand the system functioning process and we can pick out the minimal feared scenario.

Karnaugh Table process is more orderly process requiring fewer steps and always producing a minimum expression (minimal feared state) for dependability system. The combination of TT with KT presents two advantages. On the one hand, it allows a reduction of the feared state (minimal feared state), on the other hand, with the simplified output expression (reduced to fewer terms), we reduce the combinatorial explosion of the number of states of the Markov Graph (construct the RMG) for quantitative optimization. This allows for modeling complex systems, and to find the dependencies between failures. Reduced Markov graph permits the representation of state dependent behaviour, including different information of the nature of components (electronic, sensor, software,...) and system reparation. The quantitative evaluations are most easily performed if the minimal feared scenario is obtained.

The advantage of reduced Markov graph lies in their ability to take into account the dependencies between components and the possibility to obtain various measurements from the same database modelling (Reliability, Availability, Maintainability, Security...).

The simulation with Isograph Reliability Workbench verifies the effectiveness of our approach.

6 References

[1] Elena Dubrova, Fundamentals of Dependability, Chapter 2. Book, Fault-Tolerant Design, ISBN: 978-1-4614-2112-2. ©Springer 2013.XV, 185p. https://doi.org/10.1007/978-1-84996-414-2

[2] László Pokorádi. Failure Probability Analysis of Bridge Structure Systems. 10th Jubilee IEEE International Symposium on Applied Computational Intelligence and Informatics. Timişoara, Romania, May 21-23, 2015. https://doi.org/10.1109/SACI.2015.7208220

[3] Albert Myers, Complex System Reliability. Springer-Verlag, London, 2010. https://doi.org/10.1007/978-1-84996-414-2

[4] Hamid Demmou, Sarhane Khalfaoui, Edwige Guilhem, Robert Valette. Critical scenarios derivation methodology for mechatronic systems. Reliability engineering and system safety, 84 Elsevier. 33-44, 2004. https://doi.org/10.1016/j.ress.2003.11.007

[5] CS 410/510 - Software Engineering. System Dependability. Reference: Sommerville, Software Engineering, 10 ed., Chapter 10.

[6] Fabrice Guerin, Alexis Todoskoff, Mihaela Barreau, Jean-Yves Morel, Alin Mihalache, Dumon Bernard. Reliability analysis for complex industrial real-time systems: application on an antilock brake system. IEEE International Conference on Systems, Man and Cybernetics, Hammamet, October 6-9, 2002. https://doi.org/10.1109/ICSMC.2002.1175666

[7] Cristina Johansson. On System Safety and Reliability in Early Design Phases: Cost Fo cused Optimization Applied on Aircraft Systems. Linköping University Electronic Press, Sweden. Thesis, ISSN 0280-7971; 1600. 2013. p. 62 URN: urn:nbn:se:liu:diva-94354

[8] Pierre-Yves Piriou. Contribution to model Based Safety Analysis for dynamic repairable reconfigurable systems. Paris-Saclay University. Thesis presented at ENS Cachan, 27/11/2015. https://tel.archives-ouvertes.fr/tel-01251556

[9] Krishna B. Misra. Handbook of Performability Engineering. Book. Springer-Verlag London, 2008 https://doi.org/10.1007/978-1-84800-131-2

[10] Manno, Gabriele Antonino. Reliability modelling of complex systems: an adaptive transition system approach to match accuracy and efficiency. PhD Thesis, University of Catania, 2012. http://archivia.unict.it/bitstream/10761/1039/1/MNNGRL82L03C351S-PhD_Thesis_GM_A.pdf

[11] Norman B. Fuqua. The applicability of Markov analysis methods to Reliability, Maintainability, and Safety. Selected Topics in Assurance Related Technologies, Vol. 10, N. 2. Reliability Analysis Center, 2003. https://www.dsiac.org/sites/default/files/reference-documents/markov.pdf

[12] IEC 61165. Application of Markov techniques. International Electrotechnical Commission. 2006.

[13] Bateman. K. A., Cortes. E. R. Availability Modeling of FDDI Networks, Proceedings of Annual Reliability and Maintainability Symposium, IEEE. pp. 389-395, 1989. https://doi.org/10.1109/ARMS.1989.49632

[14] Kaufman. L.M., Johnson. B.W. Embedded Digital System Reliability and Safety Analyses. NUREG/GR-0020. University of Virginia. Department of Electrical Engineering Center for Safety-Critical Systems -Thornton Hall Charlottesville, VA 22904. xi, 75 p. 2001.

[15] Paraskevas Stavrianidis. Reliability and Uncertainty Analysis of Hardware Failures of a Programmable Electronic System. Reliability Engineering and System Safety, Elsevier, vol. 39, issue 3, pp. 309-324, 1993. https://doi.org/10.1016/0951-8320(93)90006-K

[16] Raphaël Schoenig. Definition of a design methodology for mechatronic systems including dependability analysis. PhD thesis of the National Polytechnic Institute of Lorraine, 2004. https://tel.archives-ouvertes.fr/tel-00126057

[17] Salem Derisavi, Peter Kemper, William H. Sanders. Lumping Matrix Diagram Representations of Markov Models. International Conference on Dependable Systems and Networks. Yokohama, Japan. IEEE, pp. 742–751, 2005. https://doi.org/10.1109/DSN.2005.59

[18] Way Kuo, Xiaoyan Zhu. Relations and generalizations of importance measures in reliability. IEEE Transactions on Reliability, Vol. 61, N. 3, pp. 659–674, 2012. https://doi.org/10.1109/TR.2012.2208302

[19] Sally Beeson, John D. Andrews. Importance measures for noncoherent-system analysis. IEEE Transactions on Reliability, Volume 52, issue: 3, pp. 301–310, 2003. https://doi.org/10.1109/TR.2003.816397

[20] Elena Zaitseva, Vitaly Levashenko, Jozef Kostolny, Miroslav Kvassay. Algorithms for Definition of Minimal Cut Sets in Reliability Evaluation of Green IT System. Department of Informatics, University of Zilina, Zilina, Slovakia. 2015. https://www.pdffiller.com/jsfiller-desk5/?projectId=226202130&expId=3950&expBranch=1#834b8f1bbf854c3e9f4c996e3b01e38a

[21] Alain Villemeur. Dependability of industrial systems. Collection of the Direction of Studies and Research of Electricity France, ISSN 0399-4198, Volume 67, 795 pages. Eyrolles, 1988.

[22] Pankaj Bansod. System Reliability and Challenges in Electronics Industry. SMTA Chapter Meeting 25th September 2013, India. https://pdfs.semanticscholar.org/presentation/64e3/b4774be3dad7f988fb5893a1a174e6cfabfa.pdf

[23] Popov Peter, Manno Gabriele. The effect of correlated failure rates on reliability of continuous time 1-out-of-2 software. International Conference on Computer Safety, Reliability, and Security,

SAFECOMP 2011. Lecture Notes in Computer Science, vol. 6894, Springer, Berlin, Heidelberg, pp. 1-14, 2011. https://doi.org/10.1007/978-3-642-24270-0_1

[24] Peter Cheung Professor. Lecture5: Logic Simplification & Karnaugh Map. Department of EEE. Lecture 5 - Imperial College London. 2007.

[25] Enrico Zio. Reliability engineering: Old problems and new challenges. Reliability Engineering & System Safety, Elsevier, Vol. 94(2), pp. 125–141, 2009. https://doi.org/10.1016/j.ress.2008.06.002

[26] https://www.isograph.com/software/reliability-workbench/2013.

https://doi.org/10.31449/inf.v42i3.1516 Informatica **42** (2018) 451–466 **451**

Bio-IR-M: A Multi-Paradigm Modelling for Bio-Inspired Multi-Agent Systems

Djamel Zeghida
LISCO Laboratory, Department of Computer Science, 20 Août 1955, Skikda University,
P.O. Box 26 Route El Hadaeik, Skikda, 21000, Algeria
dj.zeghida@gmail.com

Djamel Meslati and Nora Bounour
LISCO Laboratory, Department of Computer Science, Badji Mokhtar, Annaba University,
P.O. Box 12, Annaba, 23000, Algeria
meslati_djamel@yahoo.com, nora_bounour@yahoo.fr

Keywords: bio-inspired system, agent-oriented software engineering, influence/reaction principle, biomorphic system

Received: January 31, 2017

Nowadays bio-inspired approaches are widely used. Some of them became paradigms in many domains, such as Ant Colony Optimization (ACO) and Genetic Algorithms (GA). Despite the inherent challenges of surviving, in the natural world, biological organisms evolve, self-organize and self-repair with only local knowledge and without any centralized control. The analogy between biological systems and Multi-Agent Systems (MAS) is more than evident. In fact, every entity in real and natural systems is easily identified as an agent. Therefore, it will be more efficient to model them with agents. In a simulation context, MAS has been used to mimic behavioural, functional or structural features of biological systems. In a general context, bio-inspired systems are carried out with ad hoc design models or with a one target feature MAS model. Consequently, these works suffer from two weaknesses. The first is the use of dedicated models for restrictive purposes (such as academic projects). The second one is the lack of a design model.

In this paper, our contribution aims to propose a generic multi-paradigms model for bio-inspired systems. This model is agent-based and will integrate different bio-inspired paradigms with respect of their concepts. We investigate to which extent is it possible to preserve the main characteristics of both natural and artificial systems. Therefore, we introduce the influence/reaction principle to deal with these bio-inspired multi-agent systems.

Povzetek: Avtorji prispevka analizirajo podobnosti med biološkimi in multiagentnimi sistemi in predlagajo Bio-IR-M, integrirano shemo, ki zajema tako genetske algoritme kot npr. modele, temelječe na mravljah.

1 Introduction

In computer science, bio-inspired approaches are getting a particular interest. Their mechanisms and their behavioural, functional or structural features remain favourable fields of study and inspiration for multidisciplinary researches. Therefore, most researchers agree that both natural and bio-inspired systems are complex. In each system distribution and decentralization are inherent features.

We see now a large emergence of bio-inspired systems. These systems, inspired from nature and living organisms, extract metaphors for solving complex problems, getting new dimensions for systems we design. Some of these bio-inspired approaches became paradigms in many domains such as in hard optimization as heuristics [16], highlighting by the way the ACO meta-heuristic of *Dorigo* [15].

We can find early examples as use cases for instance in optimization with an evolutionary approach [23] or with a swarm intelligence (SI) using Ant Colony (AC) [10]. Other examples are presented for the use of Artificial Neural Network (ANN) in control and decision systems [35] or for object-class detection (specifically face detection) [53]. While [58, 59] present respectively the use of Artificial Immune Systems (AIS) and AC in the security domain. Although, [63] presents the use of AC intelligence with agent for scheduling and [4] illustrates routing with GA.

Some recent applications can be found, such as a parallel extended algorithm for the Ant Colony algorithm in [27] and Particle Swarm Optimization (PSO) algorithm in [37]. We can cite two others applications of the ACO Meta-heuristic for resource discovery in a grid using the agent technology [46] and for home automation networks [60]. A mobile agent Ant Algorithm (AA) has been used in an Ant-Based Cyber Defence system [21], when a hybrid Ant-Bee Algorithm was used for multi-robot coverage in [7]. For multi-objective optimization, we found the use of a Bat Algorithm (BA) in [2] and an

evolutionary algorithm in [49]. In vision, Artificial Neural Networks (ANN) are used for place recognition [11].

This proliferation is mostly due to technological and methodological advances in application areas and a better understanding of biological natural mechanisms.

Historically, the evolution of any approach or paradigm must be accompanied by a methodological evolution to carry the design side. Therefore, the need for an associated and specific bio-inspired modelling is becoming increasingly urgent. Such unified abstract representation will, at least, help overcome the lack of reuse in this domain.

A straight analogy can be easily identified between natural and multi-agent systems. Formally, we clearly distinguish two levels as follows:

- *Micro level*: held by the Agent concept in MAS and by an individual in natural system. An agent and an individual are both: autonomous, reactive, proactive and social.
- *Macro level*: referring to the MAS concept (an aggregate of interacting agents) and to a sub-system or to the entire natural system. In both systems we can find a set of features such as: diversity and distribution of knowledge, decentralization of data, distributed control, asynchronous calculations and processing, efficiency of parallel treatments, robustness, fault tolerance and dependability, flexibility, sophisticated plans of interaction (cooperation, coordination and negotiation), asynchronous local communication and emergent functionalities.

In this paper we focus on the modelling issue. We show the interest of a dedicated multi-paradigm model for bio-inspired multi-agent systems.

In fact, by exploiting the evident analogy between biological and multi-agent systems and highlighting the fact that these agent/multi-agent concepts are a common denominator for bio-inspired paradigms; it is quite natural to model these systems using autonomous agents. With regard to this perspective we suggest a unifying and generic influence/reaction agent model for several bio-inspired paradigms.

In Section 2, we give the background used in this paper that presents natural/multi-agent systems and the influence/reaction principle. Section 3 gives some reflections and analysis on Agent/Actor/Object concepts and the micro/macro levels in both MAS and bio-inspired paradigms, and then we show the convenience of using the agent concept as a generic model. All this help to position our contribution. Section 4 presents the concept of bio-inspired design. Throughout Section 5, we focus on the details of our proposed generic influence/reaction agent model, which is based on an explicit environment model and a separate interaction module. Section 6 presents some case studies. We discuss related works in Section 7, and Section 8 concludes the paper.

2 Background

This section provides the basic concepts and features of natural and multi-agent systems, highlighting the influence/reaction principle.

2.1 Specificities of natural systems

If we consider any ecosystem or biotope we can see that several autonomous species cohabit together with various complex interactions and interdependencies.

Biologists define the biotope as a small box with a separate set of environmental conditions (climatic and geological) that supports an ecological community composed of plants and animals. In a biotope, interdependence is complex and species survival depends on it. It is important to notice that all the biotope forms a coherent system and that various species cohabit while they differ greatly in terms of mechanisms and behaviours. These species require continuous changes of organization: decomposition/aggregation to face these very constraining and changing environments (Figure 1).

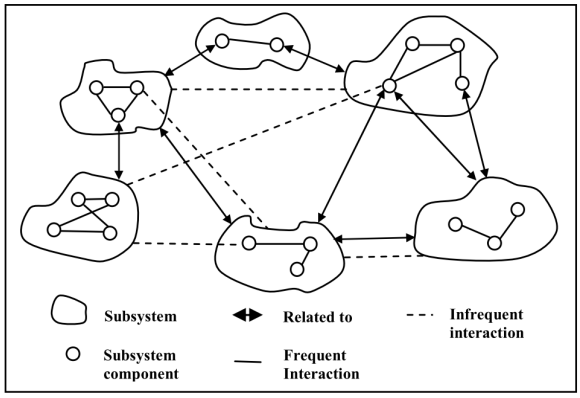


Figure 1: Canonical view of a complex natural system [30].

Note that distribution and complexity are innate features of these systems rather than casual. These systems are *Auto organized Group* of *individuals*. These last are *Autonomous*, *Simple* and *Cooperative*, put together in *local communication* to perform *Complex operations* in a *Distributed* and *Parallel* manner. Where the behaviour shown by the group is not *explicitly programmed* in the members but *emerges* from their *interactions*. These members *join* and *leave* freely the group in continual change. All this is performed without *any central control* [36]. With all this *chaos* and *anarchic interactions*, the organization continues to *grow*, to *live*, to *adapt* and *repair itself.*

2.2 Multi-agent systems

The multi-agent systems are based on the distribution of knowledge and control, spread over a set of entities called agents. MAS are a metaphor of social organization [9]. Agent technology comes from several fields: artificial intelligence, software engineering and human machine interfaces.

According to *J. Ferber* [19],"an agent is an autonomous entity, real or abstract, which can act on itself and its environment, which, in a multi-agent universe can communicate with other agents, and whose behaviour is a consequence of its observations, knowledge and interactions with other agents".

An agent is mainly [29, 62]:

- *Autonomous*: its behaviour is guided by objectives; it has an internal state on which it holds total control. This internal state is particularly inaccessible to other agents. Furthermore, the agent makes decisions that are based on this internal state without external intervention (human or other agent).
- *Reactive*: an agent is situated in an environment. It is able to perceive this environment and respond to events in it by its actions.
- *Social*: An agent is a social entity in the sense that it is able to interact and communicate with other agents through its environment.
- *Proactive*: an agent does not just react to its environment, but it is also able to produce self-actions motivated by its own goals (agent takes initiative).

An agent may be: reactive, cognitive or hybrid.

MAS based on reactive agents are characterized by a large number of simple agents, by emergence and eco-resolution. MAS based on cognitive agents are characterized by a small number of intelligent agents, by coordination, negotiation and cooperation. In this case, the system depends on the *agents' intelligence*.

When multi-agent systems are based on reactive agents (not intelligent), they depend on the *agents' interactions* to get intelligent collective behaviour. It defines a particular kind of Distributed Artificial Intelligence called Swarm Intelligence (SI). In such systems, intelligent functionalities (which haven't been explicitly coded in the system) can emerge throughout the agents' interactions.

MAS are usually characterized by:

- *Diversity and distribution of knowledge*: each agent has information and limited problem solving abilities (incomplete information and limited scope of action), and each agent has a partial view of the system,
- *Decentralization* of data,
- *Asynchronous* calculations and processing,
- *Distributed control*: there is no overall control of the system,
- *Efficiency* of treatments: the agents work in parallel and communicate asynchronously,
- *Robustness*, *fault tolerance* and *dependability*: the disconnections of some agents do not substantially affect the overall behaviour of the system,
- *Flexibility*: we can always increase (or decrease) the number of agents to treat larger and larger systems, without disturbing the work of existing agents who can adapt themselves,
- *Sophisticated plans* of *interaction*: they include cooperation, coordination and negotiation,
- Ideal for representing problems with multiple solution methods, multiple perspectives and/or multiple solvers. They have the traditional advantages of distributed and concurrent resolution of problems such as *modularity*, speed (with *parallelism*) and *reliability* (due to *redundancy*).

2.3 The influence/reaction principle

Besides being solution for simultaneity, the Influence/Reaction principle provides bases of good agent modelling/programming [41, 42] to accomplish more formally some aspects of the agent paradigm.

