Volume 40 Number 4 December 2016 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Special Issue:

Applications in Information Technology

Guest Editors:

Vitaly Klyuev

Evgeny Pyshkin

Alexander Vazhenin

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.

Editors' Introduction to the Special Issue on "Applications in Information Technology"

Towards Better Human-Centric Solutions in Information Technology Applications

This special issue includes five revised and extended papers presented at the *2nd Conference on Applications in Information Technology* (ICAIT-2016) held in Aizu-Wakamatsu (Japan) between October 6th to 8th, 2016. This conference was organized by the University of Aizu in collaboration with international academic partners from Peter the Great Saint-Petersburg Polytechnic University, Saint-Petersburg State University, and Novosibirsk State University. The primary objective of the conference was to foster rich creativity of students and young scientists and to encourage them to participate actively in open discussions with their colleagues, advancing the design, development, use, and evaluation of information technology applications from many respected research institutions all around the world.

ICAIT-2016 accumulated good traditions established in the past conferences organized by the University of Aizu including *The Conference on Humans and Computers* in 1998-2010, its successor, *The 2012 Joint International Conference on Human-Centered Computer Environments* and *The 2015 International Workshop on Applications in Information Technology*. After event completion we are assured that the conference meetings and discussions facilitated significantly numerous partnership research activities. As guest editors, we particularly thank *Informatica* journal for giving us an excellent opportunity to publish the extended versions of the distinguished conference papers in the Special Issue on Applications in Information Technology.

For this issue we selected a number of works which pay particular attention to advancing human-centric solutions in the huge domain of information technology applications. Our authors address such important areas as developing approaches for computer-assisted learning in mathematics, data acquisition and visualization for the purpose of human emotion analysis, improving models and algorithms used in urban computing, enhancing quality of current recommendation systems, as well as modeling, design and verification of power efficient hardware systems. All the presented works are in strong connection to the current trends in developing better social and technology environments used in a wide range of present day human-centric systems.

The authors of the article "Mathematical Equation Structural Syntactical Similarity Patterns: A Tree Overlapping Algorithm and Its Evaluation" (Evgeny Pyshkin and Mikhail Ponomarev) investigate possible improvements of tree overlapping algorithms for the case of mathematical equations structural syntactical similarity. The work addresses the case of using such algorithms for educational purposes, particularly, for finding appropriate test and exam preparation exercises. The article "Analysis of Emotions in Real-time Twitter Streams" (Yuki Kobayashi, Myriam Munezero, and Maxim Mozgovoy) presents a system for visualizing discussions and emotions of Twitter users in real time over a specific geographical location. The authors describe the existing implementation by using a couple of examples illustrating the process of exploring and comparing ongoing discussions. The article "OD-Matrix Estimation based on a Dual Formulation of Traffic Assignment Problem" (Alexander Yu. Krylatov, Anastasiia P. Shirokolobova, and Victor V. Zakharov) is devoted to problems of improving traffic assignment algorithms used in modern urban computing systems. Particular attention is paid to the process of decision making while processing big volumes of transportation data describing travel demands between travel origins and destinations. The focus of the article "Design of an Asynchronous Processor with Bundled-Data Implementation on a Commercial Field Programmable Gate Array" is on improving design and modeling tools for developing hardware systems using asynchronous control circuits. The authors specifically address the aspects of increasing computational performance with making significant efforts in reducing energy consumption which is one of urgent needs in advancing smart and environment friendly systems. Finally, the authors of the paper "Performance Comparison of Featured Neural Network Trained with Backpropagation and Delta Rule Techniques for Movie Rating Prediction in Multi-Criteria Recommender Systems" (Mohammed Hassan and Mohamed Hamada) examine current trends in the area of recommendation systems and describe a multi-criteria recommendation technique using feedforward neural network for model user preference modeling.

We are pleased to acknowledge the great efforts of ICAIT-2016 organizers. We would like specially mention the conference honorary chairs Prof. Ryuichi Oka, President of the University of Aizu, Prof. Leon Petrosjan, Dean of the Faculty of Applied mathematics – Control Processes of Saint-Petersburg State University, Prof. Vladimir Zaborovsky, Director of the Institute of Computer Science and Technology of Saint-Petersburg Polytechnic University, and Prof. Mikhail M. Lavrentiev, Dean of the Faculty of Information Technologies of Novosibirsk State University.

We would like to express our great appreciation to the program committee members who reviewed the conference submissions and ensured high quality of published papers. Among them we thank Thomas Baar (Hochschule für Technik und Wirtschaft Berlin, Germany), Natalia Bogach (Peter the Great Saint-Petersburg Polytechnic University, Russia), Paolo Bottoni (University of Rome, Italy), John Brine (University of Aizu, Japan), Alfredo Capozucca (University of Luxembourg, Luxembourg), Hapugahage

Thilak Chaminda (Informatics Institute of Technology, Sri Lanka), Ruth Patricia Cortez (Simulatio, Japan), Vlatko Davidovski (Cognizant Business Consulting, Switzerland), Vladimir Dobrynin (Saint-Petersburg State University, Russia), Mikhail Glukhikh (JetBrains, Peter the Great Saint-Petersburg Polytechnic University, Russia), Nicolas Guelfi (University of Luxembourg, Luxembourg), Mohamed Hamada (University of Aizu, Japan), Yannis Haralambous (Institut Mines-Télécom, Télécom Bretagne, France), Houcine Hassan (Universitat Politècnica de València, Spain), Vladimir Itsykson (Peter the Great Saint-Petersburg Polytechnic University, Russia), Qun Jin (Waseda University, Japan), Sergei Krutolevich (Belarusian Russian University, Belarus), Andrey Kuznetsov (Motorola Solutions Inc., Russia), Petri Laitinen (Karelia University of Applied Sciences, Finland), Ruediger Lunde (Ulm University of Applied Sciences, Germany), Viacheslav Marakhovsky (Peter the Great Saint- Petersburg Polytechnic University, Russia), Calkin Suero Montero (University of Eastern Finland, Finland), Maxim Mozgovoy (University of Aizu, Japan), Kendall Nygard (North Dakota State University, USA), Kohei Ohno (Meiji University, Japan), Mikhail Okrepilov (Peter the Great Saint-Petersburg Polytechnic University, Russia), Vladimir Oleshchuk (University of Adger, Norway), Benoît Ries (University of Luxembourg, Luxembourg), Alexey Romanenko (Novosibirsk State University, Russia), Hiroshi Saito (University of Aizu, Japan), Roman Shtykh (CyberAgent Inc, Japan), Nikolay Smirnov (Saint-Petersburg State University, Russia), Kari Smolander (Aalto University, Finland), Senzhang Wang (Nanjing University of Aeronautics and Astronautics, China), Ying-Hong Wang, (Tamkang University, Taiwan, China), Yang Zhibin (Nanjing University of Aeronautics and Astronautics, China).

We also thank Prof. Matjaz Gams (managing editor of *Informatica*) for his support and very valuable comments which helped to improve this special issue significantly.

This year, we are organizing a special session on Information Management in Human-Centric Systems (IMHCS'17) as an event of the *3rd IEEE International Conference on Cybernetics* (IEEE CYBCONF 2017) to be held in Exeter, UK on June 21 – 23, 2017. We expect to continue our efforts in organizing high quality discussions of current trends in information acquisition, representation and processing in human-centric systems and applications.

Evgeny Pyshkin

Vitaly Klyuev

Alexander Vazhenin

Mathematical Equation Structural Syntactical Similarity Patterns: A Tree Overlapping Algorithm and Its Evaluation

Evgeny Pyshkin
University of Aizu, Japan
Tsuruga, Ikki-machi, Aizu-Wakamatsu
Fukushima, 965-8580 Japan
E-mail: pyshe@u-aizu.ac.jp

Mikhail Ponomarev
Peter the Great St. Petersburg Polytechnic University
29 Polytechnicheskaya st., 195251 St. Petersburg, Russia
E-mail: ponmike92@gmail.com

Keywords: syntactical similarity, mathematical equations, MathML, tree overlapping

Received: November 14, 2016

In this paper we examine mathematical equations structural syntactical similarity patterns. The major focus of this contribution is an NLP tree overlapping algorithm modification adopted to the case of syntactical similarity of mathematical equations presented in MathML. We describe the software implementation and the tests arranged for the cases of both structural and subexpression based similarity. The paper also contains a discussion of algorithm evaluation problems conditioned by the lack of relevant syntactical similarity centered equation corpora.

Povzetek: Prispevek se ukvarja s strukturnimi sinkaktičnimi vzorci podobnosti matematičnih izrazov.

1 Introduction

Nowadays there are only few models of adopting natural language processing (NLP) algorithms related to syntax similarity to mathematical notations. Unique structural syntax of mathematical equations with a big variety of semantically equivalent constructions provide a non-trivial case for information retrieval [11]. Many reported implementations are focused on finding exact matching of mathematical constructions rather than on recognizing their similarity [9, 8]. Indeed, for a case of mathematical equations, syntactical similarity is defined rather fuzzy by using several structural syntactical similarity patterns. However, a model that would deal with syntactical similarity seems to be very useful while developing searching and classification tools used in education, so as to allow math learners and tutors selecting suitable tasks nailing down a topic presented during a classroom session. An obvious possible use case is accessing a set of relevant mathematical equations to be used for training while a learner is doing the preparation exercises for an examination. Another interesting possibility is searching an equation by its syntactical structure, the latter being often easier to recall compare to exact mathematical formulas.

For the reason that most structural notations used for representing mathematical expressions are in fact based on directed graphs, the syntactical similarity can be defined by using tree structural similarity. Specifically, this work addresses the case when expressions are uniformly presented in MathML, the latter being one of widely used structural XML based notations used in mathematics. In turn, if better structural math equation forms are used, one can expect more efficient and accurate retrieval, in contrast to a frequent use of image based equation representation used on many web sites. At the same time we accept a possible criticism pointing an issue that not an every mathematical expression retrieval difficulty could be addressed under MathML representability constraints.

Within a context of mathematical equations similarity, it is important to mention both meaning related similarity and presentation related similarity (in MathML there are two markup schemes corresponding to these two perspectives: content markup and presentation markup). Semantic (e.g. equation meaning) similarity is out of scope of this study; this work is focused only on presentation similarity which is related to the equation syntax (the form) rather than to its meaning (the contents).