As a modelling principle, the Influence/Reaction principle has been defined for its ability to model concurrency behaviour but its interest goes beyond this objective. First, it gives a true semantic to the interactions management during the reaction phase (through influence). It, also, avoids the representation of action as a direct change in the global states of a system. This model can provide truly autonomous agents, requiring a clear distinction between the state variables of the agent decisional system (its mind) and variables relating to its physical appearance that are part of the environment (its body). The mind's variables are accessed/modified only by the agent and only during the Influence phase when the body's variables can be changed only during the Reaction phase by this environment [41, 42].

2.3.1 The influence/reaction principle for modelling simultaneous actions

Focusing on the autonomous nature of these entities, the simultaneity of action is an inherent characteristic of the agent paradigm which is, in addition, difficult to implement adequately. Constrained, agents must not have the control over the consequences of their actions, only the environment has the ability to compute them and for which the internal structure of an agent will stay unreachable. The influence/reaction principle is a solution for modelling simultaneous actions [17, 41, 42].

In two points, this principle is summarized in the fact that:

1. Agents do not have direct control over the result of their actions;
2. All the influences produced at a moment must be known to compute the new state of the world.

Every *application* of this principle will provide a *model* for its implementation.

2.3.2 The influence/reaction principle for modelling interactions

In Figure 2, let us denote $\delta(t)$ the dynamic state of a system at time t and γ_1, γ_2 two influences produced at this time.

The new state $\delta(t+dt)$ is given by the reaction function (equation1):

$$\delta(t+dt) = Reaction\ (\delta(t), \gamma_1, \gamma_2) \quad (1)$$

The parallel character of actions can be mandatory or optional depending on situations (see more details in [41, 42]).

In a mandatory parallel case, we have parallel reactions, requiring an explicit behaviour composition. To preserve the coherence of the system and to ensure the decisional autonomy of all involved agents, we calculate the reaction of the environment by treating all their influences simultaneously as a unit (equation 2):

$$\delta(t+dt) = Reaction\left(\delta(t), \bigcup_{i=1}^{2} \gamma_i\right) \quad (2)$$

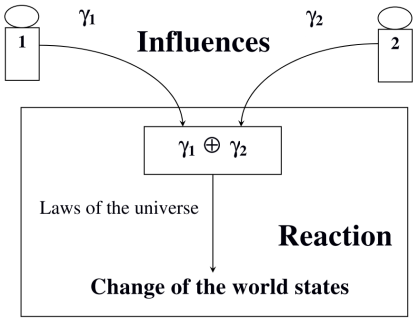


Figure 2: Illustration of the Influence/Reaction principle [41].

In the second case, the parallel character is no longer an obligation (it is just a modelling choice). Now we have serial (non-parallel) reactions. Both coherence of the system and the agent autonomy will not be compromised by the process used; we can use the equation 2 or we decompose the overall computing in elementary and independent reactions. We execute them in sequence one after another (equation 3 then equation 4). So we calculate first:

$$\delta' = Reaction\ (\delta(t), \gamma_1) \quad (3)$$

And then:

$$\delta(t+dt) = Reaction\ (\delta', \gamma_2) \quad (4)$$

(or γ_2 then γ_1)

We have to conclude here that the use of an Influence/Reaction model in the treatment of interactions calls for a *separate interactions module*.

3 Analysis and reflections

We notice that the use of the term *approach* refers to a vision or process to face or to deal with an issue, we can call it *paradigm* when it is well defined and widely used (for instance, agent/object are both paradigms in many domains, when we qualify them as approach, we mean the global vision and the way they proceed).

3.1 The challenge

Knowing the multitude and variety of bio-inspired paradigms available today (Table 1), it would be interesting to seek a unified approach for their design.

In Artificial intelligence, think *bio* is sometimes like to think multi-agent system, and think MAS is to think modelling and simulation. This transitivity of MAS is a natural bridge between the real world and the simulation and modelling in data processing. That is a generalisation of what was attested for immunology by *Bakhouya* [3]. So, for biology and MAS, the support is mutual.

Biology supports MAS in particular and the field of computer science in general, by providing artificial systems with principles, processes and mechanisms available in biological systems. This is achieved through biological metaphors as analogies established between the biological world and the artificial world, in order to propose approaches mimicking some aspects of the natural world while ignoring others. An historical overview of bio-inspired approaches can be found in [36]. Basically, the metaphors do not try to reproduce what is biological, but rather to interpret it in terms of what it is possible and reasonable to do. Thus, we can conclude that biological metaphors are evolving and depend on our understanding of reality and on our ability to extract beneficial and practical elements.

Paradigms	***Metaphor***	***Inspiration's Nature***
Artificial Neural Network (ANN)	Brain structure & functioning	Structural & Functional.
Genetic Algorithm (GA)	Genetic mechanisms	Functional.
Fuzzy System (FS)	Human reasoning	Functional.
Artificial Immune System (AIS)	Operating & organisational mechanisms of immune cells	Structural & Functional.
Ant Colony Optimization (ACO)	Ant colony behaviour	Behavioural.
Particle Swarm Optimization (PSO)	Swarm of bird in flight behaviour	Behavioural.

Table 1: Description of some bio-inspired paradigms.

On the other side, MAS allow the construction and design of complex systems highly distributed and adaptable to environmental changes. MAS offer to the biologists the ability to model and simulate, as simple as possible, complex natural systems (cells/molecules in interaction, insects, birds, fish or other living organisms) providing a reproduction of a natural phenomena through computers to:

- Understand their processes/mechanisms.
- Identify new metaphors: *computation / memorisation models* or *resolution / optimization tools.*

We have to notice that natural systems are by definition *Open Systems*, so must be artificial (bio-inspired) systems.

Beside their innate characteristics (Section 2.1), an *Open System* must have the three flowing characteristics:

1. The number of the system's components can change; the system *accepts new* components and *allows departure* of existing ones.
2. The system's organizational structure can change; there is no predefined and fixed organization to respect, components can *form* and *dissolve* aggregations and groups *freely*.
3. The two previous characteristics must be performed within "running" (in action) system.

The two first characteristics are enough in nature to define an *Open System*. The third characteristic can be ignored in living organisms and "organizations", because it is *naturally verified*: The ecosystem will not be constrained to *stop* or even *wait* the changes of its structure and the number of its components.

In artificial world (such in computer science), the third characteristic is very important. We can change the structure and the number of a system's components by modifying its *code* when it is *stopped*; in this case the system is not *Open*. To be *Open*, the two previous changes must be observed within a *running system* (system in execution).

Agent-Oriented Software Engineering (AOSE) has evolved to include the following high-level themes: *methodologies*, architectures, framework implementations, programming languages, and communication (Figure 3). Our contribution aims to address the *modelling issue* in the Agent Oriented Methodologies theme.

Figure 3: Agent-Oriented Software Engineering thematic map [55].

Mainly, a design methodology will include:

1. *Models*: Abstract representations of the real world or a part of it;
2. *Tools*: Means to represent, to manipulate and to implement the models;
3. *Process*: Coordinated set of steps, phases and tasks showing the path to achieve the system design.

For a precise positioning of our contribution, we summarize in Figure 4 and Figure 5, what has been already done in particular computer science fields and what remains to be done.

Figure 4 depicts the combined/separated use of bio-inspired approaches and Agent/multi-agent concepts in the field of Distributed Artificial Intelligence (DAI) or traditional Artificial Intelligence (AI).

The case (a1), illustrates the use of bio-inspired resolution/optimization tools (Algorithms: *computation / memorisation models* or *resolution / optimization tools*) to solve problems. All examples and applications cited in Section 1 belong to this case (except where it has been mentioned the use of agent).

The case (b1), illustrates the use of bio-inspired Agent/multi-agent modelling/simulation tools (Platforms) to model/simulate bio-inspired multi-agent systems. For instance, the use of *Turtlekit* tool in *Madkit* platform [26] for simulating artificial life/reactive systems and the use of *Repast* platform for simulating social science applications [22]. It can illustrate, too, the use of bio-inspired Agent/multi-agent tools and models (Algorithms) such in [7, 21, 46, 59, 63].

The case (c1), illustrates the use of Agent/multi-agent modelling/simulation tools (Platforms) to model and simulate multi-agent systems. *Gama*, *NetLogo* and *PRESAGE2* are examples of still used agent simulation platforms [22].

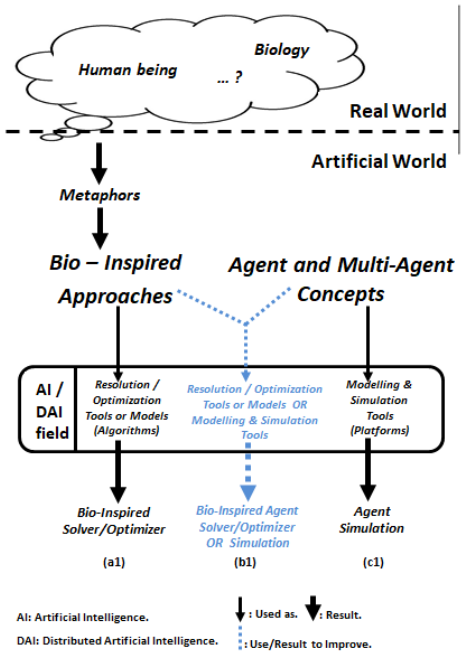


Figure 4: Bio-inspired approaches' use in AI/DAI field combined or not with MAS.

Figure 5 depicts the combined/separated use of bio-inspired approaches and Agent/multi-agent concepts in the field of Software Engineering (SE).

The case (a2), illustrates the use of bio-inspired Ad hoc methodologies (models/process/tools) to develop bio-inspired systems. It concerns most of developed bio-inspired systems.

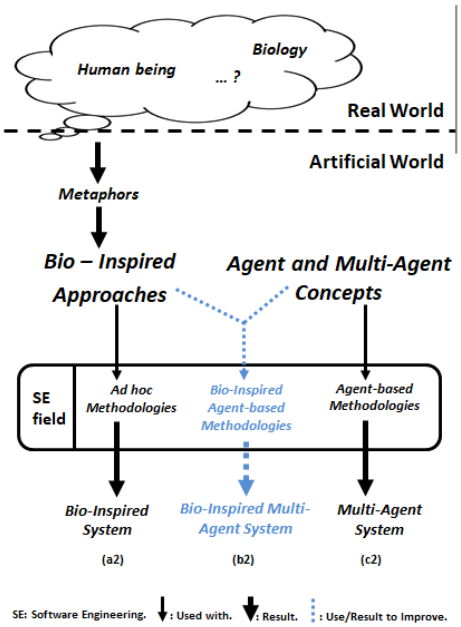


Figure 5: Bio-inspired approaches' use in SE field combined or not with MAS.

The case (b2), illustrates the use of bio-inspired agent/multi-agent methodology (models/process/tools) to develop bio-inspired multi-agent systems. There is no model nor methodology to deal with this case [24, 25, 34, 44, 50, 51]. Otherwise, we can find, only, methodology supporting a one target feature (for example *Adelfe* agent methodology supports *emergent functionalities*) [6].

It is the case that needs improvement, and where we aim to contribute in this paper.

The case (c2), illustrates the use of Agent/multi-agent methodology (models/process/tools) to develop multi-agent systems. For instance, we can cite the *AGR* and *AGRE* organisational models [18, 20].

For the methodologies we have, for instance: *Gaia*, *MaSE*, *O-MaSE*, *Passi*, *Prometheus*, *INGENIAS*, *Tropos* [22, 47, 56]. Some examples of their application can be found in [38, 40, 54], when others don't mention, at all, any methodologies [33, 52, 63].

3.2 Agent versus Object and Actor

As a modelling concept, to overcome the passive nature of *Object*, the less known concept of *Actor* was launched.

In Table 2, we situate the *Agent* with regard to the well-known and widely used concept of *Object*.

The *Actor* concept is a mathematical model of concurrent computation used for several practical implementations of distributed systems. It was built with a main added value; its asynchronous behaviour (Figure 6). The *Actor* concept initiated by 1973 was left out and ignored for decades. It has been relaunched first by *Gul* [1] and after that by *Karmani* and *Gul* [31, 32]. The *Agent* concept overtakes the *Actor* by its skills in interactivity (Figure 6). The three concepts became paradigms in computer science domain.

Comparison criteria	***Object approach***	***Agent approach***
Nature	Passive	Active and Autonomous
State/behaviour realization	Encapsulate	Encapsulate
Behaviour activation	Don't encapsulate	Encapsulate
Generic system functions	Focus	Neglect
Describing interaction' types	Primitive mechanisms	Advanced mechanisms
Patterns of interaction	Rigid and mandated	Flexible and sophisticated
Means of abstraction	Insufficient	Sufficient
Specifying and managing organizational relationships	Minimal support (static inheritance hierarchies)	Advanced support
Modelling complex systems	Not supported	Supported by concepts /mechanisms

Table 2: Comparing Object & Agent approaches.

If we take the most important and illustrative features; *Intelligence* and *intermediation*, Figure 6 depicts the places of the three paradigms *Agent*, *Actor* and *Object* together. This figure was inspired from a graphic description of *Agent* type and functionalities. It has been later refined in [66] and extended here to *Actor* and *Object*.

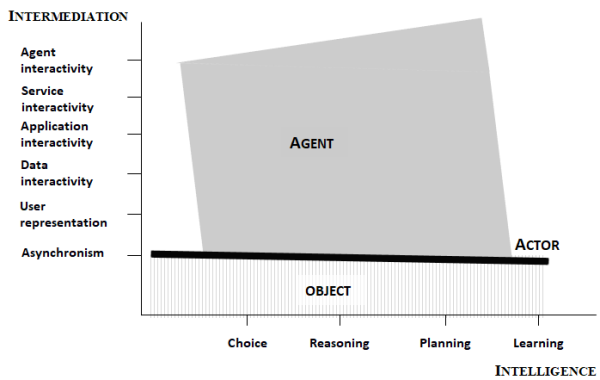


Figure 6: Positioning Agent, Actor and Object concepts according to the intelligence and intermediation features.

On the intelligence axis, both three paradigms can deal with this feature more or less easier. On the other axis, we distinguish an *inclusion* relationship (Figure 7). Indeed, *Objects* cannot even deal with the first step: *Asynchronism*, which is well-handled by *Actors*. *Agent* reaches farther steps, with its sophisticated means of *communication* preferably named *agent interaction*.

In agent interaction, we distinguish an *indirect mode*, used only for limited coordination (pheromones in ant

colonies) and a *direct mode*. The latest is widely used ranging from: agent language (*KQML* for Knowledge Query and Manipulation Language, *ACL-FIPA* for Agent Communication Language, proposed by the Foundation for Intelligent Physical Agents), ontologies and a communication support (present in agent platforms such as *JADE* [5] or *MadKit* [26]). The direct mode is structured using; protocols, dialogue games or argumentation systems [28].

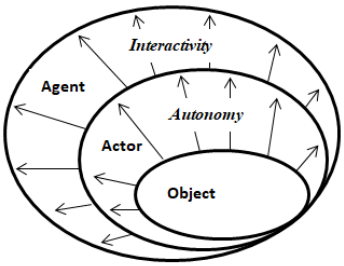


Figure 7: Object, Actor and Agent *inclusion*.

3.3 The analogy between biological systems and MAS

The first observation is the analogy between biological systems and MAS, and the mutual support of each. For instance, some bio-inspired approaches are easily identified to an aggregate of agents and have, so, a straight analogy with the MAS concept (the macro level). Others can be used in agent (the micro level) as a computational model held by the agent concept, as shown in Table 3.

Micro level (Agent)	*Macro level (MAS)*
- Artificial Neural Network (ANN) - Genetic Algorithm (GA) - Fuzzy System (FS)	- Artificial Immune System (AIS) - Ant Colony Optimization (ACO) - Particle Swarm Optimization (PSO)

Table 3: Classification of bio-inspired paradigms.

Note that for a particular use and specific abstraction need, we can use a micro level as a macro according to Table 4. For instance, with a functional metaphor (Table 4), GA was classed in the micro level (Table 3), but with deeper abstraction level it can be used in a macro level, where every genotype, for instance, will be hold by an agent.

Nature of the metaphor	*Micro level (Agent)*	*Macro level (MAS)*
Functional	Ok	
Structural		Ok
Behavioural		Ok

Table 4: Classification of bio-inspired paradigms according to their metaphor's nature.

3.4 The unifying formalism

The idea of using a unifying formalism to deal with the diversity of specific concepts to the considered paradigms became more obvious. Rather than proposing an approach that is the sum of the various concepts, or try to merge similar concepts, our vision of a unifying formalism is to wrap the various concepts by basic concepts and to operate, thereafter, a successive refinements that can be conducted in the specific contexts to each bio-inspired paradigm.

3.4.1 Adequacy of the agent approach for the development of natural systems

The multi-agent systems benefit from the effort of a wide scientific community relying on the fact that their approach adapts to various levels of abstraction. Indeed, from cognitive complex agents to very simple reactive agents, it is possible to model very different realities.

In [48], criteria that characterize bio-inspired MAS approach were proposed (Table 5). Some of these characteristics refer to the micro level, which is the individual component (agent level) and others to their aggregate (multi-agent level).

Criteria	*Nature*
Agents must correspond to entities and not to abstract functions.	Micro level: Agent
Agents should be small in size (system's parts), in time (able to forget) and in scope (avoid global knowledge/actions).	Micro level: Agent
Agents' community should be decentralized, with no single point of control or failure.	Macro level: MAS
Agents must be diverse. Randomness and repulsion are important tools for the maintenance and stabilization of this diversity.	Micro level: Agent
Agents' community should include mechanisms for disseminating information to increase its agents' reactivity.	Macro level: MAS
Agents must have means to capture and share what they know/learn.	Micro level: Agent
Agents plan and run in concurrent and parallel way.	Micro level: Agent

Table 5: Characteristics of bio-inspired multi-agent approach.

Many arguments have been given in favour of the use of agent-oriented approaches for the design of complex natural systems [30]. The role of engineering software is to provide the structures and techniques that facilitate the management of their complexity. It is in this perspective that software engineers have developed a number of fundamental tools in the field, referring to decomposition, abstraction and organization. Let us see the contributions of agent approach for each point [30].