This paper is based on Mikhail Ponomarev and Evgeny Pyshkin, *Adopting Tree Overlapping Algorithm for MathML Equation Structural Similarity*, published in the Proceedings of the 2nd International Conference on Applications in Information Technology (ICAIT-2016) [7]

MathML – Mathematical Markup Language

2 Structural similarity of mathematical equations

Since the structure of mathematical equations can be represented in the form of a tree, mathematical equation similarity may be defined using tree similarity. In [1] similarity of two trees is defined on the base of recursive examination of their subtrees. In [5] the following mathematical expressions similarity patterns are defined:

Mathematical equivalence: Equations E_1 and E_2 are mathematically equivalent if they are semantically (but not obligatorily syntactically) the same, for example $\frac{d(sin(x))}{dx}$ and $(sin(x))'$, $sin^2(x) + cos^2(x)$ and 1 are correspondingly equivalent.

Identity: E_1 and E_2 are identical if they are exactly the same.

Syntactical identity: E_1 and E_2 are syntactically identical if they are identical after normalization (dealing with variable names and numeric values). For example $sin(a)$ and $sin(b)$, $\frac{1}{sin(x)}$ and $\frac{5}{sin(x)}$ are correspondingly syntactically identical.

N–similarity: Normalized equations E_1 and E_2 are n-similar if there is a similarity (in a certain sense) which is $sim(E_1, E_2) \geqslant n$, n being a parametric value determining a threshold. There are two specific N–similarity cases:

1. **Subexpression n–similarity**: There is a subexpression n–similarity for E_1 and E_2, if E_1 and E_2 are n–similar and the corresponding trees both contain the common subtree which in turn contains all the terminal nodes of both trees. Figure 1 shows an example for the case of expressions $sin(x)^2$ and $\frac{sin(x)}{2}$.

2. **Structural n–similarity**: E_1 and E_2 are structurally n–similar if E_1 and E_2 are n–similar (in common sense) and there is a common part in both trees rooted at root nodes of compared trees with the production rules being the same for all the nodes in this part. Figure 2 illustrates this case for the equations $x + \sqrt{sin(a)}$ and $x + \sqrt{2b}$.

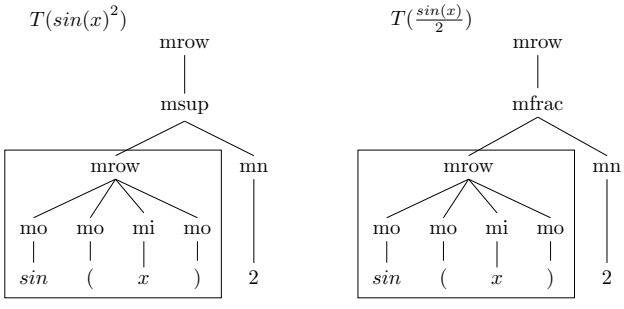


Figure 1: Equations $sin(x)^2$ and $\frac{sin(x)}{2}$ are structurally n–similar for any value of $n \geqslant \frac{18}{26}$

Note that in order to represent n-values, in Figures 1 and 2 we use the nodes ratio which is a ratio of the number of common nodes in both trees to the number of all nodes in both trees.

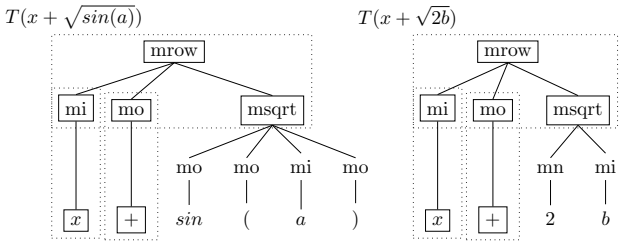


Figure 2: $x + \sqrt{sin(a)}$ and $x + \sqrt{2b}$ are structurally n–similar for any value of $n \geqslant \frac{12}{24}$

3 Equation similarity evaluation using tree similarity

In order to introduce different approaches used for evaluating mathematical equation similarity based on tree similarity, in this section some sample trees are used: T_0, T_1 and T_2 (shown in Figure 3). In the following text we demonstrate how the similarity between T_0 and T_1 as well as between T_0 and T_2 respectively can be calculated.

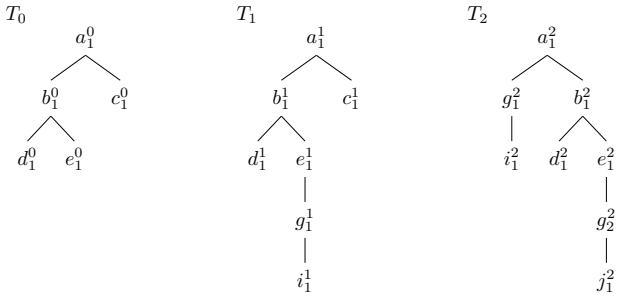


Figure 3: Sample trees T_0, T_1 and T_2

For each node in Figure 3 its upper index corresponds to the tree which this node belongs to. Thus, all the nodes of T_0 have the upper indexes which are equal to 0, the nodes of T_1 have the upper indexes which are equal to 1, and so on. The lower indexes are used so as to count the equal nodes within a tree. For instance, T_2 contains two equal nodes g, so the first appearance of g is marked by the lower index 1, while the second appearance of g is marked by the lower index 2. These indexes are suppressed in cases, when they are not necessary for algorithm description.

3.1 Tree edit distance

Tree edit distance method uses the definition of similarity (distance) between two trees as a weighted number of edit operations (insert, delete, and modify) required to transform one tree to another (as described, for example, in [10]).

Assume S is a sequence of edit operations $\{s_1, s_2, \ldots, s_k\}$ for transforming one tree to another. Assume γ is a non-negative distance measure describing node transformation from a to b (defined here as $a \to b$) such as $\gamma(a \to b) \geqslant 0$ and $\gamma(a \to b) = \gamma(b \to a)$.

For an operation sequence S we get the following sum: $\gamma(S) = \sum_{i=1}^{|S|} \gamma(s_i)$.

Then the distance between two equations is defined as follows:

$$\delta(T_1, T_2) = min\{\gamma(S)\} \tag{1}$$

Figures 4 and 5 illustrate how the transformation distances from $T0$ to $T1$ and from $T0$ to $T2$ correspondingly are calculated: a sequence of two insertions is required in order to transform T_0 to T_1; four operations (one deletion and three insertions) are required in order to transform T_0 to T_2.

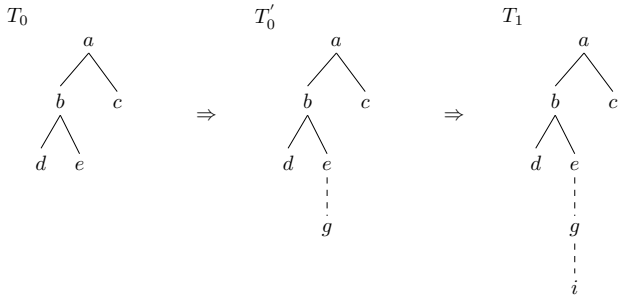


Figure 4: Tree edit transformation: T_0 to T_1

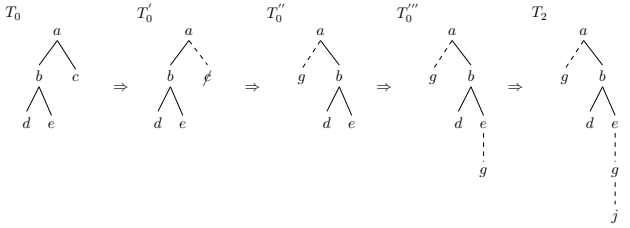


Figure 5: Tree edit transformation: T_0 to T_2

The node weights (and the operation costs as well) might not be equal, hence there might be different similarity measures based on general edit distance schema. Also, in different algorithms a sequence S_i is searched differently: in addition to the earlier mentioned work [10] there are other implementations described in [2] and [6]. Algorithmic complexity of the above mentioned approaches is summarized in Table 1.

3.2 Subpath set

Subpath set similarity between two trees is defined as the number of subpaths shared by the trees. Given a tree, its subpaths are defined as a set of all the paths from the root node to the leaves including the partial paths. Subpath

Table 1: Tree edit distance based algorithms

Algorithm	Time	Memory	Particularities
TED [10]	$O(n^4)$	$O(n^2)$	Good for balanced trees
ODTED [2]	$O(n^3)$	$O(n^2)$	Better in unbalanced trees
RTED [6]	$O(n^3)$	$O(n^2)$	Tree balance insensitive

based similarity definition for a case of natural language processing can be found in [4]. A possible application of this concept to a case of MathML equations can be illustrated by the algorithm described in [9].

Figure 6 shows a set of common subpaths in the sample trees T_0 and T_1. Hence, in this example subpath based tree similarity $S_s(T_0, T_1) = 11$. Figure 3 illustrates the same issue for a case of the sample trees T_0 and T_2. Similarly, subpath based tree similarity can be calculated as $S_s(T_0, T_2) = 9$.

A more practical case is a set TS of trees to be processed in order to find those which are similar to some given tree T_0. Concerning efficiency issues, a significant improvement may be achieved if, instead of computing the similarity for many pairs, an indexing table $I[p]$ for the whole TS corpus is used [4].

Table 2: Indexing table for T_1 and T_2

p	$I[p]$	p	$I[p]$
a	$\{1,2\}$	$e \to g$	$\{1,2\}$
b	$\{1,2\}$	$g \to i$	$\{1,2\}$
c	$\{1\}$	$a \to g$	$\{2\}$
d	$\{1,2\}$	$g \to j$	$\{2\}$
e	$\{1,2\}$	$a \to b \to d$	$\{1,2\}$
g	$\{1,2\}$	$a \to b \to e$	$\{1,2\}$
i	$\{1,2\}$	$b \to e \to g$	$\{1,2\}$
j	$\{2\}$	$e \to g \to i$	$\{1\}$
$a \to b$	$\{1,2\}$	$a \to g \to i$	$\{2\}$
$a \to c$	$\{1\}$	$e \to g \to j$	$\{2\}$
$b \to d$	$\{1,2\}$	$a \to b \to e \to g$	$\{1,2\}$
$b \to e$	$\{1,2\}$	$b \to e \to g \to i$	$\{1\}$
		$b \to e \to g \to j$	$\{2\}$

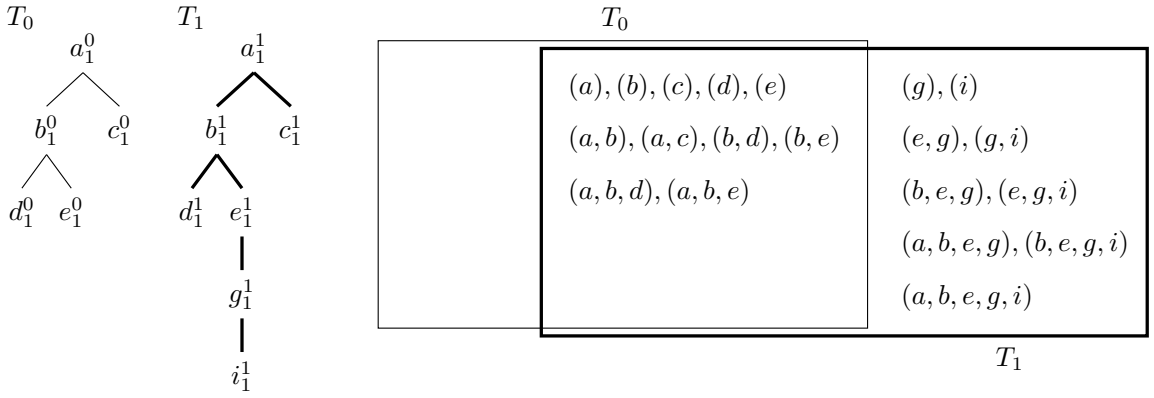


Figure 6: Subpaths in T_0 and T_1

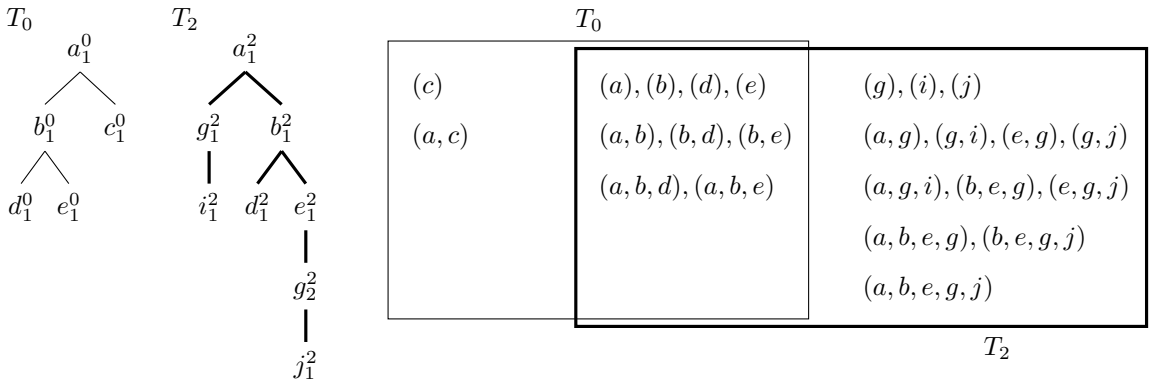


Figure 7: Subpaths in T_0 and T_2

Table 2 provides an example of creating the indexing table for a case of the set of trees consisting of only two trees T_1 and T_2 (shown in Figure 3).

In Table 2 p-columns list all the subpaths from both T_1 and T_2 (i.e. from all the corpus trees); for every subpath the corresponding $I[p]$-cell shows a list of trees where such a subpath exists. Algorithmic complexity of an indexing table based algorithm is $O(L * D^2)$, where L – maximal number of tree leaves, D – maximal tree depth among all the corpus trees [4].

3.3 Tree overlapping algorithm and its modification for math equation structural similarity

A basic tree overlapping algorithm is described in [4] for a case of sentence similarity which is defined as follows. When putting an arbitrary node n_1 of a tree T_1 on a node n_2 of a tree T_2, there might be the same production rule overlapping in T_1 and T_2. Similarity is defined as a number of such overlapping production rules.

In contrast to the base algorithm from [4] where tree terminals are naturally excluded, for a case of mathematical equations we also include terminal nodes as if they had the same production rules (*Relaxation 1*). Also we relax the strictness of the base algorithm and include the pairs of corresponding nodes which are in the same order among their siblings but do not obligatorily have the same production rules for their child nodes (*Relaxation 2*). Below there is a formal definition of our modification.

Assume $L(n_1, n_2)$ represents a set of overlapping node pairs when putting n_1 on n_2. Assume $ch(n, i)$ is i-th child of node n. The set $L(n_1, n_2)$ is being generated by applying the following rules:

1. $(n_1, n_2) \in L(n_1, n_2)$
2. If $(m_1, m_2) \in L(n_1, n_2)$, then $(ch(m_1, i), ch(m_2, i)) \in L(n_1, n_2)$
3. $L(n_1, n_2)$ includes all the pairs generated recursively by the rule No. 2.

A number $N_{TO}(n_1, n_2)$ of production rules (according to the *Relaxation 1*) is defined as follows:

$$N_{TO}(n_1, n_2) = \left\{ (m_1, m_2) \;\middle|\; \begin{array}{l} m_1 \in nodes(T_1) \\ \wedge\, m_2 \in nodes(T_2) \\ \wedge\, (m_1, m_2) \in L(n_1, n_2) \\ \wedge\, PR(m_1) = PR(m_2) \end{array} \right\} \quad (2)$$

In equation 2 $nodes(T)$ is a set of nodes (including terminals) in a tree T, while $PR(n)$ is a production rule rooted at the node n.

Figure 8 shows an example of overlapping tree modification algorithm for $N_{TO}(d_1, d_2) = \{(d^1, d^2), (f^1, f^2), (g^1, g^2)\}$.

Assume $P_{WPR}(n_1, n_2)$ is a set of nodes which is represented as a path from (n_1, n_2) to the top last pair of nodes being in the same order among their siblings. Assume n_i and m_i are nodes of a tree T_i, $ch(n, i)$ is i-th child of node n. According to the *Relaxation 2*, P_{WPR} is defined as follows:

1. $(n_1, n_2) \notin P_{WPR}$
2. If $PR(parent(n_1)) \neq PR(parent(n_2))$
   $\wedge\, ch(parent(n_1), i) = ch(parent(n_2), i)$
   $\wedge\, ch(parent(n_1), i) = n_1$
   $\wedge\, h(parent(n_2), i) = n_2$,
   $(parent(n_1), parent(n_2)) \in P_{WPR}$
3. $P_{WPR}(n_1, n_2)$ includes only pairs generated by applying rule No. 2.

Then the second component for an integral similarity measure can be defined by using the above introduced P_{WPR} as follows:

$$P_{TO}(n_1, n_2) = \left\{ (m_1, m_2) \;\middle|\; \begin{array}{l} (p_1, p_2) \in N_{TO}(n_1, n_2) \\ (m_1, m_2) \in P_{WPR}(p_1, p_2), \\ \qquad \text{if} \quad top(m_1, m_2) = (n_1, n_2) \end{array} \right\} \tag{3}$$

In equation 3 $top(n_1, n_2)$ is the last pair in set $P_{WPR}(n_1, n_2)$: $top(n_1, n_2) = p_{last}(n_1, n_2)$, $p_{last} \in P_{WPR}$.

Thus, for two nodes, the resulting combined similarity measure is defined as follows:

$C_{TO}(n_1, n_2) = |N_{TO}(n_1, n_2)| + |P_{TO}(n_1, n_2)|$

For the whole trees, we get:

$$S_{TO}(T_1, T_2) = \max_{n_1 \in nodes(T_1), n_2 \in nodes(T_2)} C_{TO}(n_1, n_2) \tag{4}$$

3.4 Software implementation

We developed a software prototype in order to arrange a series of experiments for the above described modification of the tree overlapping algorithm for a case of mathematical equations. Figure 9 gives a hint of how the application user interface is organized.

For displaying mathematical equations defined in MathML the library *net.sourceforge.jeuclid* is used.

4 Experiments

One of the problems we faced while attempting to evaluate the algorithm is that, unlike to the NLP domain, there is no substantial corpus of mathematical equation syntactical similarity classes.

4.1 Test Corpora

For our rather preliminary analysis several experts experienced in teaching mathematics in high schools and lyceums were involved. With their help we selected a number of typical trigonometry problems from the set of tasks used in Russian Unified State Examination [3] The selected equations are listed in Table 3.

With the help of our experts, the expressions were classified according their structural similarity. As a result, two types of equation classification were created: a classification based on equation structural similarity (see Table 4) and a classification based on subexpression similarity (see Table 5).

4.2 Tests

Though corpora presented in Tables 4 and 5 aren't representative enough, they make possible to proceed with some preliminary similarity precision estimation. Let us remind that precision is defined as follows:

$$precision = \frac{TP}{TP + FP} \tag{5}$$

In our tests we assume that in equation 5 TP is the number of k first true positive equations belonging to the same class as the query expression, while FP is the number of t first false positive equations: $k + t = n - 1$, where n is the number of equations belonging to the respective class. The preliminary experiments described in this work may be considered a prove-of-concept example for investigating further necessary improvements of the developed algorithm. In the future tests a standard cross-fold validation procedure will be required in order to get trustworthy precision evaluation results.

Figure 11 illustrates the process of structural similarity computation for two expressions from the tiny corpus described earlier. The first expression consists of 34 nodes while the second one has 33 nodes. 20 nodes are equal in both trees. So, $S_{TO} = \frac{20+20}{34+33} = \frac{40}{67} = 0.597$.

4.3 Analysis

Table 6 lists 5 expressions from the base defined in Table 3 which achieve the best scores for the query expression $\sqrt{2}\sin(\frac{3\pi}{2} - x)\sin x = \cos x$ (belonging to the class 1 according to Table 4).