1. *Advantage of agent-oriented decompositions:* Limiting the scope and extent of the designer, the decomposition is the basic technique that helps to counter big problems and their complexity, by dividing them into smaller parts, manageable and treatable in a relatively separated way. It is apparent that the natural way to model a complex system is based on several independent components that can act and interact in a flexible way to achieve their objectives. The agent-oriented approach seems to be the best choice.
2. *The convenience of agent-oriented abstractions*: Limiting, at a given time, interest and visual field of the designer, the process of defining a simplified model of the system, helps to overcome its complexity, by focusing on some details and ignoring others. In the case of complex systems composed of subsystems, components of subsystems and organizational relationships, it is natural to match the sub-systems to agent organizations, the components of subsystems to agents and interaction between subsystems and between their components will be viewed in terms of high-level social interactions.
3. *The need for flexible management of changing organizational relationships*: Offering the ability to specify and adopt organizational relation-ships, the process of defining and managing interactions between different components of problem solving (sub-systems and interaction links), helps designers to deal with complexity by allowing the grouping of components, to treat them as a unit of high-level analysis and to provide means for describing high-level relationships between various units. Agent-oriented systems have mechanisms for concurrent computing to form, maintain and dissolve organizations flexibly.

The multi-agent systems became a new *technology* for the *design* and control of complex, flexible and scalable systems.

3.4.2 The environment in bio-inspired multi-agent systems

In *AEIO* Vowels model [12]; *Da Silva* distinguishes four dimensions for MAS: *Agent*, *Environment*, *Interaction* and *Organization*. We notice that the *environment* component has been identified as a key element for MAS [61]. For bio-inspired systems, this component is of vital importance. This is the place where agents must co-exist and interact with the ability to form, maintain and dissolve organizations. All this changes can take place only through the environment [61].

Parunak [48] emphasizes a real consideration of the environment for "*natural*" MAS. In this context, he establishes that such system can be defined as three components:

$MAS = \{Agents, Environment, Coupling\}$

Where an $Agent_i$ is a set of four elements as follows:

$Agent_i = \{A.state_i, A.input_i, A.output_i, A.process_i\}$

The $Environment_i$ (as a scoop of $Agent_i$) is composed by two elements:

$Environment_i =< E.state_i, E.process_i >$

The exact nature of the *Coupling* depends on how we model agents and environment states and process. This *coupling* can be very complex. When agents and environment are discrete events, the *Coupling* of the $A.input_i$ and $A.output_i$ to $E.state_i$ is simply a mapping of agent and environment states. This kind of representations, dominating in the artificial intelligence area, is criticized because it generates unrealistic situations. A solution proposed for this is: the influence/reaction principle [17, 41, 42].

Obviously the autonomous of entities and simultaneity of their actions is crucial for natural MAS. So a direct validation of actions is to be avoided in such approaches. In respect of these requirements, we propose the use of the influence/reaction principle to deal with bio-inspired multi-agent systems.

4 Biomorphic systems

Nowadays we often speak about bio-inspired or biomorphic systems. Let us see their appropriate significations.

4.1 Origins

The *biomorphic* (biology-morphology) term was coined by the British zoologist *Desmond Morris* to describe the bio-inspired software approach [36].

Let us recall that a biological metaphor is an analogy sought to be determined between artificial and biological worlds, in order to provide tools which mimic some aspects of real world. The result of such process is a *bio-inspired system*.

A *biomorphic system* is simply designed based on algorithmic concepts inspired from biological systems and processes:

(*Biomorphic* = *Bio-inspiration* + *Design*).

Consequently when we speak about development, design or modelling, we precisely use the term bio-inspired instead of biomorphic which include implicitly a process and structure.

4.2 Premises of a bio-inspired design

The premises of any development process of biomorphic systems fall into two points:

4.2.1 Characterization of bio-inspired design

We had to identify the core processes and to formally describe their computational model. Since there are many paradigms, it is important to distinguish the basic paradigms and hybrid/composed ones.

Lodding [36] explains that a biomorphic system is the result of a bio-inspired design for a given system. It is designed based on concepts inspired from biological systems and processes. However, it is not easy to identify structural features for stating that a given architecture is bio-inspired.

To address this issue, several criteria have been identified to characterize the behaviour of biomorphic systems [36]. These criteria emphasize that a biomorphic system is materialized by a multitude of autonomous entities that collaborate. Table 6 depicts them and suggests their nature.

Criteria	***Nature***
The system behaviour results from the *collective interaction* of several independent and similar entities.	Macro level: MAS
The system behaviour *emerges* from the interaction of entities without being explicitly described in them.	Micro level: Agent
Entities *act autonomously*.	Macro level: MAS
The entities are operating based on *local information* and *interactions* and their spatial scope is rather local.	Micro level: Agent
The entities *appear* and *disappear freely* according to the system changes (free evolution of the group).	Micro level: Agent
The entities are able of *self-adapt* and adjust to changing objectives, knowledge and conditions.	Micro level: Agent
The entities have the ability to *evolve* over time.	Micro level: Agent

Table 6: Characteristics of a bio-inspired design.

As said for the *Parunak*'s characteristics (Table 5) these characteristics can be classified in two categories; atomic characteristics; referring to individuals and composed one, referring to a group of individuals (their aggregate).

4.2.2 Characterization of the context of applicability

The context of applicability, of each basic bio-inspired paradigm, help to reach a state where knowing specific criteria on a given problem, it will be possible to choose the bio-inspired paradigm to apply or indicate possible combinations (that suggests a multi-paradigm approach).

4.3 Consequences of a bio-inspired design

Based on the previous two premises, when we are interested in some way by a bio-inspired multi-paradigm development approach, it should be noticed that biomorphic aspect concerns the whole life cycle of a software system.

On requirements phase which is supposed to deliver the system functional and non-functional requirements, a preliminary determination of bio-inspired paradigm to use for each requirement or group of requirements is necessary. At this level we can, for example, determine that a particular requirement has characteristics that suggest the use of ant colony optimization or using a neural network classification. Determining the appropriate bio-inspired paradigm for a given requirement is closely linked to the premises previously introduced.

The design phase is a key phase. In architectural design, this phase allows to decompose the system into subsystems and to determine the role played by each one and interactions that must exist between the subsystems. For this, we must first determine the main bio-inspired paradigm to use according to the main system requirements.

Based on these requirements, it is possible that none of the basic bio-inspired paradigm matches and, at that time, it would be advisable to consider combinations (hybridization). The second step in design is the detailed design. If a subsystem must comply with a bio-inspired paradigm given its detailed design should specify inputs and outputs and the necessary adjustments to implement this paradigm.

4.4 The need for a multi-paradigm approach

Natural systems are by definition typically complex. This complexity is not only due to the multitude of entities that form their operational system, but also to the diverse nature of these entities and the varied interactions they may have.

It is sufficient for realizing it to consider an operating system and the various devices it manages, an Intranet and nested protocols which keep it operational, or an air or rail traffic management system.

4.4.1 Analogy with artificial systems

From an organizational point of view and having in mind the image of a biotope, an artificial system may be composed of interdependent subsystems where each is governed by a biological metaphor, provided by a given paradigm.

The underlying interest in this approach is to take advantage of the best paradigms for each problem. So, it is a synergy of the various paradigms that we want to achieve.

In turn, the subsystems can be decomposed and everyone will operate within a given paradigm. The relationship itself between the various sub-systems may be governed by a different paradigm from those governing the subsystems.

By analogy with the biotope where the objective is to maintain equilibrium between individuals, species and environment, the objective which we assign to a multi-paradigm approach is to provide a system with performance relatively best and good quality (reliability, development facility, maintainability, portability, etc.).

4.4.2 Rules of application of a multi-paradigm approach

This vision of complex systems raises remarks to be mentioned:

1. The multi-paradigm approach is simply a further bio-inspiration that makes the analogy between an artificial system and a biotope. It is not limited by a single metaphor but by many.
2. The multi-paradigm approach is a systemic approach that aims to integrate or hybridize the paradigms to take advantage of their synergy. For example, a system can be modelled as an ant's colony that uses genetic algorithms as a computational model.
3. In absolute, no paradigm dominates the other, but, a paradigm may be at the forefront in a context and second plane in another. For example, a system can be modelled as an evolving species (applying an evolutionary approach) where individuals are neural networks for which we try to improve the configuration or the synaptic weights. The opposite is also possible; for example a neural network where each node computes its combination function by a genetic algorithm.
4. Paradigms can be used in re-entrant order. For example, a neural network whose outputs are used to select another one among several neural networks.

Note to finish this section, that the persistence of various programming languages and their coexistence is a fact that illustrates the practical relevance of a multi-paradigm approach (Case of the .NET 'dotnet' platform of Microsoft, which is independent of any programming language and natively supports a large number).

The next section focuses on the modelling issue as part of a multi-paradigm bio-inspired approach.

5 The Bio-IR Modelling

In the context of a bio-inspired design, our goal is to use a generic model to unify the diversity of concepts specific to the considered bio-inspired paradigms.

A recapitulative reflection and analysis can be performed on what was presented in the previous sections. Indeed, besides the fact that MAS, like natural systems, consider that the systems are composed of interacting entities, there is a great similarity in the criteria for characterizing bio-inspired and MAS approaches (Table 5 and Table 6). It is possible to classify these characteristics into two categories: the intra-entity and inter-entities characteristics. In other words, we characterize the entities taken separately (atomic; referring to individuals) as we characterize their interactions (composed; referring to an aggregate of individuals). We notice that the same fact has been established for the classification of bio-inspired paradigms (Table 3).

For these reasons, we believe that the multi-agent systems approach is naturally placed as a prime candidate to act as a unifying modelling for biomorphic systems.

Figure 8 describes the meta-model of a general case of multi-paradigm bio-inspired multi-agent system with biomorphic agent and biomorphic group. We notice that it includes the six bio-inspired paradigms cited in this paper. For a new bio-inspired paradigm we have to classify it in micro/macro level. We must follow the Table 4' recommendations, according to the bio-inspired metaphor's nature, its particular use and the needed abstraction level. If it belongs to a macro level, we add it as a specialisation of the group (inheritance). Otherwise it will be added as a specialisation of the agent (being in the micro level).

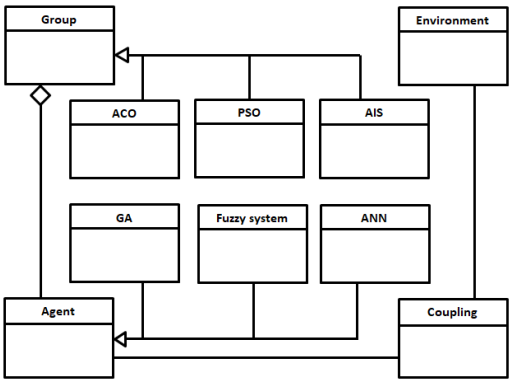


Figure 8: Meta-model for a multi-paradigm bio-inspired multi-agent system.

The complex nature of biomorphic systems is exhibited by different aspects ranging from simple computation, optimization, through complex coordination and symbolic resolutions. Using MAS to address these issues in a multi-paradigm context, we identify three possible scenarios:

1. *Intra-agent approach*: Where the agent encapsulates a processing according to a given bio-inspired paradigm (as a computational model for instance). The system is seen as an aggregation of biomorphic agents. This scenario has the advantage of encapsulating the diversity of paradigms in agents, which is interesting in terms of development: work division between teams (so it is the case of a modelling with only bio-inspired agents and without bio-inspired groups, (Figure 9));

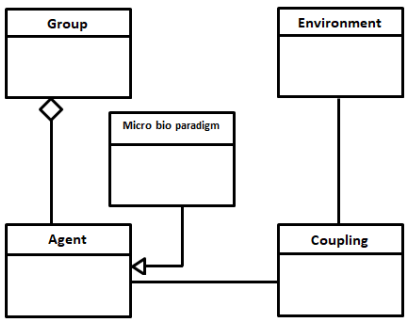


Figure 9: Meta-model for a bio-inspired agent.

2. *Inter-agents approach*: Where the bio-inspired aspect appears through the interactions of agents (i.e. MAS), we converge to a bio-inspired group

behaviour with non-bio-inspired agents (Figure 10);

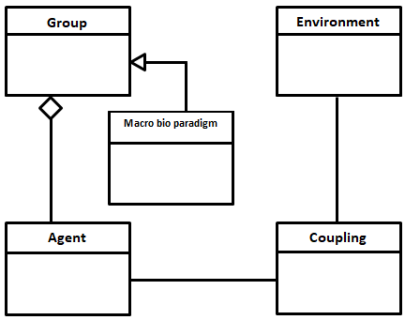


Figure 10: Meta-model for a bio-inspired group.

3. *Hybrid approach*: Where the previous two scenarios are combined. The system is then seen as a biomorphic group of biomorphic agents (the case of a modelling with bio-inspired groups and bio-inspired agents, (Figure 11)).

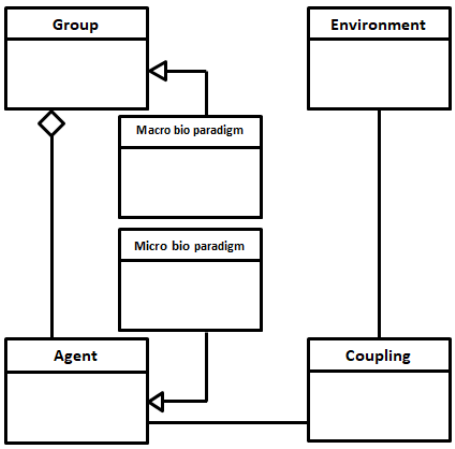


Figure 11: Meta-model for a bio-inspired agent and a bio-inspired group.

We notice that, in our model, there are no constraints on the type/architecture of the agent. In the *micro level*, the agent will be *cognitive* according to the bio-inspired approach that it holds. In this case, its *Computation module* must be, consequently, sophisticated. In a *macro level* the agent is *generally reactive*.

Formally and at a higher level of abstraction, in biomorphic MAS the three previous cases will be reflected in two levels as follows:

- *Agent level*
  We use an agent model which must support the biological dimension; it will be designed by ensuring real autonomy with the separation between the state variables of the decisional system (the mind) and the physical component (the body). These interacting agents can be structured in groups (Figure 12.a).
- *Group level*
  The resulting system is an aggregate of interacting agents. These interactions will be managed by a separate interaction module. We emphasize the active character of the environment to be modelled explicitly. This feature is because it has its own process that can change its state, regardless of the actions of its agents. The states of various agents are coupled to the state of the environment. This coupling will be performed using the influence/reaction principle.

We model a bio-inspired influence/reaction multi-agent system as follows (Figure 12):

Bio.IR.M = {{*Bio.IR.A*}, *Bio.IR.E*, *Bio.IR.C*},

Where:

1. *Bio-IR-A*; the Agent component: An agent does not have a direct control over the result of its influences on the environment, including on its physical component state variables. The agent has to emit influences to the interaction module. But in the opposite, the agent can use and modify its decisional system state variables, its physical component state variables can be changed by an external component (as a reaction to the environment component for instance) (Figure 12.a).

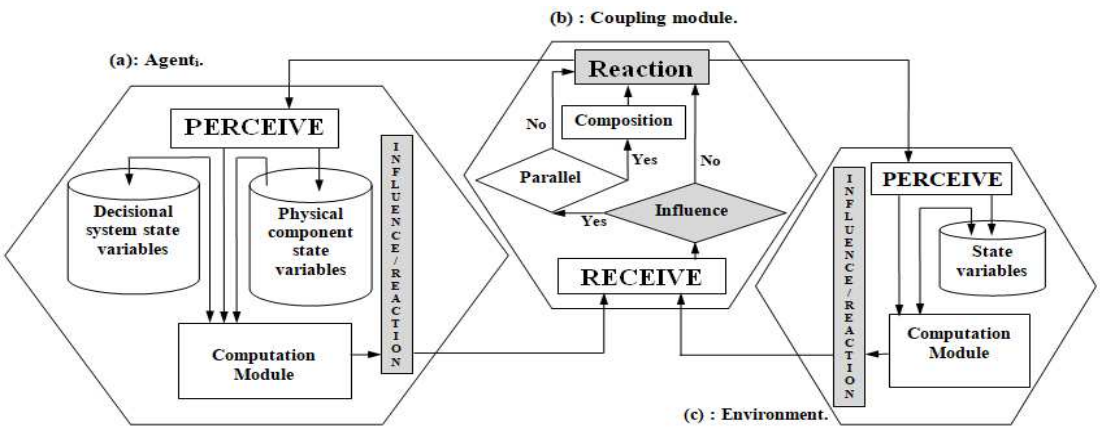


Figure 12: Bio-IR-M: The bio-inspired influence/reaction modelling; (a) Bio-IR-Agent, (b) Bio-IR-Coupling, (c) Bio-IR-Environment.

2. *Bio-IR-C*; the Coupling component: The coupling module manages interactions by composing the agent/environment influences which are simultaneous and then forward the result to the environment/agent component (Figure 12.b).
3. *Bio-IR-E*; the Environment component: as an active component, the environment re-acts (by its own influence) to the agents' influences based on its own process and state. The environment can not only use and modify its state variables but also change the agent physicapl component state variables through the coupling module (Figure 12.c). However, the environment cannot reach the agent decisional system variables.

The outgoing arrows from a database are read access, the incoming ones are updates.

This model can preserve the integrity of our agents by separating their state variables. Decisional system variables are accessed / modified only by the agent during the influence phase. The physical component variables are part of the environment and are modified only by the environment during the reaction phase.

The reaction of agent/environment is in our case an influence wished to be performed on the environment/agent and it is not, any more, a traditional action, in the artificial intelligence sense.

Even if the influence/reaction principle does not affect the simultaneous action and the interaction modelling, this principle improves the information dissemination mechanisms to increase the system's reactivity.

To this end, we summarize the main characteristics of our proposal in:

1. The application of influence/reaction principle.
   - Able to model concurrent and joined behaviours.
   - Abandon the representation of the action as a modification of the system' global state.
   - Improve mechanisms for disseminating information to increase agent reactivity.
2. Isolating an interaction module (the coupling module). Use all the influences produced at a moment to compute the new state of the world.
3. The guarantee of the agent integrity (autonomy) by the distinction between the decisional system state variables of an agent and variables concerning his physical aspect.
4. The explicit modelling of the environment.

6 Application case studies

We take, as a first case study, the use of *Ant Algorithm* (Ant Colony Optimization meta-heuristic) applied to the famous *Travelling Salesman Problem* (TSP).

Figure 13 illustrates the modelling of a TSP Ant System, according to our model and using an *adapted AGRE* organizational model [20]: a special consideration for the *environment* and a *double circle* for the bio-inspired aspect. In this case we have a macro level bio-inspiration represented with a biomorphic group "*Validation*", implementing the ACO approach to find the shortest circuit of towns.