Two best scores are for the equations which also belong to the same class 1, unlike to the equation $-\sqrt{2}\sin(-\frac{5\pi}{2} +$

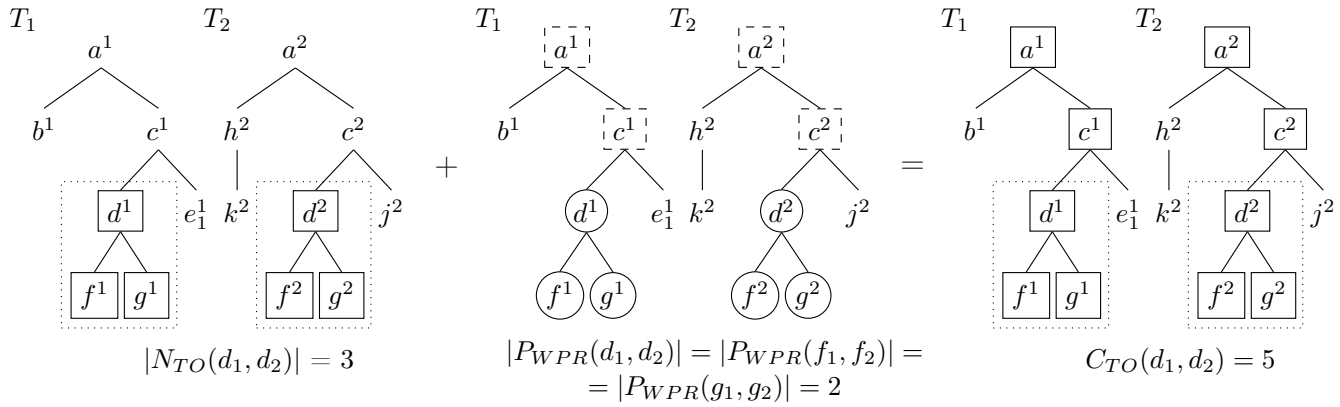


Figure 8: Modified tree-overlapping algorithm: example

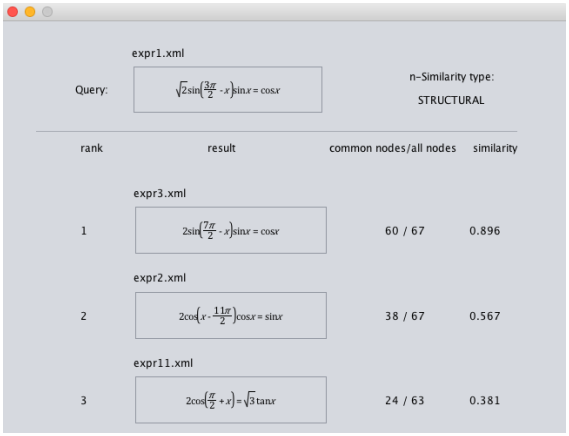


Figure 9: Structural similarity component: GUI

$x) \sin x = \cos x$ (No. 4 in Table 4) which wasn't recognized as a similar expression despite of its obvious similarity. To explain this phenomenon we have to go back to MathML equation structure. As you can see from Figure 10 (left side), two compared equations (both belonging to the class 1 of the corpus) have rather similar structure (at least, from human point of view). However, their tree roots have different number of child nodes, hence their production rules are (formally) different too. It means that we have to enhance equation normalization factor (currently limited by only variable names and numerical values): in the above mentioned case the issue can be resolved by restructuring a tree based equation representation as Figure 10 (right side) shows: both trees in the right side are semantically equivalent to those which are in the left side. After such a normalization, syntactical similarity score increases from 0 (in the "left" case) to 0.44 (in the "right" case).

Similar tests were arranged for expressions from other classes as well as for the case of subexpression similarity. Specifically, for a case of subexperssion similarity, Table 7 lists 5 best results for the query $2\sin^4 x + 3\cos 2x + 1 = 0$ (which belongs to the class 5 according the the test corpus from Table 5) against the equations from the base defined in Table 3. In Table 7, nodes ratio means a ratio of common nodes to all nodes in compared trees.

Among 5 best results listed in Table 7, only the second equation $4\sin^4 2x + 3\cos 4x - 1 = 0$ doesn't belong to the class 5 (according to the experts' classification). Let us analyze a possible reason. The experts didn't include this equation to the class 5 due to the difference between $sin^4 2x$ and $\sin^4 x$ subexpressions. They considered this part of equation as more representative from the viewpoint of structural syntactical similarity. However, the subexpressions $\cos 2x$ and $\cos 4x$ were recognized by the algorithm as subexpression based similar equations to the query since both contains the explicit multiplier before x. Similar to the case of structural similarity this issue could be addresses by the equation representation normalization (i.e. introducing an explicit multiplier equal to 1).

In sum, based on the results presented in Table 5, for the subexpression similarity sample test corpus the average precision $P = \frac{\frac{1}{1}+\frac{1}{1}+\cdots+\frac{3}{4}+\frac{4}{4}+\frac{4}{4}}{13} = \frac{12.25}{13} = 0.94$.

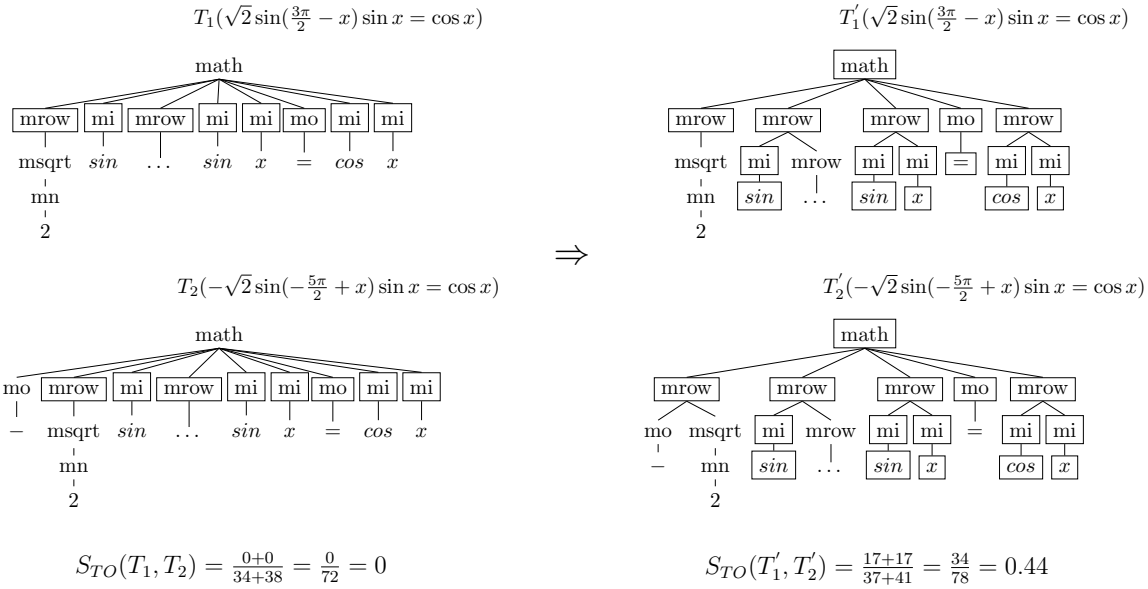
However, such an accuracy achieved for a small test corpus defined in Table 5 may be considered as rather promising but very preliminary evaluation results. Further investigations with using more representative equation corpora are necessary.

5 Conclusion

In this study we adopted a tree overlapping algorithm (used originally in NLP) for mathematical equation syntactical similarity. We implemented the algorithm as a software prototype and arranged a set of experiments with sample test corpora. We discovered that the proposed modification fits well a selection of equations from college-level teaching practice both for the cases of structural and subexpression based syntactical similarity patterns. For the reason that the current implementation has some drawbacks which became evident after the arranged experiments, the further steps towards equation normalization are required in order to achieve better equation classification accuracy.

Table 3: Base of test equations

No.	Equation	No.	Equation
1	$\sqrt{2}\sin(\frac{3\pi}{2}-x)\sin x=\cos x$	16	$2\cos(\frac{\pi}{2}+x)=\sqrt{3}\tan x$
2	$\cos(\frac{\pi}{2}+2x)=\sqrt{2}\sin x$	17	$\sin 2x+2\sin^2 x=0$
3	$2\cos(x-\frac{11\pi}{2})\cos x=\sin x$	18	$2\sin(\frac{7\pi}{2}-x)\sin x=\cos x$
4	$2\sin^4 x+3\cos 2x+1=0$	19	$2\sin^2 x-\sqrt{3}\sin 2x=0$
5	$(2\cos x+1)(\sqrt{-\sin x}-1)=0$	20	$\cos 2x-3\cos x+2=0$
6	$(2\sin x-1)(\sqrt{-\cos x}+1)=0$	21	$2\cos^3 x-\cos^2 x+2\cos x-1=0$
7	$4\sin^4 2x+3\cos 4x-1=0$	22	$\cos 2x+3\sin x-2=0$
8	$\cos 2x=\sin(x+\frac{\pi}{2})$	23	$\sin 2x+\sqrt{2}\sin x=2\cos x+\sqrt{2}$
9	$2\sqrt{3}\cos^2(\frac{3\pi}{2}+x)-\sin 2x=0$	24	$3\cos 2x-5\sin x+1=0$
10	$\cos^2 x-\frac{1}{2}\sin 2x+\cos x=\sin x$	25	$\cos 2x-5\sqrt{2}\cos x-5=0$
11	$\cos 2x=1-\cos(\frac{\pi}{2}-x)$	26	$-\sqrt{2}\sin(-\frac{5\pi}{2}+x)\sin x=\cos x$
12	$\sqrt{\cos^2 x-\sin^2 x}(\tan 2x-1)=0$	27	$\frac{2\sin^2 x-\sin x}{2\cos x-\sqrt{3}}=0$
13	$\tan x+\cos(\frac{3\pi}{2}-2x)=0$	28	$\frac{2\sin^2 x-\sin x}{2\cos x+\sqrt{3}}=0$
14	$\cos x+\cos(\frac{\pi}{2}+2x)=0$	29	$4\cos^4 x-4\cos^2 x+1=0$
15	$\frac{1}{2}\sin 2x+sin^2x-\sin x=\cos x$	30	$4\sin^2 x+8\sin(\frac{3\pi}{2}+x)+1=0$



$S_{TO}(T_1,T_2)=\frac{0+0}{34+38}=\frac{0}{72}=0$ $S_{TO}(T_1',T_2')=\frac{17+17}{37+41}=\frac{34}{78}=0.44$

Figure 10: Tree structure normalization to avoid a false negative case

References

[1] R. Bod. Beyond grammar. *An Experienced-Based Theory of Language. CSLI Lecture Notes*, 88, 1998.

[2] E. D. Demaine, S. Mozes, B. Rossman, and O. Weimann. An optimal decomposition algorithm for tree edit distance. *ACM Transactions on Algorithms (TALG)*, 6(1):2, 2009.

[3] E. Denisova-Schmidt and E. Leontyeva. The unified state exam in russia: problems and perspectives. *International Higher Education*, (76):22–23, 2014.

[4] I. Hiroshi, H. Keita, H. Taiichi, and T. Takenobu. Efficient sentence retrieval based on syntactic structure. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 399–406. Association for Computational Linguistics, 2006.

[5] S. Kamali and F. W. Tompa. Improving mathemat-

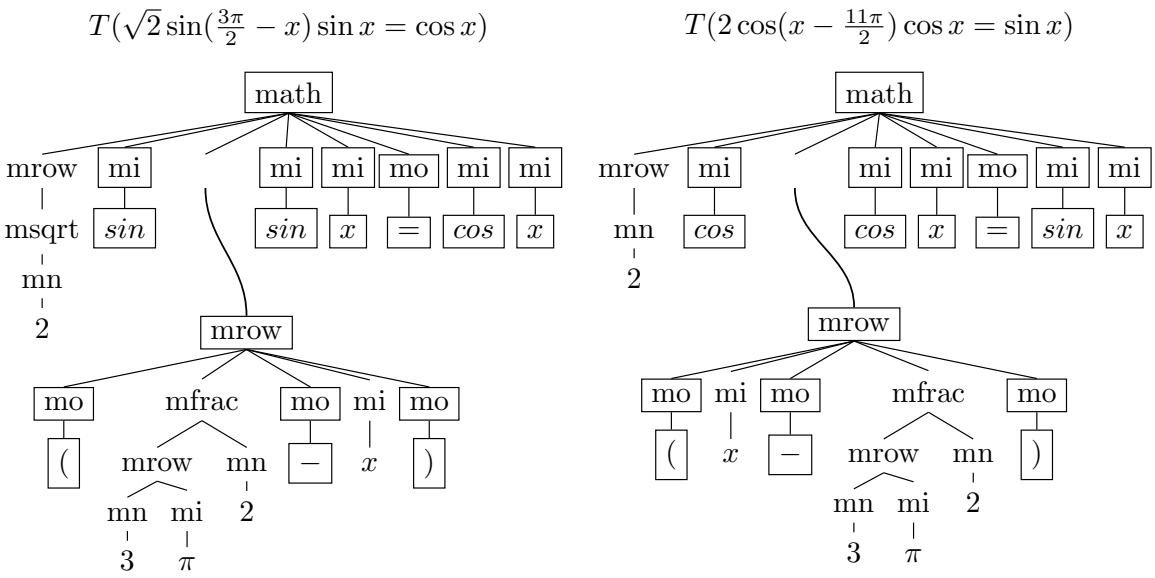


Figure 11: Structural similarity computation: example

Table 4: Structural Similarity Classification

No.	Expression	Class
1	$\sqrt{2}\sin(\frac{3\pi}{2}-x)\sin x=\cos x$	①
2	$2\cos(x-\frac{11\pi}{2})\cos x=\sin x$	
3	$2\sin(\frac{7\pi}{2}-x)\sin x=\cos x$	
4	$-\sqrt{2}\sin(-\frac{5\pi}{2}+x)\sin x=\cos x$	
5	$\cos 2x-3\cos x+2=0$	②
6	$\cos 2x+3\sin x-2=0$	
7	$3\cos 2x-5\sin x+1=0$	
8	$\cos 2x-5\sqrt{2}\cos x-5=0$	
9	$\cos(\frac{\pi}{2}+2x)=\sqrt{2}\sin x$	③
10	$\cos 2x=\sin(x+\frac{\pi}{2})$	
11	$2\cos(\frac{\pi}{2}+x)=\sqrt{3}\tan x$	
12	$2\sin^4 x+3\cos 2x+1=0$	④
13	$4\sin^4 2x+3\cos 4x-1=0$	
14	$4\cos^4 x-4\cos^2 x+1=0$	
15	$(2\cos x+1)(\sqrt{-\sin x}-1)=0$	⑤
16	$(2\sin x-1)(\sqrt{-\cos x}+1)=0$	
17	$\sqrt{\cos^2 x-\sin^2 x}(\tan 2x-1)=0$	
18	$\cos^2 x-\frac{1}{2}\sin 2x+\cos x=\sin x$	⑥
19	$\frac{1}{2}\sin 2x+sin^2 x-\sin x=\cos x$	
20	$\tan x+\cos(\frac{3\pi}{2}-2x)=0$	⑦
21	$\cos x+\cos(\frac{\pi}{2}+2x)=0$	
22	$\frac{2\sin^2 x-\sin x}{2\cos x-\sqrt{3}}=0$	⑧
23	$\frac{2\sin^2 x-\sin x}{2\cos x+\sqrt{3}}=0$	

Table 5: Subexpression Based Similarity

No.	Expression	Class
1	$\cos(\frac{\pi}{2}+2x)=\sqrt{2}\sin x$	①
2	$\cos x+\cos(\frac{\pi}{2}+2x)=0$	
3	$(2\cos x+1)(\sqrt{-\sin x}-1)=0$	②
4	$(2\sin x-1)(\sqrt{-\cos x}+1)=0$	
5	$\sqrt{2}\sin(\frac{3\pi}{2}-x)\sin x=\cos x$	③
6	$2\sin(\frac{7\pi}{2}-x)\sin x=\cos x$	
7	$\frac{2\sin^2 x-\sin x}{2\cos x-\sqrt{3}}=0$	④
8	$\frac{2\sin^2 x-\sin x}{2\cos x+\sqrt{3}}=0$	
9	$2\sin^4 x+3\cos 2x+1=0$	⑤
10	$4\cos^4 x-4\cos^2 x+1=0$	
11	$\cos^2 x-\frac{1}{2}\sin 2x+\cos x=\sin x$	
12	$\frac{2\sin^2 x-\sin x}{2\cos x-\sqrt{3}}=0$	
13	$\frac{2\sin^2 x-\sin x}{2\cos x+\sqrt{3}}=0$	

ics retrieval. *Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009*, pages 37–48, 2009.

[6] M. Pawlik and N. Augsten. Rted: a robust algorithm for the tree edit distance. *Proceedings of the VLDB Endowment*, 5(4):334–345, 2011.

[7] M. Ponomarev and E. Pyshkin. Adopting tree overlapping algorithm for mathml equation structural similarity evaluation. In *Proceedings of the 2nd International Conference on Applications in Information Technology (ICAIT-2016)*, pages 17–20. The University of Aizu, The University of Aizu Press, Oct 2016.

[8] K. Sain, A. Dasgupta, and U. Garain. Emers: a tree matching–based performance evaluation of mathematical expression recognition systems. *International Journal on Document Analysis and Recognition (IJDAR)*, 14(1):75–85, 2011.

[9] K. Yokoi and A. Aizawa. An approach to similarity search for mathematical expressions using mathml.

Table 6: Query: $\sqrt{2}\sin(\frac{3\pi}{2} - x)\sin x = \cos x$

Compared expression	Nodes ratio	Similarity
$2\sin(\frac{7\pi}{2} - x)\sin x = \cos x$	60/67	0.896
$2\cos(x - \frac{11\pi}{2})\cos x = \sin x$	40/67	0.597
$2\cos(\frac{\pi}{2} + x) = \sqrt{3}\tan x$	24/63	0.381
$3\cos 2x - 5\sin x + 1 = 0$	12/60	0.200
$\tan x + \cos(\frac{3\pi}{2} - 2x) = 0$	10/67	0.149

Table 7: Query: $2\sin^4 x + 3\cos 2x + 1 = 0$

Compared expression	Nodes ratio	Similarity
$4\cos^4 x - 4\cos^2 x + 1 = 0$	10/59	0.169
$4\sin^4 2x + 3\cos 4x - 1 = 0$	10/60	0.167
$\cos^2 x - \frac{1}{2}\sin 2x + \cos x = \sin x$	10/63	0.159
$\frac{2\sin^2 x - \sin x}{2\cos x - \sqrt{3}} = 0$	10/64	0.156
$\frac{2\sin^2 x - \sin x}{2\cos x + \sqrt{3}} = 0$	10/64	0.156

Table 8: Evaluating classification precision for a case of subexpression similarity

No.	Expression	$\frac{TP}{TP+FP}$
1	$\cos(\frac{\pi}{2} + 2x) = \sqrt{2}\sin x$	1/1
2	$\cos x + \cos(\frac{\pi}{2} + 2x) = 0$	1/1
3	$(2\cos x + 1)(\sqrt{-\sin x} - 1) = 0$	1/1
4	$(2\sin x - 1)(\sqrt{-\cos x} + 1) = 0$	1/1
5	$\sqrt{2}\sin(\frac{3\pi}{2} - x)\sin x = \cos x$	1/1
6	$\sin(\frac{7\pi}{2} - x)\sin x = \cos x$	1/1
7	$\frac{2\sin^2 x - \sin x}{2\cos x - \sqrt{3}} = 0$	1/1
8	$\frac{2\sin^2 x - \sin x}{2\cos x + \sqrt{3}} = 0$	1/1
9	$2\sin^4 x + 3\cos 2x + 1 = 0$	3/4
10	$4\cos^4 x - 4\cos^2 x + 1 = 0$	3/4
11	$\cos^2 x - \frac{1}{2}\sin 2x + \cos x = \sin x$	3/4
12	$\frac{2\sin^2 x - \sin x}{2\cos x - \sqrt{3}} = 0$	4/4
13	$\frac{2\sin^2 x - \sin x}{2\cos x + \sqrt{3}} = 0$	4/4

Towards a Digital Mathematics Library. Grand Bend, Ontario, Canada, July 8-9th, 2009, pages 27–35, 2009.

[10] K. Zhang and D. Shasha. Simple fast algorithms for the editing distance between trees and related problems. *SIAM journal on computing*, 18(6):1245–1262, 1989.

[11] Q. Zhang and A. Youssef. *An Approach to Math-Similarity Search*, pages 404–418. Springer International Publishing, Cham, 2014.

Analysis of Emotions in Real-time Twitter Streams

Yuki Kobayashi and Maxim Mozgovoy
The University of Aizu
Tsuruga, Ikki-machi, Aizuwakamatsu, Fukushima, Japan
E-mail. {m5201158, mozgovoy}@u-aizu.ac.jp

Myriam Munezero
School of Computing,
University of Eastern Finland, Joensuu, Finland
E-mail: mmunez@uef.fi

Keywords: twitter, social media analysis, streaming

Received: November 4, 2016

The purpose of this paper is to present EmoTwitter 2.0, a system for visualizing discussions and emotions of Twitter users in real time over a specific geographical location. The system, given location information as input, streams users' tweets posted in real time using the Twitter API. In addition, using content analysis, it extracts and visualizes the most frequent words and emotional content of the streamed tweets. Through demonstrations of potential use cases and testing, the system has shown to have practical applicability. It provides an opportunity to easily visualize and compare the discussions of people on Twitter in certain geographical locations, which can be useful, for instance, in targeted messaging.