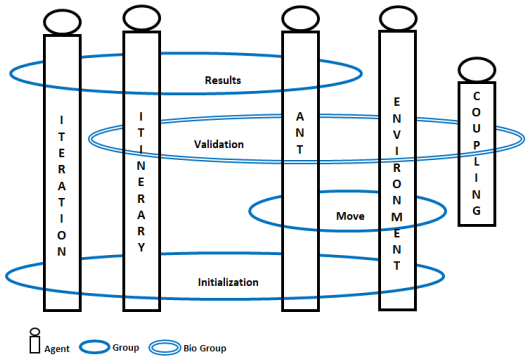


Figure 13: Bio-inspired influence/reaction TSP modelling with ACO bio group.

The agents ant in this implementation use the probability depending on distance and the pheromone density on every path between towns to choose the next town to move to (the corresponding Meta-model is given in Figure 14).

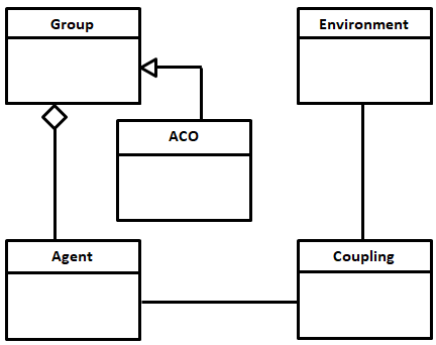


Figure 14: Meta-model for an ACO bio group.

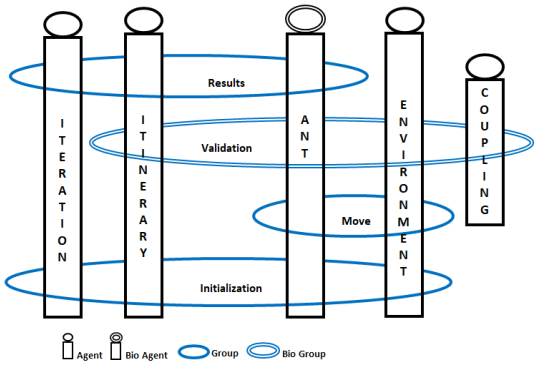


Figure 15: Bio-inspired influence/reaction TSP modelling with GA bio agent and ACO bio group.

Figure 15 shows the modelling of a TSP Ant System with a macro level bio-inspiration: a biomorphic group "*Validation*", implementing the ACO approach and a micro level bio-inspiration: a biomorphic agent "*Ant*", using, for instance, as computational model a Genetic

Algorithm to choose the next town to move to (its Meta-model is presented in figure 16).

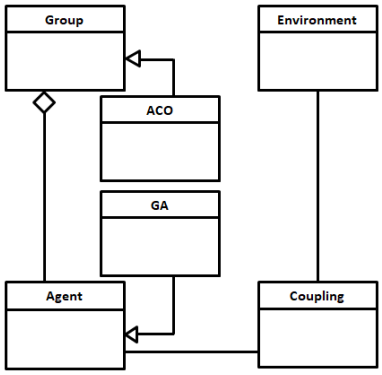


Figure 16: Meta-model for a GA bio agent and an ACO bio group.

In both cases the coupling is performed with the influence/reaction principle. The environment can be seen as a graph, where nodes are towns and arcs/weights are paths/distances between towns. An implementation on the *JADE* platform for the first case can be found in [67] comparing the three basic Ant System Variants: Ant-Cycle, Ant-Density and Ant-Quantity [13, 14]. The obtained results are promising in both SE and DAI fields (Figure 4 and Figure 5 in Section 3.1). That encourages us to look after improved variants of ant algorithms, such as the max-min ant system [57] and to explore other aspect using *JADE* and *Madkit* platforms to propose our improved Ant Algorithm.

A second case study concerns the *Time Tabling Problem* (TTP) solved with an *Ant Algorithm* too. Figure 13 and Figure 14 can illustrate, respectively the modelling of TTP Ant System and its meta-model. In this case the environment is a graph, where nodes are sessions' extremities (begins/ends); arcs and their weights are duration and classes/classrooms. Consequently, ants (teachers) perform following an adapted process.

Another case is to deal with TTP, using a *Grey Wolf Optimization* (GWO) [43]. In this case, we have, just, to replace Ant with Wolf (teacher) in Figure 13 and ACO with GWO in Figure 14 to illustrate, respectively the modelling of TTP Grey Wolf Optimization System and its meta-model.

7 Related Works

We can find various examples of bio-inspired multi-agent systems. Most works have a specific purpose and are suffering from the fact to be designed using an Ad hoc process and "*methodology*" or targeting one bio-inspired feature.

A first example and as a dedicated agent-based methodology, [6] presents the *ADELFE* methodology. *ADELFE* is devoted to the design of adaptive and cooperative multi-agent systems and relies on the AMAS theory "*Adaptive Multi-Agent Systems*". It seems to be a candidate for the handling of a class of biomorphic systems characterized by swarm intelligence.

A second example is taken from the engineering of self-organization in multi-agent systems. Inspired from multi-cellular organisms, *Nagpal* in [45] gives a set of bio-inspired primitives engineering in robotics.

In [8], author gives another example to build bio-inspired self-adapting systems; it deals with particular software systems, and presents the use of architectural styles in a software architectural perspective applied to problems with shared characteristics. It consists mainly to create a model for a given biological system. This model has to be studied until being completely understood. After that, in an iterative cycle, designers build on this initial model the target biological system. A concrete case was given for a discreet distribution problem: distributing a computation on a large network, where any small group of nodes ignore the problem they are helping to solve.

We can conclude that all existing works remain specific for particular domains and classes of problems and don't support and encourage reuse.

At variance, and with more general vision, useful guidelines to a better definition and characteristics of biomorphic MAS were given in [48, 61] encouraging an advanced bio-inspiration which can lead to a generic process according to our topic.

Another work suggests the extension of the *AGR* organizational model (Agent, Role and Group) [18], which gives rise to *AGRE* model [20]. *AGRE* includes the environmental dimension and crosses with our vision of the development issue of biomorphic multi-agent system.

In [64, 65] authors present a general multi-agent framework called *SAPERE* (Self-aware Pervasive Service Ecosystems). *SAPERE* deals with pervasive systems seen as an ecosystem where the pervasive computing services are carried with multi-agent systems. Their contribution aims to perform the interactions between these services (MASs) with respect of bio-inspired laws summed in: *Bound*, *Aggregate*, *Decay* and *Spread*.

In our case, we deal with natural systems with a multi paradigm modelling approach seen as a biotope or an ecosystem (system of interacting systems). These interacting systems implement a given bio-inspired paradigm and the interaction between them, itself, may be governed by a bio-inspired paradigm too. Our contribution aims to model these interacting systems and their interaction with multi agent systems with respect of the Influence/Reaction Principle. So, we, both, use some common concepts and terminologies but in different levels: They tackle, with a bio-inspired approach, the interaction issue between an ecosystem's systems assumed multi agent, when we tackle, with a multi agent approach (using the Influence/Reaction Principle to manage agent's interaction), the modelling issue of an ecosystem's systems and their interaction assumed, both, bio-inspired. Their work can be seen as an ideal general case study of our work, if their pervasive computing

services were *all bio-inspired* with an *influence/reaction's interaction* model.

In [39], authors allow agents, in MAS technologies, to adopt dynamically an interaction's mean among different possible ones. Concretely, they used the *TuCSoN* (Tuple Center Spread over the Network) dedicated agent platform within the *JADE* and *Jason* platforms. *TuCSoN* use a logic-based coordination language (*ReSpecT*), it is a Java library to model coordination in distributed processes (such as autonomous, *intelligent* and *mobile* agents).

The idea is interesting and can be used with our multi-paradigm vision to integrate different bio-inspired paradigms. When the bio-inspired paradigm is hold at a micro level by agents (they must be intelligent) or in a macro level by MAS based on small number of intelligent agents, the idea is worthwhile. But, when the bio-inspired paradigm is hold at a macro level by MAS based on big number of simple (not intelligent) agent (as indicated with MAS presentation in Section 2.2 and noticed in Section 5) the idea will be less useful.

8 Conclusion

To deal with the proliferation of biomorphic systems it has become necessary to focus attention and research efforts on their modelling. Such modelling must encompass all the different bio-inspired concepts.

In this paper, we have advocated for a generic influence/reaction agent-based model which integrates various bio-inspired paradigms. We consider this work as a step towards a development methodology for biomorphic MAS. Based on the fact that MAS represent a potentially unifying paradigm, a first perspective is to establish a synthesis of agent-based methodologies and identify a kernel to adapt, in order to incorporate a meta-model based on our generic bio-inspired model. The degree of adaptation of a development approach, to this objective, depends not only on the diversity of the considered bio-inspired approaches but also their possible combinations, enriching their existing scope of applicability.

In such multi-paradigm context, a second perspective would be to reconsider this kernel to exploit the power of bio-inspired approaches. Where for a given problem and knowing all its specific criteria, we will be able to reach the state for a real guidance of the user to choose the bio-inspired paradigm to apply or indicate possible combinations.

References

[1] Agha, G. A., (1985). Actors: A model of concurrent computation in distributed systems, *technical report*, DTIC Document.

[2] Amine, L. M. and Nadjet, K., (2015). A Multi-objective Binary Bat Algorithm, *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, ACM, 75.

[3] Bakhouya, M., Gaber, J. and Koukam, A., (2003). Bio-inspired model for behavior emergence: Modelling and case study, *Procs. Of Knowledge Grid and Grid Intelligence workshop* (KGGI'03) at IEEE WI/IAT'03.

[4] Bari, A., Wazed, S., Jaekel, A. and Bandyopadhyay, S., (2009). A genetic algorithm based approach for energy efficient routing in two- tiered sensor networks, *Ad Hoc Networks Journal*, Elsevier, (7), (4), 665–676.

[5] Bellifemine, F., Poggi, A. and Rimassa, G., (1999). JADE–A FIPA- compliant agent framework, *Proceedings of PAAM*, London, (99), 97–108.

[6] Bernon, C., Gleizes, M-P., Peyruqueou, S. and Picard, G., (2002). ADELFE: a methodology for adaptive multi-agent systems engineering, *International Workshop on Engineering Societies in the Agents World*, Springer, 156–169.

[7] Broecker, B., Caliskanelli, I., Tuyls, K., Sklar, E. and Hennes, D., (2015). Social insect-inspired multi-robot coverage, *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems,* International Foundation for Autonomous Agents and Multiagent Systems, 1775–1776.

[8] Brun, Y., (2008). Building Biologically- Inspired Self-Adapting Systems, *Dagstuhl Seminar Proceedings*, Schloss Dagstuhl- Leibniz-Zentrum fr Informatik.

[9] Capera, D., George, J.P., Gleizes, M.P. and Glize, P., (2003). The AMAS theory for complex problem solving based on self-organizing cooperative agents, *Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on*, IEEE, 383–388.

[10] Colorni, A., Dorigo, M., Maniezzo, V. and others, (1991). Distributed optimization by ant colonies, *Proceedings of the first European conference on artificial life*, (142), 134–142.

[11] Cuperlier, N., Guedjou, H., de Melo, F., and Miramond, B., (2016). Attention-based smart-camera for spatial cognition, *Proceedings of the 10th International Conference on Distributed Smart Camera*, ACM, 121–127. https://doi.org/10.1145/2967413.2967440

[12] Da Silva, J.L.T. and Demazeau, Y., (2002). Vowels co-ordination model, *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 3*, ACM, 1129-1136.

[13] Dorigo, M., Maniezzo, V. and Colorni, A., (1991). The ant system: An autocatalytic optimizing process. *Tech. rep.*, Italy: Dipartimento di Elettronica, Politecnico di Milano. PMid:1810600 PMCid:PMC1908841

[14] Dorigo, M. and Gambardella, L., (1997). Ant colonies for the travelling salesman problem. *BioSystems*, Elsevier, (43), (2), 73–81. https://doi.org/10.1016/S0303-2647(97)01708-5

[15] Dorigo, M. and Di Caro, G., (1999). Ant colony optimization: a new meta-heuristic, Evolutionary

Computation, *CEC 99. Proceedings of the 1999 Congress on*, IEEE, (2), 1470–1477.

[16] Dréo, J., Pétrowski, A., Siarry, P. and Taillard, E., (2006). *Metaheuristics for hard optimization: methods and case studies*, Springer Science & Business Media.
PMCid:PMC1569826

[17] Ferber, J., and Müller, J-P., (1996). Influences and reaction: a model of situated multiagent systems, *Proceedings of Second International Conference on Multi-Agent* Systems (ICMAS-96), 72–79.

[18] Ferber, J. and Gutknecht, O., (1998). A meta-model for the analysis and design of organizations in multi-agent systems, *MultiAgent Systems, Proceedings. International Conference on*, IEEE, 128–135.

[19] Ferber, Jacques, (1999) *Multi-agent systems: an introduction to distributed artificial intelligence*, Addison-Wesley Reading.
PMCid:PMC1736616

[20] Ferber, J., Michel, F. and Báez, J., (2004). AGRE: Integrating environments with organizations, *International Workshop on Environments for Multi-Agent Systems*, Springer, 48–56.

[21] Fink, G.A., Haack, J.N., McKinnon, A.D. and Fulp, E.W., (2014). Defense on the move: ant-based cyber defense, *IEEE Security & Privacy*, IEEE, (12), (2), 36–43.

[22] Florin Leon, Marcin Paprzycki, and Maria Ganzha. (2015). A Review of Agent Platforms, *Multi-Paradigm Modelling for Cyber-Physical Systems (MPM4CPS), ICT COST Action IC1404*, 1–15.

[23] Fogel, D.B., (1988). An evolutionary approach to the traveling salesman problem, *Biological Cybernetics Journal*, Springer, (60), (2), 139–144.

[24] Gengan, D., Schoeman, M.A. and Van Der Poll, J.A., (2014). An Ant-based Mobile Agent Approach to Resource Discovery in Grid Computing, *Proceedings of the Southern African Institute for Computer Scientist and Information Technologists Annual Conference 2014 on SAICSIT 2014 Empowered by Technology*, ACM, 1.
PMid:25528196

[25] Gonçalves, F. A.C.A., Guimarães, F.G. and Souza, Marcone J.F., (2013). An evolutionary multi-agent system for database query optimization, *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, ACM, 535–542.
PMid:23278174

[26] Gutknecht, O. and Ferber, J., (2000). The madkit agent platform architecture, *Workshop on Infrastructure for Scalable Multi-Agent Systems at the International Conference on Autonomous Agents*, Springer, 48–55.

[27] Hong, T-P., Huang, L-I. and Lin, W-Y., (2014). A Different Perspective on Parallel Sub-Ant-Colonies, *Proceedings of the 12th International Conference on Advances in Mobile Computing and Multimedia*, ACM, 322–325.
PMCid:PMC4346605

[28] Huget, M-P., (2014). Agent Communication, *Agent-Oriented Software Engineering*, Springer, 101–133.

[29] Jennings, N.R., Sycara, K. and Wooldridge, M., (1998). A roadmap of agent research and development, *Autonomous agents and multi-agent systems*, Kluwer Academic Publishers, (1), (1), 7–38.

[30] Jennings, N.R., (2001). An agent-based approach for building complex software systems, *Communications ACM Journal*, ACM, (44), (4), 35–41.

[31] Karmani, R.K., and Shali, A. and Agha, G., (2009). Actor frameworks for the JVM platform: a comparative analysis, *Proceedings of the 7th International Conference on Principles and Practice of Programming in Java*, ACM, 11–20.

[32] Karmani, R.K. and Agha, G., (2011). Actors, *Encyclopedia of Parallel Computing*, Springer, 1–11.

[33] Kosakaya, J., (2016). Multi-agent-based SCADA system, Event-based Control, Communication, and Signal *Processing (EBCCSP), 2016 Second International Conference on*, IEEE, 1–5.

[34] Lee, U., Magistretti, E., Gerla, M., Bellavista, P., Li´o, P. and Lee, K.W., (2009). Bio-inspired multi-agent data harvesting in a proactive urban monitoring environment, *Ad Hoc Networks Journal*, Elsevier, (7), (4), 725–741.

[35] Lin, C-T. and Lee, C.S.G., (1991). Neural-network-based fuzzy logic control and decision system, *IEEE Transactions on computers*, IEEE, (40), (12), 1320–1336.

[36] Lodding, K.N., (2004). The Hitchhiker's Guide to Biomorphic Software, *ACM Queue Journal*, ACM, (2), (4), 66–75.

[37] Ma, J., Man, K.L., Ting, T. Zhang, N., Guan, S-U. and Wong, P.WH., (2014). Accelerating Parameter Estimation for Photovoltaic Models via Parallel Particle Swarm Optimization, *Computer, Consumer and Control (IS3C), International Symposium on*, IEEE, 175–178.

[38] Manate, B., Fortis, F. and Moore, P., (2014). Applying the Prometheus methodology for an Internet of Things architecture, *Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, IEEE Computer Society, 435–442.

[39] Mariani, S. and Omicini, A., (2016). Multi-paradigm Coordination for MAS: Integrating Heterogeneous Coordination Approaches in MAS Technologies, *WOA*, 91–99.

[40] Massawe, L.V., Aghdasi, F. and Kinyua, J., (2009). The development of a multi-agent based middleware for RFID asset management system using the PASSI methodology, *Information Technology: New Generations, ITNG'09. Sixth International Conference on*, IEEE, 1042–1048.

[41] Michel, F., (2004). *Formalism, tools and methodological elements for the modeling and simulation of multi-agents systems*, PhD thesis, LIRMM, Montpellier, France.

[42] Michel, F., Ferber, J., Drogoul, A. and others, (2009). Multi-Agent Systems and Simulation: a

Survey From the Agents Community's Perspective, *Multi-Agent Systems: Simulation and Applications Journal*, 3–52.

[43] Mirjalili, S., Mirjalili, S.M. and Lewis, A., (2014). Grey wolf optimizer, *Advances in engineering software*, Elsevier, (69), 46–61.

[44] Mochalov, V., (2015). Multi-agent bio-inspired algorithms for wireless sensor network design, *Advanced Communication Technology (ICACT), 17th International Conference on*, IEEE, 33–42.

[45] Nagpal, R., (2003). A catalog of biologically-inspired primitives for engineering self-organization, *International Workshop on Engineering Self-Organising Applications*, Springer, 53–62.