Povzetek:. Opisan je sistem EmoTwitter 2.0 za vizualizacijo čustev uporabnikov Twiterja.

1 Introduction

Twitter is a social microblog platform that allows people to post their views and emotions on any subject, from current events, political situations, and new products launched, to favorite movies or music [1]. It is one of the most famous microblogs in the world, having more than 300 million active users. The popularity of Twitter and the vast amount of information posted through it, combined with accessibility of the data via public APIs, has made it attractive for purposes such as marketing, and understanding of customers and discussions around the world on varying topics and events. Temporal and spatial analysis are possible thanks to the fact that tweets are time-stamped, and location information is included in tweets and user profiles. To navigate efficiently this large data set, it is important to have visualization capabilities that can provide insights and extract valuable information.

This paper presents EmoTwitter 2.0, an extension of the earlier system EmoTwitter [2], designed as a fine-grained emotion detection and visualization instrument. The system supports several types of text analysis. Its basic functionality is to take user tweets as input and build a word cloud based on the words in the tweets. In addition, using a lexicon-based approach, the system identifies emotions in tweets (classified according to Plutchik's eight emotion categories) and visualizes them. The potential impact and usefulness of the first EmoTwitter version was demonstrated in a user-based evaluation.

Since analyzing tweets in real time with their location information is playing an increasing role in various tasks, especially in monitoring natural disasters [3] and crisis mapping [4], EmoTwitter 2.0 supports filtering tweets by location, and allows to monitor ongoing (real-time) tweet streams.

The purpose of the paper is to present the implementation and some case studies of this functionality. The main contribution of this work is to demonstrate the ability to provide an overview of discussion topics of tweets in a specific geographical area in real time, allowing for easy comparison of what is happening in the Twittersphere in different areas.

2 Related work

Various works exist that have aimed to harvest the Twitter resource in real time for obtaining valuable information, including geographical location of posters.

Closest to the current work is the contribution by Jung et al. [5] that presents GeoViewer, a web-based map application for monitoring real-time social media messages in selected areas. The application helps to visualize and query location-based tweets and provide

The paper is based on: Y. Kobayashi and M. Mozgovoy. Realtime Analysis of Tweet Streams with EmoTwitter, *Proceedings of the 2nd Int'l Conference on Applications in Information Technology (ICAIT-2016)*, 2016, pp. 114-115.

interactive mapping and spatial query functions. Their work though focuses more on disaster events.

Dahyot et al. [6] propose a web-based geolocated visualization tool for summarizing information harvested from the web, including an animated and interactive audio-visual tweet sentiment map. In their work, they focused on the GPS location and time stamp of a tweet. The system processes tweet words to compute a sentiment score using the Stanford Core NLP library.

Graham [7] analyzed and classified a large collection of geotagged Twitter data in order to automatically find popular places from where people are tweeting. This approach makes use of spatial analysis of the GPS latitude and longitude values of a tweet, and combines it with content analysis of the text and hashtags of the respective tweet. In this work however, only the top five hashtags are used to identify the theme of tweets in a region.

Heatmaps are used in "Global Twitter Heartbeat" project [13] to map world emotions expressed on Twitter in real time. The project analyzes every tweet to assign its location (in addition to GPS data it processes the text of the tweet itself), and tone values and then visualizes the conversation in a heat map infographic that combines and displays tweet location, intensity and tone.

Other researchers analyzed both the location information of a twitter user and the location of physical events referred to in twitter feeds [8]. They designed an algorithm that identifies distinct event signatures, clusters them based on events they describe, and analyzes the resulting clusters for location information. This information is subsequently translated using Google Maps API for geolocation, thus offering a real-time view of ongoing events on a map. Our work though focuses on a real-time view of the users.

The Google API was also used to display and study the spatial temporal activity of the West Nile Virus using Twitter data, thus allowing to observe changes in tweet messages regarding the virus [9].

Working with geolocation information has limitations as identifying geolocation information of tweets is not always possible. The study of a month of the Twitter Decahose (10% of the global Twitter stream, consisting of over 1,5 billion tweets from more than 70 million users) identified that just over 3% of all tweets include native geolocation information, with 2% offering street address-level resolution in real time [10].

In all the analyzed related works, we have not found a similar type of visualization as presented in our paper — the display of changes in the tweets topics in real time for a particular location.

3 System description

EmoTwitter 2.0 consists of backend and frontend layers, described below. For the backend, we focus on the new implemented functionality: tweet streaming and tweet location analysis.

3.1 Backend: analyzing tweet streams

Twitter provides two kinds of APIs: REST API and Streaming API. The REST API allows the retrieval of tweets, while the Streaming API gives access to global streams of tweet data in real time. Figures 1 and 2 show the comparison between REST API and Streaming API processes. Both APIs rely on HTTP connections, but Streaming API requires keeping a persistent connection open [3].

The Streaming API is asynchronous: the tweets are returned to the caller as they arrive, and the download process continues to work. The Streaming API further lets the user to filter the stream by geographical location of tweets and tweet language. This location information is available when a Twitter user enables the geographical location on their devices. Moreover, while the REST API has a Twitter-imposed limit of downloading only the most recent 3200 tweets of each user, and the total hourly limit of queries, the Streaming API does not impose such a limit. However, arriving tweets represent only a small segment of the raw global Twitter stream, which is another limitation of the Twitter service.

3.2 Backend: location analysis

To determine and retrieve tweets based on the geographic location, we use tweet geotags. This information is available when a user enables location information sharing on their device. Thus, we are limited to the users who enable this option.

In order to analyze and visualize the tweets based on the location information, EmoTwitter uses the Google Maps API — an interface for interacting with the backend web service of Google Maps [11]. Taking a location string as an input, EmoTwitter 2.0 queries Google Maps to retrieve the geographical coordinates of the string (presuming that Google Maps is able to interpret the string as a location). Next, EmoTwitter receives the values of latitude and longitude as a response. Using these values and the streaming API, the system retrieve tweets posted near the specified geographical location. Currently the system returns all the tweets inside a square having the specified point in the center and sides of 1.0-mile length.

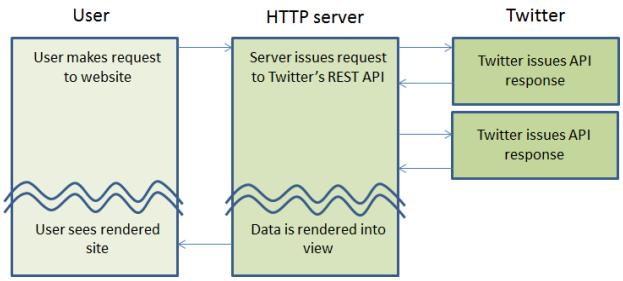


Figure 1: REST API process (adapted from [12]).

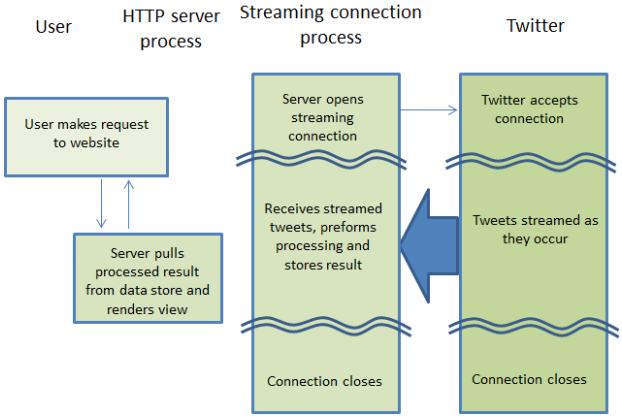


Figure 2: Streaming API process (adapted from [12]).

3.3 Frontend

EmoTwitter 2.0 provides a graphical user interface for interacting with the system. First, a user logs into the system using their Twitter username and password. This is required in order to create an access token that Twitter application uses to acquire data. Then the system displays a window where the user can enter a location (see Figure 3). Any free-form address that can be processed with Google Maps can be entered. Next, the user has to click the "start stream" button, and the system starts streaming tweets within the given location range. The results are displayed in a textbox as they arrive.

The system also updates in real time a word cloud containing the most frequent words in the tweets, a functionality implemented in the first version of the system as described in [2]. The user can stop the analysis of incoming tweets at any moment.

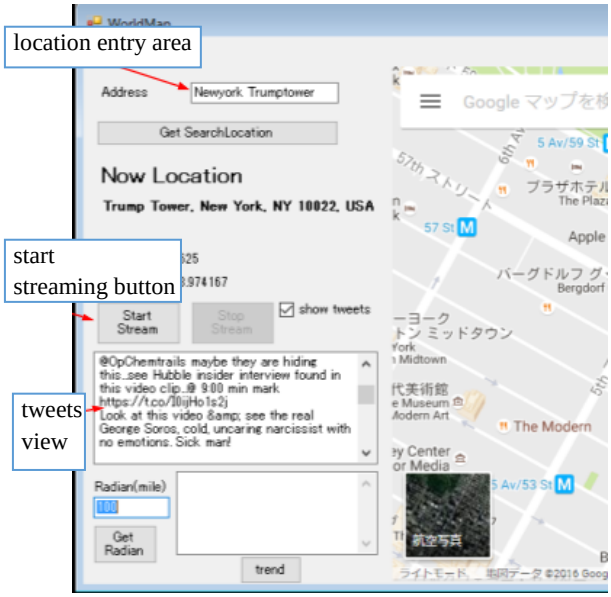


Figure 3: Location-based tweet retrieval.

4 Demonstration

To demonstrate the practical viability of EmoTwitter, we shall discuss two interesting cases that occurred while we were working on the paper.

Case 1. The former Cuban president, Fidel Castro passed away at the age of 90 on Friday evening, 26 November 2016. As expected, a lot of Twitter chatter on the Saturday morning for instance in Cuba's capital Havana and Birán (Castro's birth town) were related to Castro and Cuba (See Figures 4a and 4b).