[46] Olaifa, M., Mapayi, T. and Van Der Merwe, R., (2015). Multi Ant LA: An adaptive multi agent resource discovery for peer to peer grid systems, *Science and Information Conference (SAI)*, IEEE, 447–451.

[47] Padmanaban, R., Thirumaran, M., Suganya, K. and Priya, R.V., (2016). AOSE Methodologies and Comparison of Object Oriented and Agent Oriented Software Testing, *Proceedings of the International Conference on Informatics and Analytics*, ACM, 119.
https://doi.org/10.1145/2980258.2982111

[48] Parunak, H.V.D., (1997). "Go to the ant": Engineering principles from natural multi-agent systems, *Annals of Operations Research Journal*, JC BALTZER AG, (75), 69–102.

[49] Perez-Carabaza, S., Besada-Portas, E., Lopez-Orozco, J.A. and de la Cruz, J.M., (2016). A Real World Multi-UAV Evolutionary Planner for Minimum Time Target Detection, *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference*, ACM, 981–988.
https://doi.org/10.1145/2908812.2908876

[50] Perez-Diaz, F., Zillmer, R. and Groß, R., (2015). Firefly-Inspired Synchronization in Swarms of Mobile Agents, *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems, International Foundation for Autonomous Agents and Multiagent Systems*, 279–286.

[51] Qian, B. and Cheng, H.H., (2016). A mobile agent-based coalition formation system for multi-robot systems, *Mechatronic and Embedded Systems and Applications (MESA), 2016 12th IEEE/ASME International Conference on*, IEEE, 1–6.

[52] Rehberger, S., Spreiter, L. and Vogel-Heuser, B., (2016). An agent approach to flexible automated production systems based on discrete and continuous reasoning, *Automation Science and Engineering (CASE), IEEE International Conference on, IEEE*, 1249–1256.

[53] Rowley, H.A., Baluja, S. and Kanade, T., (1998). Neural network-based face detection, *Pattern Analysis and Machine Intelligence Journal*, *IEEE Transactions on*, IEEE, (20), (1), 23–38.

[54] Silva, D.C., Braga, R.A-M., Reis, L.P. and Oliveira, E., (2010). A generic model for a robotic agent system using GAIA methodology: Two distinct implementations, *Robotics Automation and Mechatronics (RAM), IEEE Conference on*, IEEE, 280–285.

[55] Sturm, A. and Shehory, O., (2014). Agent-oriented software engineering: revisiting the state of the art, *Agent-Oriented Software Engineering*, Springer, 13–26.

[56] Sturm, A. and Shehory, O., (2014). The landscape of agent-oriented methodologies, *Agent-Oriented Software Engineering*, Springer, 137–154.

[57] Stützle, T. and Hoos, H., (1998). Improvements on the ant-system: Introducing the max-min ant system. *Artificial Neural Nets and Genetic Algorithms*, Springer, 245–249.
https://doi.org/10.1007/978-3-7091-6492-1_54

[58] Tarakanov, A., (2001). Information security with formal immune networks, *Information Assurance in Computer Networks Journal, LNCS*, Springer, (2052), 115–126.

[59] Tsang, C.H. and Kwong, S., (2005). Multi-agent intrusion detection system in industrial network using ant colony clustering approach and unsupervised feature extraction, *Industrial Technology, ICIT 2005. IEEE International Conference on*, IEEE, 51-56.

[60] Wang, J., Cao, J., Li, B., Lee, S. and Sherratt, R. S., (2015). Bio-inspired ant colony optimization based clustering algorithm with mobile sinks for applications in consumer home automation networks, *IEEE Transactions on Consumer Electronics*, IEEE, (61), (4), 438–444.

[61] Weyns, D., Parunak, H.V.D. and Michel, F., (2006). Environments for Multi-Agent Systems II, Second *International Workshop, E4MAS 2005, Utrecht, The Netherlands, July 25, 2005, Selected Revised and Invited Papers*, Springer, (3830).

[62] Wooldridge, M. and Jennings, N.R., (1994). Agent theories, architectures, and languages: a survey, *International Workshop on Agent Theories, Architectures, and Languages*, Springer, 1–39.

[63] Xiang, W. and Lee, HP., (2008). Ant colony intelligence in multi-agent dynamic manufacturing scheduling, *Engineering Applications of Artificial Intelligence Journal*, Elsevier, (21), (1), 73–85.

[64] Zambonelli, F., (2015). Engineering Environment-Mediated Coordination via Nature-Inspired Laws, *Agent Environments for Multi-Agent Systems IV*, Springer, 63–75.

[65] Zambonelli, F., Omicini, A., Anzengruber, B. and others, (2015). Developing pervasive multi-agent systems with nature-inspired coordination, *Pervasive and Mobile Computing*, Elsevier, (17), 236–252.

[66] Zeghida, D., (2003). MALCS: *Multi-Agent Learning Companions System,* DEA thesis, Badji Mokhtar Annaba University, Algeria.

[67] Zeghida, D., Meslati, D., Bounour, N. and Allat Y., (2018). Agent Influence/Reaction Ant System Variants: An Experimental Comparison, International Journal of Artificial Intelligence, (16), (2), 60-77.
EID: 2-s2.0-85053662680

Empirical Study on the Optimization Strategy of Subject Metro Design Based on Virtual Reality

Zhendong Wu
College of Mechanical Engineering and Automation, Huaqiao University, Xiamen, Fujian 361021, China
E-mail: zdongwuhqu@yeah.net

Technical Paper

Keywords: virtual reality eye movement, visual attention, theme subway, design strategy

Received: August 23, 2018

A three-dimensional simulation interactive virtual scene was established taking the theme subway in Chengdu and Guangzhou as the typical case, and the standard metro in Xiamen as the reference. An experiment was designed using virtual reality built-in eye movement equipment and following the principle of visual attentiveness. The conscious and unconscious visual behaviors of users were analyzed and the impacts of different design methods on user experience and behaviors were analyzed. This study extracted the key elements of the theme subway design and recombine them to compared the design of facilities at the same position but in different themes and the design of space interface at different positions but in the same theme. Moreover, optimization strategies were put forward for the design of theme subway space to enhance the availability of the design.

Povzetek: Predstavljena je virtualna študija za učinkovito predstavitev podzemnih kitajskih železnic uporabnikom.

1 The status of internal design of theme subway

According to the statistics of China's rail transit network, 24 of 30 cities which have subway have opened theme subway, and 69 themes are included [i]. Theme design refers to a design method of setting up a series of scenes or events in the form of visual creativity by means of narrative techniques to establish a transfer link with audience. Theme subway which applies theme design is taken as the material carrier of social and cultural information, which solves the problem of characteristic crisis of space [ii], makes people perceive the cultural atmosphere and connotation of a city or region, and enhance the identifiability of a city.

The current design of theme subway in China focuses on the publicity of theme, still in the initial stage. Design ideas of the internal space of subway is monotonous. The design ideas include texture design and design of three-dimensional modeling. Texture design means directly applying theme related picture materials on the wall surface of carriages. Design of three-dimensional modeling means transforming three-dimensional cultural model or plane cultural elements to three-dimensional models, taking them as facilities or decoration carriers, and endowing them with practical or decorative functions.

Wang Dawei from Shanghai Academy of Fine Arts proposed the theoretical model of space design of subway station which is a cube model composing of professional emphasis, design elements and subway space. The design should consider not only visual aesthetics but also the security, comfortability, economical efficiency and sustainability of metro space design [iii]. The design of theme subway needs to reduce the psychological pressure of passengers in the claustrophobic space from the psychological point of view and improve the visual comfort of passengers in the subway environment. As to the content of theme culture, the design should guide passengers to form short-term memory and information feedback. The current theme subway is mainly based on experience design; the good and bad efficacy are intermingled because of the lack of scientific basis and experimental verification.

2 Reflection of design of theme subway based on the principle of visual attention

James (1890) first proposed that the directivity and centrality of perception are the two basic characteristics of attention and only a few objects will be noticed at the moment when objects are sensed [iv]. It shows that attention has the role of screening. In the process of visual scene observation, the user's visual presentation is progressive and incomplete [v].When attention is focused on the perceived area, consciousness can capture the stimulation of interest in that area. Von Helmholtz (1925) and James (1890) put forward "where"and "what" [vi]."Where" is a view put forward by Vol Helmholtz. He focused on the relationship between eye movement and spatial position.

It means that involuntary attention is a fixation behavior which is based on individual experience or task objectives and controlled by consciousness or autonomous behaviors. "What" is a view put forward by James. He thought that attention is a mechanism with hidden inner. It is active and voluntary and relates to properties, significance and expectation of attention focus, involving information processing, refers to voluntary fixation behavior under an unconscious state. Short-term memory and even long-term memory will form when people use the attention of high-level cognitive ability.

Attention is an important psychological adjustment mechanism in the process of visual information processing, and it is not only related to individual cognition, but also influenced by emotional mechanism [vii]. In *Emotional Design*, Donald Norman proposed three levels of brain information processing, in which the instinctive level is the subconscious judgment determined by biological heredity (subconscious judgment of individual behavior), and the behavior layer is a habit based reaction (based on the individual experience). They correspond to high level of attention (involuntary attention) and low level of attention (voluntary attention) respectively.

According to the above principles, when passengers enter a carriage, observation of the instinctive layer is firstly induced, and the attention is active and voluntary at that moment. If a certain area in a scene arises passengers' interest, the area will be perceived by the vision around the central fovea, and then more detailed content is perceived. The visual fixation area and time data of passengers in this state are obtained through first fixation time among eye movement indexes. First fixation time refers to early identification process of area of interest and sensitivity to processing difficulty of area of interest. Shorter time indicates that the region is easier to be concerned by users [viii]. The visual design features of the interest area inside the carriage were summarized. In addition, the involuntary attention of the behavior level is controlled by consciousness and autonomous behaviors. The scope of attention is very small; hence passengers perceive all the parts of the scene through continuous scanning. Whether the design of carriage induces the short-term memory of passengers can be analyzed based on the division of area of interest of heat map and retention time. Area of interest (AOI): It is usually used for design availability analysis. Some element of interface is isolated as a specific area or content for further analysis [ix]. Heat map refers to information visualization graph which presents eye movement data in the form of cloud picture [x]. It can intuitively present the area of interest of the subject and analyze the focus area and retention time. Retention time: retention time is a very good index for testing the degree of interest for a specified area of interest. Longer retention time means more interests of users on an area of interest [xi].

3 Necessity of virtual reality eye movement linkage experiment

In the past, the experimental research on internal design of rail transit was mainly based on rendering pictures and portable eye movement equipment. The maturity of the technology which combines head mounted virtual equipment with eye movement instrument in 2016 provides technical support for the accuracy of data and control of independent variables of such kind of study [xii]. Compared with the previous experiments, the advantages of the experiments which apply the new technology are mainly reflected in the following three aspects.

(1) Strong immersion in visual scene and more objective data

The visual scene is a three-dimensional simulation model with high preciseness, which restores the internal facilities, lighting, dimension sense of space and texture of the real scene; therefore subjects can obtain real experience in the virtual environment [xiii]. For example, in the course of the experiment, subjects try to grasp a handrail after entering the scene, which shows that the scene is very vivid and more objective eye movement data can be obtained. In the previous experiments, two-dimensional pictures were usually used as stimuli, and subjects cannot feel immersed, which affected the objectivity of data.

(2) Extraction of implicit data and recording while looking

The experiment of portable head eye tracker combined with pictures is very difficult for users to concentrate due to the large error of eye movement data and the small size of stimulon. The virtual reality scene can be observed in 360 degrees, and the subjects' perspective is large, which can not be constrained by the size of picture. It can collect the eye tracking data consciously and unconsciously (implicit) in the virtual scene in real time to perfect categories of data [xiv].

(3) Effective control of independent variables is beneficial to comparison

Design factors which can affect user experience include content of theme, shape design, area of pattern and position of decoration. Changes of variables in virtual eye movement experiments will generate different scenes; in this way, scene changes can be realized under no disturbance. It is beneficial for comparing reactive states of user experience and analyzing the relationship between variables and design. For example, visual attention of subject will transfer when the color saturation inside subway carriage is too high.

4 Experimental design

Taking the theme subway in Xiamen, Chengdu and Guangzhou as the research subjects, this study established a three-dimensional simulation interactive virtual scene. The eye movement of users in the scene was recorded. The influence of different designs on user experience was compared and analyzed based on the

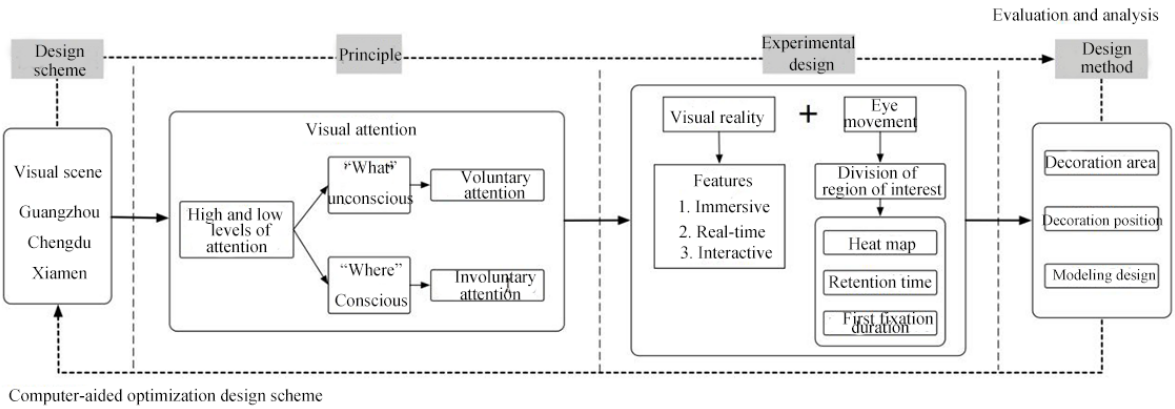


Figure 1: Research ideas.

Figure 2: Experimental samples.

principle of visual attention. Based on it, design strategies for subway space were summarized.

The technology combing virtual reality with eye movement which was independently developed by Shanghai Qingtech Co., Ltd., China. An eye tracking module was inserted into HTC vive to track and record the real-time eye movement data in the virtual visual scene. Research ideas are shown in Figure 1.

4.1 Survey of design cases and selection of samples

Sixty-nine design cases of theme subway in 24 cities in China were collected, and two of them was selected as the representative experimental samples. Panda theme subway on Line 3 in Chengdu and cartoon theme subway in Guangzhou were selected as comparison samples, and the standard subway on Line 1 in Xiamen was selected as the reference of this study, as shown in Figure 2.

In the panda theme subway on Line 3 in Chengdu, panda which is a regional cultural characteristic of Chengdu was taken as the design element, and the image of panda was integrated into the appearance design of seat, handle and side walls to create three-dimensional models.

The standard subway on Line 1 in Xiamen has no theme, which is the mode of most standard subways in China. Analysis on recombination of design elements. As the design of theme subways involves many factors, pattern area,modeling technique and decoration position were selected as the key elements for comprehensive analysis.

The proportion of pattern area refers to the proportion of the pattern area inside a carriage to the total area, and it has three grades, 10% ~ 30%, 30% ~ 60% and 60% ~ 100%.

Modeling techniques include design of three-dimensional modeling and design of texture. Design of three-dimensional modeling mainly focuses on positions of handles, handrailings and seats, while design of three-dimensional modeling focuses on side walls, end walls and top surface.

Design elements were classified and then recombined. Two design issues were analyzed. The first issue was the comparison of subway facility design in different themes but at the same position, and the second issue was the comparison of visual perception of different spatial interface design in the same theme but at different positions. Based on it, design strategies of hardware facilities and interface which was more in line with the principle of visual attention could be put forward. Details are shown in Figure 3.

4.2 Subjects

In this experiment, there were 30 subjects, aged 18 ~ 35 years. In order to ensure that all the subjects had the same cognitive level, all of them had no virtual reality experience, but had the experience of taking the subway. The uncorrected or corrected visual acuity of the subjects were normal, and neither of them had color blindness. At the beginning of the experiment, the subjects were asked to receive an eye movement calibration test which lasted for 30 ~ 60 s in the scene. The formal test started after the eye movement calibration; they had a visual activity

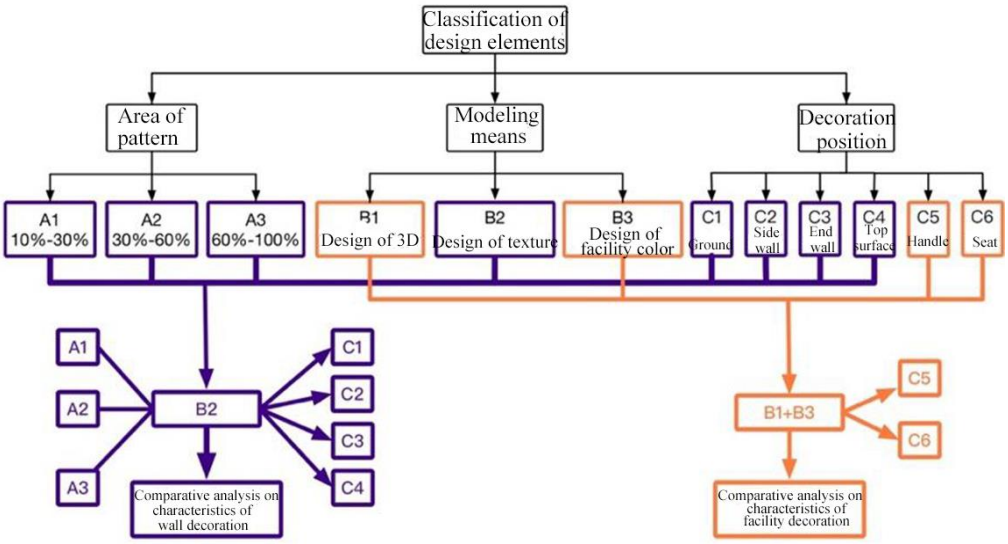


Figure 3: Two design issues corresponding to the recombination of design elements.