At the same time, we examined the chatter around other countries that due to political history situations, might have been expected to have reactions to the death of the former president. For instance, the USA was one of the countries where diplomatic relations were severed. Thus looking at the Twitter chatter in Washington DC, the capital city and government center, on the same Saturday morning for an equal period of time, there was some chatter on Fidel Castro, but not so much in comparison to

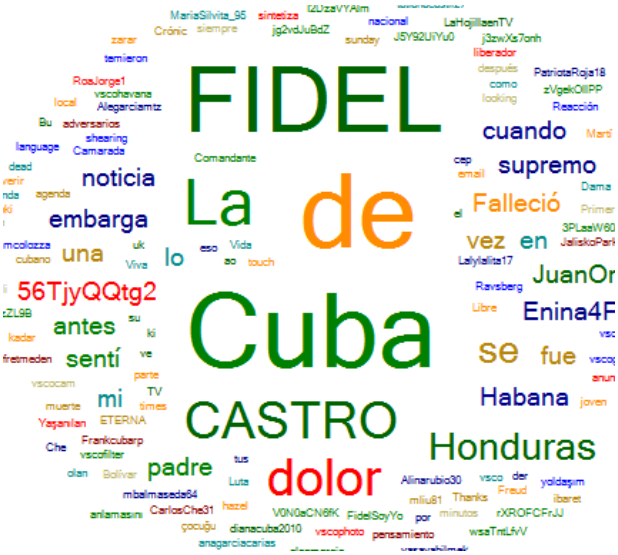


Figure 4a: Twitter word cloud in Havana, Cuba.

Figure 4b: Twitter word cloud in Birán, Cuba.

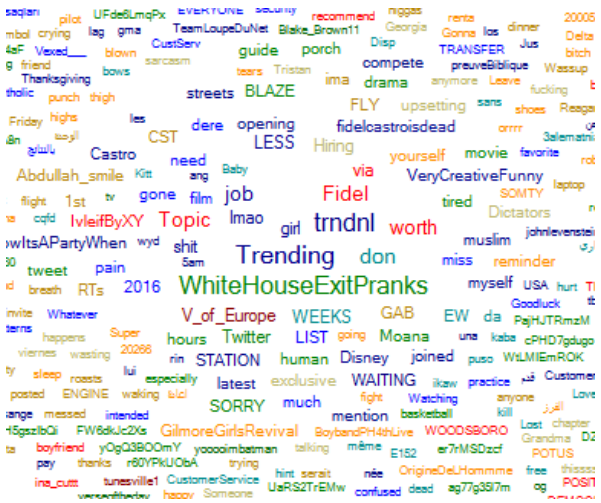


Figure 4c**:** Twitter word cloud in Washington DC, USA.

Havana and Birán (see Figure 4c for snapshot of the chatter). However, continuing to stream for a longer period, we might observe increased chatter as coverage of the death spreads.

Case 2. Another interesting case for EmoTwitter was the Scottish League Cup final match that took place on 27 November 2016. The final teams were Celtic and Aberdeen. The match promised to be interesting since if Celtic won, it would be their 100th trophy, and it would be the first piece of silverware for the former Liverpool manager, Brendan Rodgers, with the Celtic team.

On Sunday, 27 November 2016, at 17:00 EET, we entered the location of the stadium where the final match was taking place, which was at Hampden Park, Glasgow, We used the exact address, 'Glasgow G42 9BA, UK' to give us the exact location (see Figure 5).

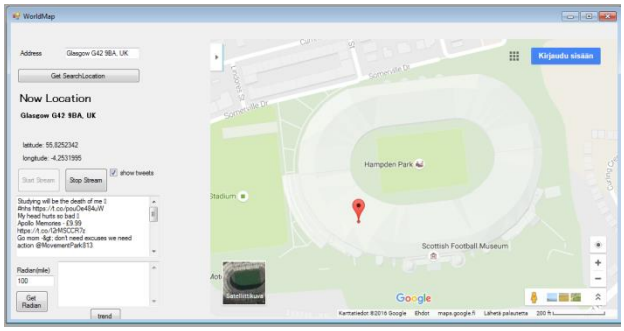


Figure 5: Location of streaming at Hampden Park.

When the match started at 17:00, we started collecting the Twitter streaming data, and stopped this process in the middle of the game (during half-time). By the end of the first half, Celtic was leading by two goals, while Aberdeen scored zero. As can be seen from the snapshots in Figures 6a and 6b, there was some Twitter chatter about the game, as expected. The word 'goal' (see Figure 6a) was in the spotlight by the end of the first half as a player Forrest scored a goal 8 min before the half time.

Figure 6a**:** Word cloud after the goal by Forrest.

Figure 6b**:** Word cloud of the chatter at half time.

After the mid-game break, we resumed data collecting, and stopped the process after the game ended. Shortly after the break, Celtic player Dembele scored

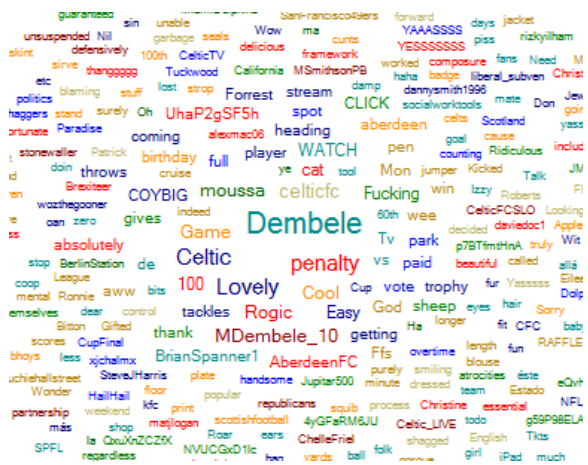


Figure 7: Word cloud after the Dembele goal.

another goal through a penalty. This information appeared in the spotlight in the Twitter chatter (see Figure 7a). The Figure 8b shows the chatter at the end of the game. There are some happy expressions, as well as the number 100, indicating that Celtic has won their 100th game.

5 Conclusion

We have presented EmoTwitter 2.0 with its extended functionality of analyzing tweets streams in real time on the basis of location information. We illustrated with two example cases how the visualization of frequent words in real time can be used to explore and compare ongoing discussions in the Twittersphere. Our use cases revealed that in most cases words in a word cloud are not stable, and quickly change over time. However, sometimes it is possible to catch a somewhat longer trend, when a number of keywords stay in active use for minutes and even hours. This was the case, for example, with the death of the former president of Cuba (Case 1). In Case 2, we showed how EmoTwitter can be used to examine Twitter chatter during a football match.

Figure 8**:** Word cloud after the game.

The analysis and visualization is however limited, since we do not have access to the complete real-time stream of tweets, and not all Twitter users have their location sharing enabled on their devices. Still, we are able to gain insights from visualizing real-time Twitter chatter in a particular location. Such information opens up opportunities, for instance in targeted messaging and advertising.

Our future work will be related to further extensions of EmoTwitter functionality. We are also going to release the system as open source software.

6 References

[1] E. Martínez-Cámara, M. Martín-Valdivia, L. Ureña-López, and A. Montejo-Raéz, "Sentiment Analysis in Twitter," *Natural Language Engineering*, vol. 20, no. 01, pp. 1–28, 2014.

[2] M. Munezero, C. S. Montero, M. Mozgovoy, and E. Sutinen, "EmoTwitter — A Fine-Grained Visualization System for Identifying Enduring Sentiments in Tweets," in *International Conference on Intelligent Text Processing and Computational Linguistics*, 2015, pp. 78–91.

[3] P. S. Earle, D. C. Bowden, and M. Guy, "Twitter earthquake detection: earthquake monitoring in a social world," *Annals of Geophysics*, vol. 54, no. 6, 2012.

[4] P. Meier, "How the UN used social media in response to Typhoon Pablo," *Standby Task Force*, 2012.

[5] C.-T. Jung, M.-H. Tsou, and E. Issa, "Developing a real-time situation awareness viewer for monitoring disaster impacts using location-based social media messages in Twitter," in *International Conference on Location-Based Social Media Data, Athens, GA, USA March*, 2015, pp. 12–14.

[6] R. Dahyot, C. Brady, C. Bourges, and A. Bulbul, "Information visualisation for social media analytics," in *Computational Intelligence for Multimedia Understanding (IWCIM), 2015 International Workshop on*, 2015, pp. 1–5.

[7] T. Graham, "Geospatial Clustering and Classifying of Twitter Data," *Projects in Geospatial Data Analysis: Spring 2015*, 2015.

[8] P. Giridhar, T. Abdelzaher, J. George, and L. Kaplan, "Event localization and visualization in social networks," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2015, pp. 35–36.

[9] R. Sugumaran and J. Voss, "Real-time spatio-temporal analysis of west nile virus using twitter data," in *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*, 2012, p. 39.

[10] K. Leetaru, S. Wang, G. Cao, A. Padmanabhan, and E. Shook, "Mapping the global Twitter heartbeat: The geography of Twitter," *First Monday*, vol. 18, no. 5, 2013.

[11] E. Newton, *Google Maps API for .NET*. Available: https://github.com/ericnewton76/gmaps-api-net (2016, Dec. 08).

[12] Twitter Inc, *Twitter Developer Documentation: Streaming APIs*. Available: https://dev.twitter.com/streaming/overview (2016, Dec. 08).

[13] Global Twitter Heartbeat. Project homepage: http://www.sgi.com/go/twitter

OD-matrix Estimation Based on a Dual Formulation of Traffic Assignment Problem*

Alexander Yu. Krylatov, Anastasiia P. Shirokolobova and Victor V. Zakharov
Saint Petersburg State University
7/9 Universitetskaya nab., St. Petersburg, 199034 Russia
E-mail: a.krylatov@spbu.ru, a.shirokolobova@spbu.ru, v.zaharov@spbu.ru

Keywords: OD-matrix, Wardrop's user equilibrium, duality

Received: December 1, 2016

Congestion, accidents, greenhouse gas emission and others seem to become unsolvable problems for all levels of management in modern large cities worldwide. The increasing motorization dynamics requires development of innovative methodological tools and technical devices to cope with problems emerging on the road networks. First of all, control system for urban traffic area has to be created to support decision makers by processing a big volume of transportation data. The input for such a system is a volume of travel demand between origins and destinations — OD-matrix. The present work is devoted to the problem of OD-matrix estimation. Original OD-matrix estimation technique is offered by virtue of plate scanning sensors location. Mathematically developed technique is based on a dual formulation of the traffic assignment problem (equal journey time by alternative routes between any OD-pair). Traffic demand between certain OD-pair is estimated due to journey time obtained from plate scanning sensors. Moreover, the explicit relationship between traffic demand and journey time is obtained for network of parallel routes with one OD-pair. Eventually, the developed method was experimentally implemented to the Saint-Petersburg road network.