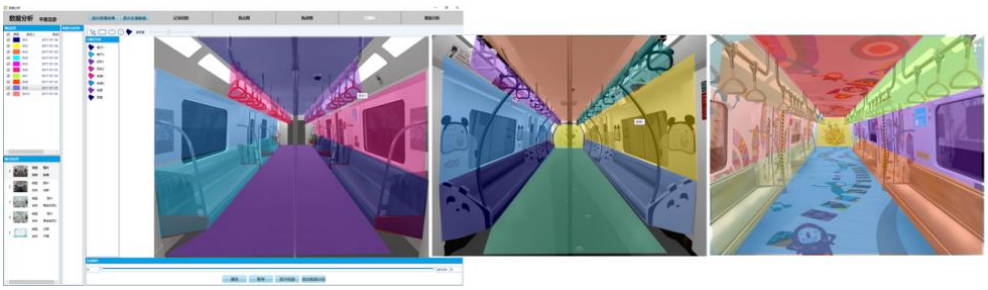


Figure 4: The division of area of interest in three subway carriages.

of random observation in the virtual scene which lasted for 120 s.

4.3 Analysis of experimental results

Firstly, the area of interest inside the subway was divided. As shown in Figure 4, the subway carriage was divided into six regions of interest according to the space region and functional facilities: top surface, side walls, end walls, ground, handles and seats. The first fixation duration and retention time in the six regions of interest were recorded. The mean values and variances of the eye movement data were statistically analyzed using SPSS to evaluate the internal design of the carriage.

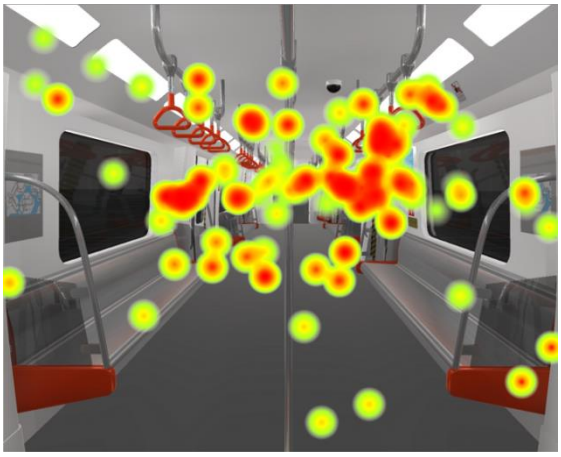


Figure 5: Analysis of the heat map of a standard subway.

4.3.1 The comparison of facility design in the same theme but at different positions

User will unconsciously observe firstly when he enters a [Pritegnite pozornost bralca z odličnim citatom iz dokumenta ali pa izkoristite ta prostor, da poudarite ključno točko. Če želite premakniti to polje z besedilom na katero koli drugo mesto na strani, ga preprosto povlecite.]
carriage for the first time. The impact of different subway facility designs on the attention of users was analyzed. According to the analysis of the heat map of the standard subway, it was found that the seat and handrail facilities of the subway were the concerns of passengers, as shown in Figure 5. Therefore, different designs of positions of handrails and seats in the subway carriage was compared. The fixation condition of users in an unconscious state was analyzed by performing descriptive analysis on the first fixation time of users, as shown in Table 1.

Different design methods for the same location and different themes have different effects on eye movement data. The three-dimensional design of the seats and handles which took panda as the element in the Chengdu theme subway attracted the most interests and attentions from users, as shown in Figure 6. The minimum value was 43.83, and the sensitivity was high in the early recognition process. Next was the standard subway, the

Table 1: The descriptive analysis results of the first fixation in the facility design.

Descriptive statistics of the first fixation time in the facility design					
Facility	**Experiemntal samples**	**Mean value**	**Remark**	**Standard deviation**	**Remark**
Seats	Standard subway	70.27	Smaller mean value means users are more likely to pay attention to the facility.	73.21	Smaller value of standard deviation means smaller difference of experience tendency.
	A subway in Chengdu	43.83		57.13	
	A subway in Guangzhou	82.00		81.76	
Handles	Standard subway	65.49		69.97	
	A subway in Chengdu	51.55		50.79	
	A subway in Guangzhou	68.24		42.83	
Remark: the unit of the first fixation time is second.					

Figure 6: Design of facilities.

handles and seats were red with high saturation degree, which was in sharp contrast with the surrounding environment. Its value was larger than the Chengdu subway (70.27 > 43.83). Therefore, the three-dimensional design was better than the high saturation color design. The handles and seats of the theme subway in Guangzhou were gray and unified with the surrounding environment. Its value was the highest (82.00 > 70.27 > 43.83) and had a large gap with the eye movement data of the other subways. Therefore, model color design which was close to the environmental color had the least attractiveness and the lowest sensitivity to the early recognition reaction.

Through the above analysis, it was concluded that there were two design methods of theme subway facility. The first one was design of color, and the second one was design of three-dimensional modeling. Both had advantages and disadvantages. The design strategies are shown in Table 2.

In China, the Disney theme subway in Hongkong is a combination of three-dimensional modeling and color design, which conveys the theme of Mickey always accompanies with passengers. In the design of windows

Table 2: The design strategy for hardware facilities in theme subway.

	Design method of facility	
	Design of color	Design of three-dimensional modeling
Advantages	High economical efficiency: low construction cost	1. The application of thematic image has strong visual attraction. 2. High identifiability and highly sensitive to the early recognition reaction. The application of personification design and skeuomorph makes facilities more visually hierarchical.
Disadvantages	1. Weak visual attraction 2. Low sensitivity to reaction and weak identifiability 3. Similar design style	1. Complex manufacturing technique (customized design) 2. Higher cost compared to the design of standard subway 3. Individualized design for every theme, lacking of sustainability.
Conclusions for optimizing strategy	Three-dimensional modeling + color design 1. Abstract design of thematic images (serialization design of handle, support rod and connecting rod) 2. Serialization design of handles and seats to strengthen content of theme 3. Pay attention to the complementarity of facility color and surrounding environment in subway carriage.	

Table 3: The sorting of average retention time of different space interface.

The sorting of average retention time of different space interfaces												
		1		2		3		4		5		6
Guangzhou	Top surface	6.33	End wall	4.80	Ground	4.33	Side wall	3.47	Seat	1.79	Handrail	1.46
Chengdu	Side wall	3.81	Ground	1.87	Seat	1.87	Top surface	1.56	End wall	1.35	Handrail	1.28
Standard	Side wall	2.84	Handrail	1.1.8	Ground	1.11	Top surface	0.91	Seat	0.89	End wall	0.77
Remark: the unit of the retention time is second.												

and handles, the three-dimensional design of "Mickey head" is adopted. The three-piece handle attracts attentions of passengers by contrast colors, red, yellow and black. The arrangement of the seats breaks out the previous end-to-end arrangement. The L-shaped blue corner cloth sofa contrasts vividly with the yellow on the surrounding supports in the whole space. The echoing of color stimulus and theme modeling also leave a deep impression on people, as shown in Figure 7 [xv].

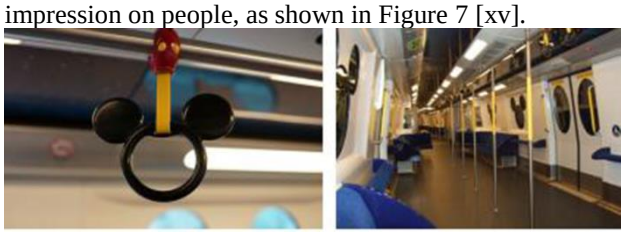


Figure 7: The design of internal facilities of the Disney theme subway.

The space interface of subway carriage can be divided into top surface, ground, side walls and end walls; passengers pay more attentions to these four parts. Therefore, dual requirements of functional technology and the aesthetic level of space need to be satisfied. Taking the theme subway in Guangzhou and Chengdu as examples, the influence of wall design on visual retention and attention of users was discussed to conclude the design features of different locations inside the subway [xvi].

4.3.2 The comparison of facility design in the same theme but at different positions

First of all, variance analysis of retention time of eyes was made. The significance of position * theme Sig=0.027<0.05 indicated a significant difference; it meant that the decoration position was interactive with theme design. Eye retention time of the decoration

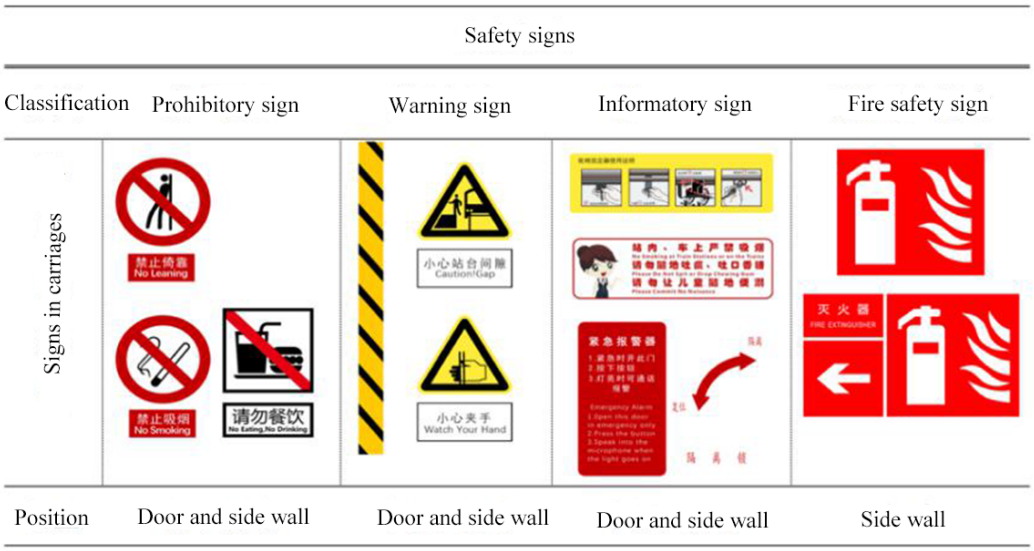


Figure 9: Functional information signs and the distribution positions.

position varied with the theme. Significance Sig=0<0.05 indicated a statistically significant difference in data of different locations. However, the variance analysis of the first fixation time found that significant of position * theme Sig=0.762>0.05 indicated no statistically significant difference, showing that the user was in the unconscious state and the decoration position was non-interactive with the theme design. For further analysis, data were processed by descriptive statistics, and the eye movement data at different locations on the same theme were sorted preferentially, as shown in Table 3.

Through descriptive analysis, it was found that users had different degrees of information processing at different locations after entering the carriage. The comparison of the top three positions suggested there was a commonality although users had different fixation points. In all regions of interest, ground and side walls in all the carriages were observed, which conformed to the behavioral mode of people in subway; they were also the keys in the design.

The side walls of Chengdu theme subway is designed based on panda and labeled with text information, as shown in Figure 8. The first fixation time of the side walls was 60.13 s, indicating that it was paid less attention to compared to the Guangzhou subway. But after a long-time observation, the fixation time of the side walls was 3.81 s; with a high readability, it could attract more attentions and interests.

Therefore, it could be concluded that the side wall is an important position which users will pay attention to for a long time. Situational decorative design was not suitable for side walls because of the region segmentation and functional information, as shown in Figure 9. These signs can provide information services such as instructions, hints and warnings to passengers through visual communication, which plays a key role in the safety of passengers in the subway station. Without affecting the search of functional information, small texture design or three-dimensional modeling design can be used.

Figure 8: Text information on the side walls of the Chengdu subway.

The Guangzhou theme subway focused on texture design, and users paid more attentions to the top surface (6.33 s), the end wall (4.80 s) and the ground (4.33 s) which had complete content. Due to the pattern integrity and sufficient scene presentation of the top surface, users observed it for a long time and paid the most attentions to it. Therefore top surface was the best place to display the situational theme design; while maintaining the spatial integrity, it would not interfere with the search of the functional information inside the carriage. Currently, there are few theme subways with designed top surface, and it is also easily to be ignored by designers. For the design of ground, small area or monotonous color design can be used as it has certain behavioral functions and easy to wear because of the large staff mobility.

In the process of experiment, the visual attention reaction of users was tested by changing the saturation of the side walls of the theme subway in Guangzhou. When the saturation was too high or low, the user's attention was quickly transferred to other areas. Therefore, it was concluded that color saturation was an important factor affecting the visual attention of users. It was suggested that designers use moderate saturation for the overall space of carriage, and the area that needs to be noticed by the user can be used in the contrast of high saturation color. The optimization strategy is shown in Table 4.

Table 4: The summary of optimization strategies for the interface design of theme subway.

Position	Characteristics of positions	Conclusions of optimization strategy
Side walls	1. Region division and many functional information 2. Small designable area 3. An area which is focused on 4. High information readability	Consideration for safety: 1. Design of small-area texture or design of three-dimensional modeling is allowed on the premise of not affect searching of functional messages. 2. Not suitable for large area of situational pattern design 3. Suitable for reading of text messages in theme design 4. Pattern design around guiding messages in the area of side door is not suitable as it will increase time of searching messages.
Top surface	1. Large designable area 2. No region division on the wall surface, with a high integrity 3. Presenting visual height	**Consideration for comfort** 1. Present situational theme design with design of texture 2. Not suitable for layout of text information 3. Passengers may feel reduced visual height and feel depressive because of complex pattern 4. The saturability, relative brightness and hue of patterns has large influence in improving the visual height of space (visual perception stratification).
Ground	1. Large designable area 2. No region division on the wall surface, with a high integrity 3. With a behavioral function 4. Easy to wear	Consideration for function 1. Design of small-area texture or monotonous color design are feasible. 2. Large-area texture design is not suitable as the large passenger flow in the carriage is prone to cause wearing.

5 Conclusion

Several design strategies of color, shape and texture were developed based on virtual reality technology, reference to different theme subway space, eye movement data and subjective evaluation for the design of hardware facilities and interface in theme subway, which can provide a reference for future design. In future research, virtual reality technology in combination with eye movement technology can be used for the study of the spatial availability of the environment to offer users a better experience in the space.

6 Acknowledgement

This study was supported by the 2017 Youth Fund of Humanities and Social Sciences Research of Ministry of Education under grant number of 17YJC760100.

7 References

[i] Zhao JP (2016). Summary of Chinese theme subway. Time Report, (40).

[ii] Wu LW (1989). Generalized Architecture. Tsinghua University Press.

[iii] Guo XY, Wang ZS (2014). Metro Station Space Environment Design: method, procedures and example. China Water Power Press.

[iv] James W (1890). The principles of psychology, Vol. 2. NY, US: Henry Holt and Company. https://doi.org/10.1037/10538-000

[v] Duchowski AT (2015). Eye Tracking Methodology Theory and Parctice. Zhao XB, Ziu XC, Zhou YJ translated. Beijing: Science Press, pp. 3-4. https://doi.org/10.1007/978-1-84628-609-4

[vi] Helmholtz H, Southall JPC (1925). Treatise on physiological optics. III. The perceptions of vision.

[vii] Liang Y, Liu HZ (2010). Study of Image Retrieval Based on Vision Attention Mechanism. Journal of Beijing Union University, (1), pp. 30-35.

[viii] Duchowski AT (2003). Eye Tracking Methodology: Theory and Practice. Springer London. https://doi.org/10.1007/978-1-84628-609-4

[ix] Eckhoff ND, Ervin PF (1971). Area-of-interest unfolding. Nuclear Instruments & Methods, 97(2), pp. 263-266. https://doi.org/10.1016/0029-554X(71)90280-1

[x] Hashimoto Y, Matsushita R (2012). Heat Map Scope Technique for Stacked Time-series Data Visualization. International Conference on Information Visualisation, IEEE, Montpellier, France, pp. 270-273. http://doi.ieeecomputersociety.org/10.1109/IV.2012.53

[xi] Duan CH, Yan ZQ, Wang FX (2012). Eye Movement Research on the Influence of Animation Rendering Speed on Multimedia Learning Effect.

National Psychological Academic Conference, pp. 46-53.

[xii] Farooq U, Glauert J, Zia K (2017). Load Balancing for Virtual Worlds by Splitting and Merging Spatial Regions. Informatica, 42(1).

[xiii] Liu XX (2016). Research on the Influencing Factors of Visual Cognition of "Dynamic Illumination"——Taking the Nightscape Lighting of "Tokyo Skytree" in Japan as an Example. Decoration, (3), pp. 101-103.

[xiv] Nini B (2013). Bit-projection based color image encryption using a virtual rotated view. Informatica, pp. 283-291.

[xv] Zhao ZC, Zhao J, Zhou C (2015). Saving Design Practice and Exploration: The Conceptual Design of Public Space about Nanjing Metro Line 2. Art & Design, (8), pp. 87-89.

[xvi] Zhong X (2017). The study of city public traffic visual identification system of regional characteristics-a case study of Chengdu. Southwest Jiaotong University.

Defect Features Recognition in 3D Industrial CT Images

Haina Jiang
Chongqing College of Electronic Engineering, Chongqing City, 401331, China
Corr. address: No. 70-2-4-3, Huxi Garden, University Town, Shapingba District, Chongqing City, 401331, China
E-mail: hainjiangcq@126.com

Technical Paper

Keywords: industrial CT image, defect location, image segmentation, feature extraction, feature recognition

Received: September 15, 2018

Due to the limitations of production conditions, there is a certain probability that workpiece product has internal defects, which will have a certain impact on the performance of workpiece. Therefore, the internal defects detection of workpiece is essential. This study proposed a defect recognition method based on industrial computed tomography (CT) image to identify the internal defects of workpiece. The block fractal algorithm was used to locate the defect parts of the image, then the improved k-means clustering algorithm was used to segment the defect parts, and feature vector was extracted by Hu invariant moments. Finally, the firefly algorithm and radial basis function (RBF) neural network were combined to identify the defect. It was found from the experiments that the algorithm in this study had the accuracy of 97.89%, which proved the reliability of the algorithm and provided some suggestions for the defects recognition.

Povzetek: Za prepoznavanje okvar na 3D slikah industrijskih izdelkov je uporabljena metoda vinske mušice.

1 Introduction

Defect detection plays a very important role in the industrial field. Through defect detection, product quality can be effectively improved. Alimohamadi et al. [1] proposed a new defect detection method based on the optimal Gabor wavelet filters, which combined with morphological analysis. The experimental results on different type of textiles showed that this algorithm was robust for defects detection of various kind of textile. Chen et al. [2] proposed a new defect detection method based on dual-tree complex wavelet transform (DT-CWT) and took advantage of near shift-invariance of DT-CWT to extract weak defect feature. Experimental results demonstrated the validity of the proposed method. Sabeenian et al. [3] presented an algorithm of defect detection by making use of multi-resolution combined statistics and spatial frequency method. The accuracy obtained by the simulation using MATLAB was found to be 99%, which proved the practicality of the algorithm. Liu et al. [4] optimized subtractive clustering method (SCM) by Akaike information criterion (AIC) and then constructed radial basis function (RBF) model by using the obtained AIC-SCM algorithm, which improved the adaptability of the RBF model. Experimental results showed that this method could identify defects with a high accuracy. Leng et al. [5] used convolutional neural network method in the detection and classification of galvanized stamping parts and obtained a precision of 99.6%. Industrial computed tomography (CT) image is a simple and efficient method [6, 7] for internal defect detection of workpiece, which can effectively detect internal defects and significantly reduce the detection cost [8]. Samarawickrama et al. [9] made defect detection of tile based on industrial CT images and found it was more accurate and efficient than the manual method. In this study, based on the industrial CT image, the defect was obtained through the localization and segmentation of the defect image, then the feature extraction was conducted by using Hu invariant moments, and finally the RBF neural network which was optimized by using the firefly algorithm was used for recognizing defects to explore the reliability of this method in defect recognition.

2 Internal defect detection of workpiece

Due to the production technology, production conditions and other aspects, workpiece products often have a certain probability of internal defects. These defects not only affect the performance of workpiece, but also have certain safety risks in the actual use process. Therefore, the detection of defects is an important part of the industrial production. Industrial CT image is an effective method for workpiece defect detection. With the development of technology, the performance of industrial CT image is gradually improving, and its cost is also reducing. It has been widely used in aerospace, military, electronics, petroleum and other fields. Industrial CT images can easily be stored and analyzed, and enable quick and accurate detection of the presence or absence of defects in workpiece, as well as evaluation of the size and location of defects [10-12]. It has higher resolution

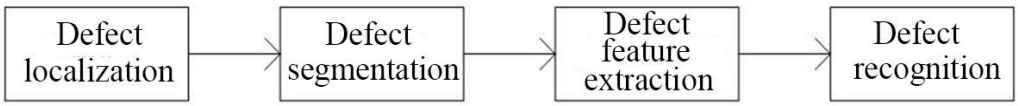


Figure 1: Flow chart of industrial CT image defect recognition.

and adaptability, so images of different gray levels can be effectively detected. At present, the detection of defects in CT images is mostly carried out manually with low accuracy. An intelligent identification method can effectively improve efficiency and reduce errors. This is the development direction of defect detection methods.

The process of defect detection based on industrial CT images proposed in this study is shown in Figure 1.

3 Defect detection algorithm

3.1 Defect localization algorithm based on block fractal

Automatic defect localization was performed using fractal theory [13]. Fractal theory is put forward by Mandelbrot, which has been extensively applied in graphics and geography. Fractal dimension [14] was obtained by Blanket algorithm which is put forward by Peleg; then the block with defects was marked, and the defect was localized.

3.1.1 Blanket algorithm

Suppose the gray function of the image is $h(i, j)$, imagine a blanket covering the gray surface of the image, suppose the upper surface as $O_\delta(i, j)$, the lower surface as $U_\delta(i, j)$, and scale as $\{\delta \mid \delta = 1,2,\cdots,N\}$, then the upper and lower surface under different scales can be expressed as:

$$O_0(i, j) = U_0(i, j) = h(i, j) \text{ , (1)}$$

$$O_\delta(i, j) = \max\left\{O_{\delta-1}(i, j)+1, \max_{|(m,n)-(i,j)|\le 1} O_{\delta-1}(m, n)\right\}, \quad (2)$$

$$U_\delta(i, j) = \min\left\{U_{\delta-1}(i, j)+1, \min_{|(m,n)-(i,j)|\le 1} U_{\delta-1}(m, n)\right\}. \quad (3)$$

According to the above expressions, the area and volume of gray surface can be calculated, and the fractal area A_δ can be obtained. The relation between the fractal area and the fractal dimension is:

$$A_\delta \approx \beta\delta^{2-W} \text{ , (4)}$$

where W stands for the fractal dimension, and β stands for the constant. The following expression can be obtained by calculation:

$$\log A_\delta \approx (2-W)\log\delta + \log\beta \text{ . (5)}$$

We see that the area of the fractal dimension is linearly related to the logarithm of the fractal dimension, and the slope of the line can be calculated to obtain the fractal dimension W:

$$W = 2 - \frac{N\sum_i \log\delta_i \log A_{\delta_i} - \sum_i \log\delta_i \sum \log A_{\delta_i}}{N\sum_i (\log\delta_i)^2 - (\sum_i \log\delta_i)^2} \text{ . (6)}$$

3.1.2 Block fractal algorithm

(1) The image is divided into rectangular regions of the same size.

(2) The corresponding fractal area A_{δ_i} of different scales δ_i on each region is calculated by the Blanket algorithm, and (A_{δ_i}, δ_i) can be obtained.

(3) The fractal dimension W can be calculated according to formula (6).

(4) Mark the fractal threshold as K. If the fractal dimension of the block is greater than this threshold, it indicates that there is an edge, and the part with the edge is marked with white.

(5) Determine if the marked blocks are workpiece edges or defect areas. As the number of blocks in defect areas is less than the workpiece edges, through calculation, if the number of blocks is greater than the connected threshold value T, it means that they are workpiece edges.

(6) Remember the defect area of the workpiece after removing the workpiece edge.

3.2 Image segmentation algorithm based on improved k-means clustering

The traditional K-means clustering algorithm may reduce the reliability. To make up the deficiencies of the algorithm, the initial clustering center automatic generation algorithm [15] was used to improve the traditional algorithm.

Suppose that T is the coordinate set of image data set D, $h(x, y, z)$ stands for the gray value of somewhere in D, $V_1^i, \cdots, V_k^i$ stands for K classes in the clustering process, $C_1^i, \cdots C_k^i$ stands for K clustering centers, and i stands for the number of iterations. The algorithm steps are as follows:

(1) Determine K and the accuracy of clustering δ.

(2) The clustering center is generated by the initial clustering center automatic generation algorithm.

(3) Take each initial clustering center as the set member of the initial cluster $V_1^i, \cdots, V_k^i$, and $C_1^1 \in V_1^1, \cdots, C_k^1 \in V_k^1$.

(4) Conduct the iterations and divide $h(x, y, z)$ into one cluster according to the minimum distance, i.e.,

$$D(T, C_l^i) < D(T, C_j^i), \text{then } h(x, y, z) \in V_l^i,$$

$$D(T, C_l^i) \ge D(T, C_j^i), \text{then } h(x, y, z) \in V_j^i,$$

where $j = 1, \cdots, k; l = 1, \cdots, k;\ j \ne l, D(T, C_l^i)$ stands for the distance function of the algorithm, i.e., the distance between $h(x, y, z)$ and the clustering center in the i-th iteration.

(5) Reset the clustering center and then cluster again. Suppose $h(x, y, z) \in V_l^i$, then its clustering center is:

$$Z_l^{i+1} = \frac{1}{N_l}\sum h(x, y, z),$$

where N_l stands for the feature points number of C_l^i at the i-th iteration. The resetting of clustering center of $h(x, y, z) \in V_j^i$ is also carried out.

(6) Repeat (4) and (5) until the clustering center remains unchanged $Z^{i+1} = Z^i$ or $\left|Z^{i+1} - Z^i\right| \le \delta$.

(7) Output image segmentation results according to clustering results. The number of clustering K stands for the number of peak values of gray histogram of reference image or the type of reference image object. The distance function is:

$$D(T, C_l^i) = \sqrt{(h(x, y, z) - C_l^i)^2}.$$

3.3 Feature extraction algorithm based on Hu invariant moments

Three common defects in workpieces are stomata, cracks and slag inclusion. The shape and gray information of the three defects are very different, and the feature information can be extracted for identification.

(1) Shape features

① The length-width ratio of the defect part is $z = \frac{R}{K}$, where R stands for the long axis and K stands for the short axis.

② The circularity of the defect is $e = \frac{L^2}{A}$, where L^2 stands for square of circumference and A stands for area.

(2) Gray information

$$mean = \frac{\sum_{x=R_{\min}}^{R_{amx}} \sum_{y=L_{\min}}^{L_{\max}} h(x, y)}{n},$$

$$v = \frac{\sum_{x=R_{\min}}^{R_{\max}} \sum_{y=L_{\min}}^{L_{\max}} (h(x, y) - mean)^2}{n},$$

where Mean stands for the Mean value of grayscale image, v stands for the variance of grayscale, $h(x,y)$ stands for the gray value of the defect pixel points, and n stands for the number of pixels.

Seven invariant moments, φ_1 - φ_7, can be obtained according to Hu invariant moment theory [16]. Table 1 is obtained after abstraction on the moments.

R1	R2	R3	R4	R5
$\frac{\sqrt{\varphi_2}}{\varphi_1}$	$\frac{\varphi_1 + \sqrt{\varphi_2}}{\varphi_1 - \sqrt{\varphi_2}}$	$\frac{\sqrt{\varphi_3}}{\varphi_4}$	$\frac{\sqrt{\varphi_3}}{\sqrt{\lvert\varphi_5\rvert}}$	$\frac{\sqrt{\varphi_4}}{\sqrt{\lvert\varphi_5\rvert}}$
R6	R7	R8	R9	R10
$\frac{\lvert\varphi_6\rvert}{\varphi_1 \times \varphi_3}$	$\frac{\lvert\varphi_6\rvert}{\varphi_1 \times \sqrt{\lvert\varphi_5\rvert}}$	$\frac{\lvert\varphi_6\rvert}{\varphi_3 \times \sqrt{\lvert\varphi_2\rvert}}$	$\frac{\lvert\varphi_6\rvert}{\sqrt{\varphi_2 \times \lvert\varphi_5\rvert}}$	$\frac{\lvert\varphi_5\rvert}{\varphi_3 \times \varphi_4}$

Table 1: Feature value obtained from the abstracted Hu invariant moment.

The above 10 feature values, two shape features (length-width ratio and degree of circularity) and two gray features (gray average and gray variance) can be used as feature vectors to identify the defects.

3.4 Defect recognition algorithm based on firefly neural network

In this study, a combination of firefly algorithm and RBF neural network was adopted to identify the defect feature. The flowchart of the algorithm is shown in Figure 2.

Establishment of RBF neural network

The number of nodes in the input layer and output layer of the neural network is 14 and 3, the hidden layer and input layer are the same, and 001, 010 and 100 stands for the stomata, cracks and slag inclusion respectively.

(1) Weight threshold optimization

① Initialize parameters: ρ stands for the volatilization rate of luciferin at t-1, γ stands for the update rate of luciferin, β stands for the change rate of field, s stands for the moving step length, rs stands for the threshold of the perceived range of firefly, and nt stands for the threshold of the number of neighbor fireflies.

② Initialization algorithm: The current position of firefly i is $X_i(t), i = 1,2,3,\cdots,N$, each firefly has the same luciferin l_0 and the same decision radius r_0.

③ The update formula of luciferin is $l_i(t) = (1-\rho)l_i(t-1) + \gamma f(X_i(t))$, $l_i(t-1)$ stands for the luciferin of firefly i at t-1, $f(X_i(t))$ stands for the

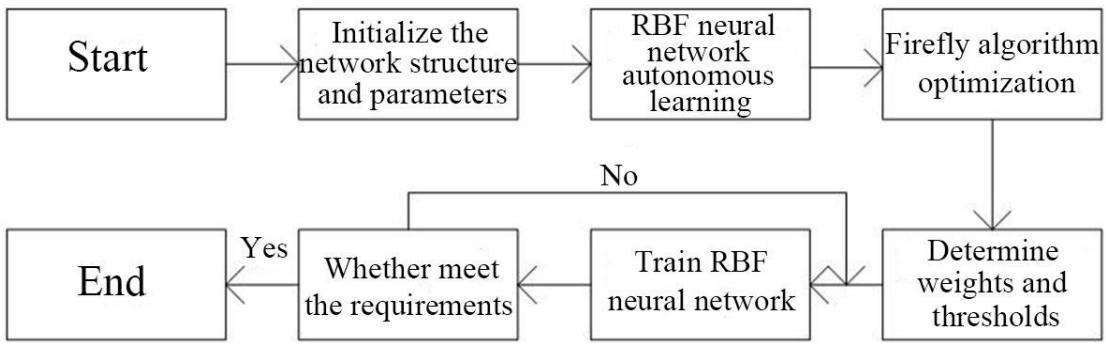


Figure 2: Flow chart of firefly neural network.

position fitness value of firefly *i* at *t*.

④ The neighbor fireflies set formula is $N_i(t) = \{ j : \| X_j(t) - X_i(t) \| < r_i(t); l_i(t) < l_j(t) \}$, where $N_i(t)$ stands for the neighbor fireflies set of fireflies i at t, $\| \bullet \|$ stands for the Euclidean distance, and $r_i(t)$ stands for the decision radius of fireflies *i* at *t*. The probability that firefly *i* chooses j as neighbor firefly is $P_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{k \in N_i(t)} l_k(t) - l_i(t)}$, the position updating formula is $X_i(t+1) = X_i(t) + s(\frac{X_j(t) - Xi(t)}{\| X_j(t) - Xi(t) \|}$, and the updating formula of decision radius is $r_i(t) = \min\{rs, \max[0, r_i(t) + \beta(nt - |N_i(t)|]\}$, where $r_i(t)$ stands for the perceived range of firefly i at t, $0 < r_i(t) < rs$, and $|N_i(t)|$ stands for the size of neighbor set.

⑤ After the iteration, it is judged whether the iteration number reaches the maximum. If it does, the algorithm is finished and the optimal value is recorded; if not, the iteration is continued.

(3) The above data are used as training samples for neural network testing and training.

4 Example analysis of defect workpiece

In order to verify the correctness of the method in this study, defect recognition was carried out on 100 industrial CT images of solid rocket engine model which was in a size of 512 × 512. The material of the motor body was 30GrMnSiA, the length of the motor grain was 1000 mm, and the external diameter was 150 mm. The artificial detection results were 80 defective images, and 20 non-defective images and 142 defects).

4.1 Defect localization results

The defects of the industrial CT images were positioned. The size of block was a quarter. The calculation results of fractal dimension are shown in Figure 3.

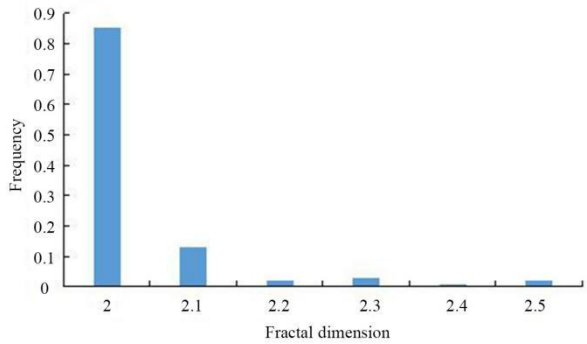


Figure 3: The frequency distribution of fractal dimension.

Fractal dimension has a large influence on the accuracy of defect marks, it can be noted from Figure 3 that image defects could be clearly positioned when the fractal dimension was between 2 and 2.1 and the frequency was 0.85. Therefore, the fractal dimension was set as 2.1. The defect localization results are shown in Figure 4.

As shown in Figure 3, the algorithm used in this study can locate the defect area of the workpiece accurately and facilitate the subsequent defect detection.

4.2 Defect segmentation results

One hundred industrial CT images (80 defective images and 20 non-defective images) were processed by the improved k-mean clustering algorithm, and the results obtained were compared with the results of manual judgment, as shown in Table 2.

As can be seen from the Table 2, the algorithm successfully segmented 78 defective images, only one non-defective image was wrongly segmented, and the overall segmentation accuracy rate was relatively high, indicating that the proposed segmentation algorithm was highly reliable.

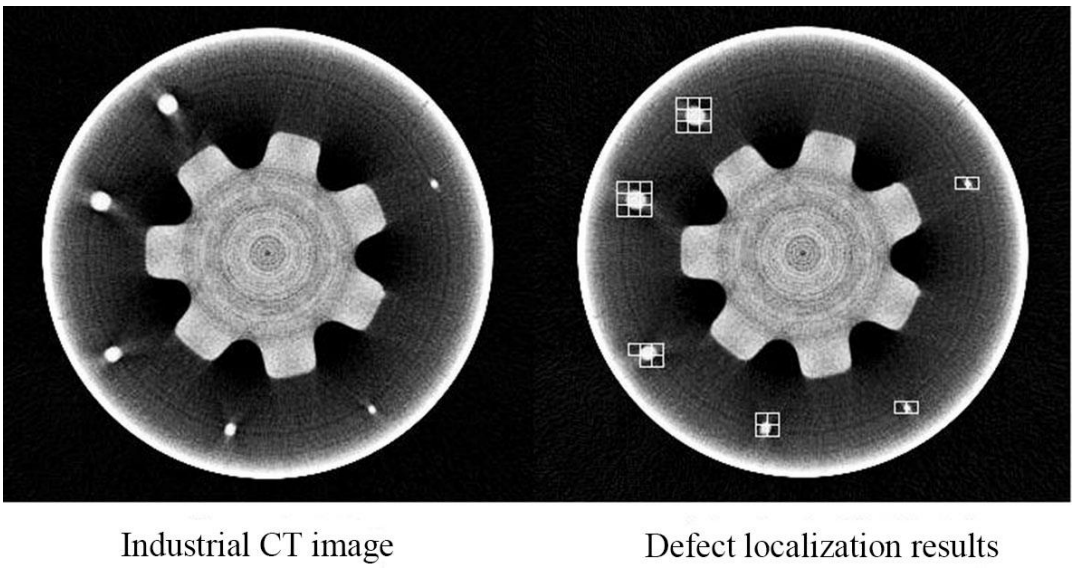


Figure 4: Defect localization results.

	Defective images	Non-defective images
Number of segmented defects	78	1
Number of unsegmented defects	2	19
Accuracy rate	97.5%	95%

Table 2: Defect segmentation results.

4.3 Feature extraction results

The feature extraction of defect images was carried out by the method of abstract invariant moments. Taking the stomata as an example, its feature quantity is shown in Table 3.

	Original image	Translated Image	Image which is clockwise rotated for 90 degrees
R1	0.315687	0.314256	0.312456
R2	1.935621	1.935124	1.935214
R3	4.500254	4.502103	4.505321
R4	3.785214	3.782158	3.780215
R5	2.124521	2.125632	2.120325
R6	0.234665	0.239654	0.236589
R7	0.621453	0.621036	0.625879
R8	0.235462	0.231456	0.236587
R9	0.625471	0.620852	0.623168
R10	0.442123	0.441258	0.446852

Table 3: Defect feature value of stomata.

These 10 feature values were extracted, plus two shape features and two gray features, a total of 14 feature vectors were obtained.

4.4 Feature recognition results

The theoretical output value and actual output value of 10 defects identified in 8 pictures are shown in Table 4.

The theoretical output values of the neural network should be 001 (crack), 010 (stomata) and 100 (slag inclusion), but there always exists error in the actual output. Therefore, the error was controlled to 0.2, and the actual output less than 0.2 was rounded to 0, while larger than 0.8 was rounded to 1. Only the recognition of A7 was wrong in the ten defects of the Table 4. The 142 defects in the processed 100 CT images were recognized, and 139 defects were correctly recognized, and 3 defects were misjudged. The accuracy rate was 97.89%, which indicated that the defect recognition method in this study had high reliability.

No.	The theoretical output			The actual output		
A1	0	0	1	0.02132	0.01253	0.91021
A2	0	0	1	0.01023	0.02154	0.92521
A3	1	0	0	0.89652	0.10235	0.02365
A4	0	1	0	0.02158	0.95213	0.02157
A5	0	0	1	0.01245	0.08521	0.94587
A6	0	1	0	0.01852	0.95210	0.01658
A7	1	0	0	0.89658	0.42011	0.02856
A8	0	1	0	0.02145	0.90258	0.01589
A9	0	0	1	0.12035	0.02157	0.96324
A10	1	0	0	0.95462	0.02145	0.01856

Table 4: The comparison between the theoretical output and the actual output.

5 Discussion and conclusion

The internal defects of a workpiece can greatly affect the practicability and safety of the workpiece. With the emphasis on the workpiece quality, the internal defect detection technology of the workpiece has been developed. Common internal defect detection technologies of the workpiece include ultrasonic, laser holography, X-ray photography, etc. Industrial CT images are currently the most effective non-destructive testing technology [17], making it easier to identify defects. Defect recognition based on industrial CT images is a simple and efficient method.

Before defect recognition, it is necessary to locate and segment defects in the image. Defect localization can obtain the location information of defects from CT images, including the method of fractal, Gabor wavelet, statistics, etc. Yang et al [18] proposed a localization method based on cubature Kalman smooth filter, which can effectively locate defects. In this study, the block fractal algorithm was selected for image localization, and an industrial CT image was taken as an example. It was found that the algorithm can successfully locate the defect part.

The improved k-means clustering algorithm was selected to segment the image. The experiments showed the accuracy of the algorithm more than 95%, which provided a good foundation for the following defect feature extraction and recognition.

Hu invariant moment theory was adopted in this study for defect feature extraction. The abstraction of Hu invariant moment can make it better to extract features. Then, ten feature vectors can be obtained, plus the two shape features and the two gray features equal the total of 14 feature vectors which were used in feature recognition. For feature recognition there exist numerous methods such as artificial neural networks (ANN), support vector machine (SVM), principal component analysis (PCA) and other. In this study, the RBF neural network was selected to recognize the features, and then the weights and the threshold of the neural network were optimized by using the firefly algorithm. The accuracy rate of 97.89% was obtained in the experiments, which proved the reliability of the defect recognition algorithm in this study.

Industrial CT image is one of the effective methods for non-destructive testing of workpiece. In this study,

based on the industrial CT image, the image defects were located by the block fractal algorithm, then the improved k-means clustering algorithm was used to segment the defect image, the abstracted Hu invariant moment algorithm was adopted for feature extraction, and finally the firefly algorithm and the RBF neural network were used for feature recognition.

6 Acknowledgement

We completed this paper based on the project of contours extraction from industrial computed tomography images supported by Chongqing Municipal Education Committee.

7 References

[1] Alimohamadi H, Ahmadyfard A, Shojaee E. (2010). Defect Detection in Textiles Using Morphological Analysis of Optimal Gabor Wavelet Filter Response. International Conference on Computer and Automation Engineering, IEEE, Bangkok, Thailand, pp. 26-30. https://doi.org/10.1109/ICCAE.2009.43

[2] Chen Z, Wang C, Hu Z, Xie S. (2010). Dual-tree complex wavelet analysis and its application in defect detection of workpiece for cross wedge rolling. International Conference on Advanced Computer Theory and Engineering, IEEE, Chengdu, China, pp. 861-72. https://doi.org/10.1109/ICACTE.2010.5579227

[3] Sabeenian RS, Paramasivam ME. (2010). Defect detection and identification in textile fabrics using Multi Resolution Combined Statistical and Spatial Frequency Method. Advance Computing Conference, IEEE, Patiala, India, pp. 162-166. https://doi.org/10.1109/IADCC.2010.5423017

[4] Liu BT, Hou DB, Liu BL, Zhao L, Huang PJ, Zhang GX. (2014). Defect detection and identification in eddy current testing using subtractive clustering algorithm combined with RBFNN. Insight - Non-Destructive Testing and Condition Monitoring, 56(7), pp. 375-380(6). https://doi.org/10.1784/insi.2014.56.7.375

[5] Leng Y, Xiao Z, Geng L, Xi J. (2018). Defect detection and classification of galvanized stamping parts based on fully convolution neural network. International Conference on Graphic and Image Processing, pp. 188. https://doi.org/10.4028/www.scientific.net/AMR.675.55

[6] Zhang RF. (2013). Study on CT Image Reconstruction Applications in Industry. Advanced Materials Research, pp. 55-58. https://doi.org/10.4028/www.scientific.net/AMR.675.55

[7] Carmignato S. (2012). Accuracy of industrial computed tomography measurements: Experimental results from an international comparison. CIRP Annals - Manufacturing Technology, 61(1), pp. 491-494. https://doi.org/10.1016/j.cirp.2012.03.021

[8] Chen L. (2011). The Application and Investigation about Industry CT Scan Technology in the Measure and Design about Complex Box. Applied Mechanics & Materials, pp. 319-322. https://doi.org/10.4028/www.scientific.net/AMM.86.319

[9] Samarawickrama Y C, Wickramasinghe C D. (2017). Matlab based automated surface defect detection system for ceremic tiles using image processing. Technology and Management, IEEE, 34-39. https://doi.org/10.1109/NCTM.2017.7872824

[10] Jiang SQ, Luan CB, Man YE, Zhao YY. (2017). Application of Industry CT in Large Complicated Casing Inspection. Nondestructive Testing, 39(2), pp. 18-21. https://doi.org/10.1109/NSSMIC.2011.6154616

[11] Chang M, Xiao Y, Chen Z, Li L. (2011). Preliminary study of rotary motion blurs in a novel industry CT imaging system. Nuclear Science Symposium and Medical Imaging Conference. IEEE, Valencia, Spain, pp. 1358-1361. https://doi.org/10.1109/NSSMIC.2011.6154616

[12] Zabler S, Fella C, Dietrich A, Nachtrab F, Salamon M, Voland V, Ebensperger T, Oeckl S, Hanke R, Uhlmann N. (2012). High-resolution and high-speed CT in industry and research. SPIE Optical Engineering + Applications, San Diego, California, United States, pp. 850617. https://doi.org/10.1117/12.964588

[13] Mandelbrot BB, Wheeler JA. (1998). The Fractal Geometry of Nature. American Journal of Physics, 51(4), pp. 468 p. https://doi.org/10.1119/1.13295

[14] Normand MD, Peleg M. (1988). Evaluation of the 'blanket' algorithm for ruggedness assessment. Powder Technology, 54(4), pp. 255-259. https://doi.org/10.1016/0032-5910(88)80055-X

[15] Tang YH, Gong A, Wang C. (2012). Automatic Generation Algorithms Based on Optimization Initial Population. Computer & Modernization, 562(1), pp. 131–147.

[16] Yan BJ, Zheng L, Wang KY. (2001). Fast Target-Detecting Algorithm Based on Invariant Moment. Infrared Technology, 23(6), pp. 8-12.

[17] Xie L, Huang R, Gu N, Cao Z. (2014). A novel defect detection and identification method in optical inspection. Neural Computing & Applications, 24(7-8), pp. 1953-1962. https://doi.org/10.1007/s00521-013-1442-7

[18] Yang L, Li H, Zhou F, Jin P. (2015). The Pipeline Defect Location Technology Based on Cubature Kalman Smooth Filter. Chinese Journal of Sensors & Actuators, 28(4), pp. 591-597. https://doi.org/10.3969/j.issn.1004-1699.2015.04.023

JOŽEF STEFAN INSTITUTE

Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of S**love**nia (or S♡nia). The capital today is considered a crossroad between East, West and Mediterranean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park "Ljubljana" has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park "Ljubljana". The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
Tel.:+386 1 4773 900, Fax.:+386 1 251 93 85
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Public relations: Polona Strnad

INFORMATICA

AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

INVITATION, COOPERATION

Submissions and Refereeing

Please register as an author and submit a manuscript at: http://www.informatica.si. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica LaTeX format and figures in .eps format. Style and examples of papers can be obtained from http://www.informatica.si. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

SUBSCRIPTION

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: drago.torkar@ijs.si

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than twentyfour years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica web edition is free of charge and accessible at http://www.informatica.si.
Informatica print edition is free of charge for major scientific, educational and governmental institutions. Others should subscribe.

Informatica WWW:

http://www.informatica.si/

Referees from 2008 on:

A. Abraham, S. Abraham, R. Accornero, A. Adhikari, R. Ahmad, G. Alvarez, N. Anciaux, R. Arora, I. Awan, J. Azimi, C. Badica, Z. Balogh, S. Banerjee, G. Barbier, A. Baruzzo, B. Batagelj, T. Beaubouef, N. Beaulieu, M. ter Beek, P. Bellavista, K. Bilal, S. Bishop, J. Bodlaj, M. Bohanec, D. Bolme, Z. Bonikowski, B. Bošković, M. Botta, P. Brazdil, J. Brest, J. Brichau, A. Brodnik, D. Brown, I. Bruha, M. Bruynooghe, W. Buntine, D.D. Burdescu, J. Buys, X. Cai, Y. Cai, J.C. Cano, T. Cao, J.-V. Capella-Hernández, N. Carver, M. Cavazza, R. Ceylan, A. Chebotko, I. Chekalov, J. Chen, L.-M. Cheng, G. Chiola, Y.-C. Chiou, I. Chorbev, S.R. Choudhary, S.S.M. Chow, K.R. Chowdhury, V. Christlein, W. Chu, L. Chung, M. Ciglarič, J.-N. Colin, V. Cortellessa, J. Cui, P. Cui, Z. Cui, D. Cutting, A. Cuzzocrea, V. Cvjetkovic, J. Cypryjanski, L. Čehovin, D. Čerepnalkoski, I. Čosić, G. Daniele, G. Danoy, M. Dash, S. Datt, A. Datta, M.-Y. Day, F. Debili, C.J. Debono, J. Dedič, P. Degano, A. Dekdouk, H. Demirel, B. Demoen, S. Dendamrongvit, T. Deng, A. Derezinska, J. Dezert, G. Dias, I. Dimitrovski, S. Dobrišek, Q. Dou, J. Doumen, E. Dovgan, B. Dragovich, D. Drajic, O. Drbohlav, M. Drole, J. Dujmović, O. Ebers, J. Eder, S. Elaluf-Calderwood, E. Engström, U. riza Erturk, A. Farago, C. Fei, L. Feng, Y.X. Feng, B. Filipič, I. Fister, I. Fister Jr., D. Fišer, A. Flores, V.A. Fomichov, S. Forli, A. Freitas, J. Fridrich, S. Friedman, C. Fu, X. Fu, T. Fujimoto, G. Fung, S. Gabrielli, D. Galindo, A. Gambarara, M. Gams, M. Ganzha, J. Garbajosa, R. Gennari, G. Georgeson, N. Gligorić, S. Goel, G.H. Gonnet, D.S. Goodsell, S. Gordillo, J. Gore, M. Grčar, M. Grgurović, D. Grosse, Z.-H. Guan, D. Gubiani, M. Guid, C. Guo, B. Gupta, M. Gusev, M. Hahsler, Z. Haiping, A. Hameed, C. Hamzaçebi, Q.-L. Han, H. Hanping, T. Härder, J.N. Hatzopoulos, S. Hazelhurst, K. Hempstalk, J.M.G. Hidalgo, J. Hodgson, M. Holbl, M.P. Hong, G. Howells, M. Hu, J. Hyvärinen, D. Ienco, B. Ionescu, R. Irfan, N. Jaisankar, D. Jakobović, K. Jassem, I. Jawhar, Y. Jia, T. Jin, I. Jureta, Đ. Juričić, S. K, S. Kalajdziski, Y. Kalantidis, B. Kaluža, D. Kanellopoulos, R. Kapoor, D. Karapetyan, A. Kassler, D.S. Katz, A. Kaveh, S.U. Khan, M. Khattak, V. Khomenko, E.S. Khorasani, I. Kitanovski, D. Kocev, J. Kocijan, J. Kollár, A. Kontostathis, P. Korošec, A. Koschmider, D. Košir, J. Kovač, A. Krajnc, M. Krevs, J. Krogstie, P. Krsek, M. Kubat, M. Kukar, A. Kulis, A.P.S. Kumar, H. Kwaśnicka, W.K. Lai, C.-S. Laih, K.-Y. Lam, N. Landwehr, J. Lanir, A. Lavrov, M. Layouni, G. Leban, A. Lee, Y.-C. Lee, U. Legat, A. Leonardis, G. Li, G.-Z. Li, J. Li, X. Li, X. Li, Y. Li, Y. Li, S. Lian, L. Liao, C. Lim, J.-C. Lin, H. Liu, J. Liu, P. Liu, X. Liu, X. Liu, F. Logist, S. Loskovska, H. Lu, Z. Lu, X. Luo, M. Luštrek, I.V. Lyustig, S.A. Madani, M. Mahoney, S.U.R. Malik, Y. Marinakis, D. Marinčič, J. Marques-Silva, A. Martin, D. Marwede, M. Matijašević, T. Matsui, L. McMillan, A. McPherson, A. McPherson, Z. Meng, M.C. Mihaescu, V. Milea, N. Min-Allah, E. Minisci, V. Mišić, A.-H. Mogos, P. Mohapatra, D.D. Monica, A. Montanari, A. Moroni, J. Mosegaard, M. Moškon, L. de M. Mourelle, H. Moustafa, M. Možina, M. Mrak, Y. Mu, J. Mula, D. Nagamalai, M. Di Natale, A. Navarra, P. Navrat, N. Nedjah, R. Nejabati, W. Ng, Z. Ni, E.S. Nielsen, O. Nouali, F. Novak, B. Novikov, P. Nurmi, D. Obrul, B. Oliboni, X. Pan, M. Pančur, W. Pang, G. Papa, M. Paprzycki, M. Paralič, B.-K. Park, P. Patel, T.B. Pedersen, Z. Peng, R.G. Pensa, J. Perš, D. Petcu, B. Petelin, M. Petkovšek, D. Pevec, M. Pičulin, R. Piltaver, E. Pirogova, V. Podpečan, M. Polo, V. Pomponiu, E. Popescu, D. Poshyvanyk, B. Potočnik, R.J. Povinelli, S.R.M. Prasanna, K. Pripužić, G. Puppis, H. Qian, Y. Qian, L. Qiao, C. Qin, J. Que, J.-J. Quisquater, C. Rafe, S. Rahimi, V. Rajkovič, D. Raković, J. Ramaekers, J. Ramon, R. Ravnik, Y. Reddy, W. Reimche, H. Rezankova, D. Rispoli, B. Ristevski, B. Robič, J.A. Rodriguez-Aguilar, P. Rohatgi, W. Rossak, I. Rožanc, J. Rupnik, S.B. Sadkhan, K. Saeed, M. Saeki, K.S.M. Sahari, C. Sakharwade, E. Sakkopoulos, P. Sala, M.H. Samadzadeh, J.S. Sandhu, P. Scaglioso, V. Schau, W. Schempp, J. Seberry, A. Senanayake, M. Senobari, T.C. Seong, S. Shamala, c. shi, Z. Shi, L. Shiguo, N. Shilov, Z.-E.H. Slimane, F. Smith, H. Sneed, P. Sokolowski, T. Song, A. Soppera, A. Sorniotti, M. Stajdohar, L. Stanescu, D. Strnad, X. Sun, L. Šajn, R. Šenkeřík, M.R. Šikonja, J. Šilc, I. Škrjanc, T. Štajner, B. Šter, V. Štruc, H. Takizawa, C. Talcott, N. Tomasev, D. Torkar, S. Torrente, M. Trampuš, C. Tranoris, K. Trojacanec, M. Tschierschke, F. De Turck, J. Twycross, N. Tziritas, W. Vanhoof, P. Vateekul, L.A. Vese, A. Visconti, B. Vlaovič, V. Vojisavljević, M. Vozalis, P. Vračar, V. Vranić, C.-H. Wang, H. Wang, H. Wang, H. Wang, S. Wang, X.-F. Wang, X. Wang, Y. Wang, A. Wasilewska, S. Wenzel, V. Wickramasinghe, J. Wong, S. Wrobel, K. Wrona, B. Wu, L. Xiang, Y. Xiang, D. Xiao, F. Xie, L. Xie, Z. Xing, H. Yang, X. Yang, N.Y. Yen, C. Yong-Sheng, J.J. You, G. Yu, X. Zabulis, A. Zainal, A. Zamuda, M. Zand, Z. Zhang, Z. Zhao, D. Zheng, J. Zheng, X. Zheng, Z.-H. Zhou, F. Zhuang, A. Zimmermann, M.J. Zuo, B. Zupan, M. Zuqiang, B. Žalik, J. Žižka,

Informatica

An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: http://www.informatica.si.

Subscription Information Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.
The subscription rate for 2018 (Volume 42) is
– 60 EUR for institutions,
– 30 EUR for individuals, and
– 15 EUR for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar.
Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email (drago.torkar@ijs.si), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):
Slovene Society for Pattern Recognition (Vitomir Štruc)
Slovenian Artificial Intelligence Society (Mitja Luštrek)
Cognitive Science Society (Olga Markič)
Slovenian Society of Mathematicians, Physicists and Astronomers (Marej Brešar)
Automatic Control Society of Slovenia (Nenad Muškinja)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Mark Pleško)
ACM Slovenia (Borut Žalik)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math

Volume 42 Number 3 September 2018 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **42** (2018) Number 3, pp. 285–483