Povzetek: OD-matrika povezuje izvore in cilje prometnih povezav. Prispevek se ukvarja z iskanjem približkov OD-matrike z dvojno formulacijo.

1 Introduction

OD-matrix estimation and reconstruction are urgent and complicated challenges, since road networks of modern cities are extremely large and intricate. In general, OD-matrix estimation and reconstruction are different problems: the first means to obtain approximate values, while the second has a goal to obtain precise values of actual traffic demands [1]. One of the first mathematical models for OD-matrix estimation was formulated in a form of bi-level program [2]. Despite numerous publications, this problem still attracts researchers from all over the world [3–8]. A detailed comparative analysis of methods for trip matrix estimation was made in [4]. From a practical perspective, the most promising technique is combination of data obtained both from plate scanning sensors and link-flow counts [5].

This paper is also devoted to OD-matrix estimation problem. We believe that a plate scanning sensor is highly efficient engineering equipment. Indeed, due to link-flow counts one could obtain solely amount of vehicles on the link, while plate scanning allows to estimate the average travel time between origin and destination by identification the vehicle in origin and destination points. Since travel time between an origin-destination pair is a Lagrange multiplier for a primal traffic assignment problem (TAP), it is the variable in a dual formulation of TAP. Therefore, we are able to formulate a new bi-level optimization program for OD-matrix estimation based on data obtained from link-flow plate scanning sensors on congested networks.

The present article is organized as follows. In Section 2 the network of parallel routes with one OD-pair is investigated. The idea of OD-matrix estimation based on information about travel times between OD-pairs is clarified. Section 3 provides a dual formulation of traffic assignment problem in two subsections: for a simple network with one OD-pair and parallel routes, and for the general topology network. Section 4 describes bi-level optimization program for OD-matrix estimation on a congested network by virtue of plate scanning sensors. Section 5 is devoted to the experimental implementation of the developed approach to the Saint-Petersburg road network. Conclusions are given in Section 6.

2 The network of parallel routes

Let us ntroduce the following notation: F is traffic demand between OD-pair; f_i is traffic flow on the route i, $i = \overline{1,n}$, $f = (f_1, \dots, f_n)$, $\sum_{i=1}^{n} f_i = F$; $t_i(f_i) = a_i + b_i f_i$ is

* This paper is based on Alexander Krylatov, Anastasiia Shirokolobova and Victor Zakharov, A dual formulation of the traffic assignment problem for OD-matrix estimation, published in the Proceedings of the 2nd International Conference on Applications in Information Technology (ICAIT-2016).

travel time on congested arc i, $i = \overline{1,n}$. In the present work we model travel time on a congested arc as the linear function.

Let us formulate traffic assignment problem on network of parallel routes as an optimization program [9, 10]:

$$z(f^*) = \min_f z(f) = \min_f \sum_{i=1}^{n} \int_0^{f_i} t_i(u)du, \qquad (1)$$

subject to

$$\sum_{i=1}^{n} f_i = F, \qquad (2)$$

$$f_i \geq 0 \quad \forall i = \overline{1,n}. \qquad (3)$$

Wardrop's first principle states that the journey times in all routes actually used are equal and less than those that would be experienced by a single vehicle on any unused route [10, 11]. Traffic flows that satisfy this principle are usually referred to as "user equilibrium" (UE) flows, since each user chooses the route that is the best. On the network of parallel routes UE is reached by such assignment $f^* = (f_1^*, \ldots, f_n^*)$ as:

$$\begin{cases} t_i(f_i^*) = t^* > 0 & \text{when } f_i^* > 0, \\ t_i(f_i^*) > t^* & \text{when } f_i^* = 0, \end{cases} \quad i = \overline{1,n}.$$

Thus, the mathematically formalized idea of UE (1)–(3) can be used in reconstruction of traffic assignment on the network between origin-destination pair. On the other hand, if we know travel time t^* between OD-pair, we are able to reconstruct traffic demand F on the linear network of parallel routes.

Without loss of generality we assume that routes are numbered as follows:

$$a_1 \leq \ldots \leq a_n.$$

Theorem 1. *Traffic demand F for a linear network of parallel routes can be obtained explicitly:*

$$F = t^* \sum_{s=1}^{k} \frac{1}{b_s} - \sum_{s=1}^{k} \frac{a_s}{b_s}, \qquad (4)$$

where k satisfies

$$a_1 \leq \ldots a_k < t^* \leq a_{k+1} \ldots \leq a_n. \qquad (5)$$

Proof. Travel time t^* on used routes is the Lagrangian multiplier that corresponds to the restriction (2) of optimization program (1)–(3) [9, 12, 13].

Since goal function (1) is convex then the Kuhn-Tucker conditions are both necessary and sufficient. Let us introduce Lagrange multiplier μ for the flow conservation constraint (2) and multipliers $\eta_i \geq 0$, $i = \overline{1,n}$ for (3). The Lagrangian for optimization problem (1)–(3) is

$$L = \sum_{i=1}^{n} \int_0^{f_i} t_i(u)du + \mu \left(F - \sum_{i=1}^{n} f_i \right) + \\ + \sum_i \eta_i \left(-f_i \right). \qquad (6)$$

The derivative of Lagrangian (6) at variable f_i has to be equal to zero:

$$\mu = t_i(f_i) - \eta_i.$$

Complementary slackness condition states that $\eta_i \cdot f_i = 0$. This equation holds when at least one of the variables is zero. Thus, if $f_i > 0$, then $\eta_i = 0$ and

$$\mu = t_i(f_i) = a_i + b_i f_i. \qquad (7)$$

However, if $f_i = 0$, then $\eta_i \geq 0$ and

$$\mu = t_i(f_i) - \eta_i = a_i - \eta_i.$$

Hence, if $a_i \geq \mu$ then $f_i = 0$. On the contrary, if we express f_i in terms of μ from (7) we get

$$f_i = \frac{\mu - a_i}{b_i}.$$

Therefore, if $f_i > 0$ then

$$\mu > a_i.$$

Thus we are able to define the set of actually used routes (routes with nonzero flows):

$$f_i = \begin{cases} \frac{\mu - a_i}{b_i} & \text{when } a_i < \mu, \\ 0, & \text{when when } a_i \geq \mu, \end{cases} \quad i = \overline{1,n}. \qquad (8)$$

Without loss of generality, we could renumber routes in such a way that $a_1 \leq \ldots \leq a_k < \mu \leq a_{k+1} \leq \ldots \leq a_n$. Then, due to (2) and (8) we obtain

$$\sum_{i=1}^{n} f_i = \sum_{s=1}^{k} \frac{\mu - a_s}{b_s} = F$$

and, consequently,

$$F = \sum_{s=1}^{k} \frac{\mu - a_s}{b_s}.$$

Eventually, since μ is t^* by definition, the theorem is proved. □

Therefore, if we know travel time of a vehicle on any alternative route between OD-pair, appropriate traffic demand can be uniquely reconstructed. Due to such results the developed approach seems to be promising. The main idea of this method is based on the first principle of Wardrop is as follows: if we define journey time of a vehicle on any actually used route between certain OD-pair, then we believe that journey time on all other used routes is the same.

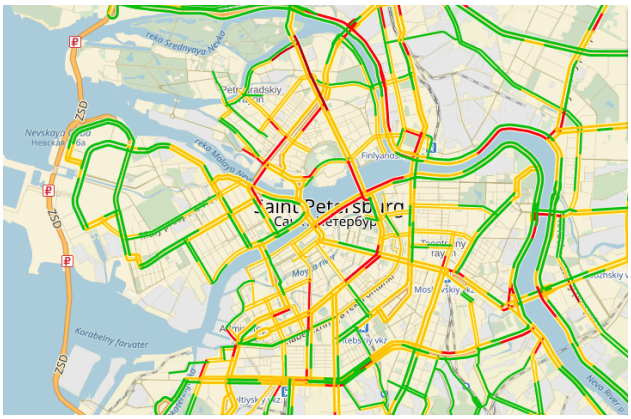


Figure 1: The traffic in Saint-Petersburg city

3 Dual formulation of TAP

Generally, t^* could be easily determined between any pair of origin and destination on a real city road network. Indeed, there are online services collecting information about average travel speeds on all arcs of a city road network.

For instance, "Yandex.Traffic" based on "Yandex.Maps" reflects the current road situation online (fig.1). Due to the large scale databases this service is able to make statistical predictions for different time periods and different days of week. Average travel speed through any arc of road network is in the public domain (fig.2).

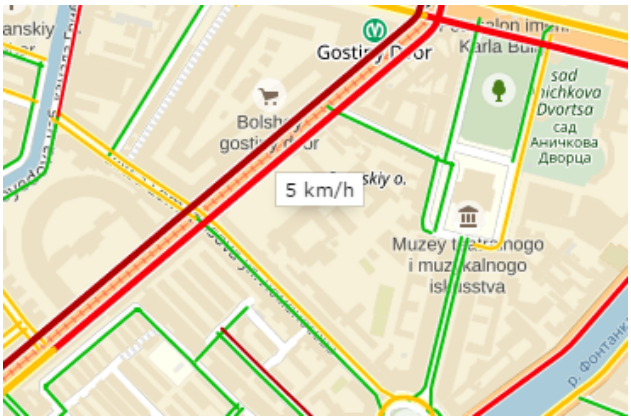


Figure 2: Data from Yandex.Maps

Therefore, we can easily determine travel time through the whole route between any pair of origin and destination, using the information about average speed on the arcs. We believe that road traffic is assigned according to the first Wardrop principle. Thereby, if we estimate travel time through the shortest route between OD-pair then we obtain t^* for this OD-pair. Note, that such information about road network could also be useful for more efficient allocation of resources by different companies [14].

Actually, due to information about equilibrium travel time between OD-pairs we can estimate OD-matrix. Indeed, let us refer to the duality theory of mathematical programming.

3.1 Dual formulation of TAP for the network of parallel routes

We introduce dual variables $\eta = (\eta_1, \dots, \eta_n)$ and define dual traffic equilibrium problem for the network of parallel routes:

$$\max_{\eta} \sum_{i=1}^{n} \theta_i(\eta),$$

with constraints

$$\eta_i \geq 0, \quad \forall i = \overline{1,n},$$

where $\theta_i(\eta)$ for all $i = \overline{1,n}$ satisfies the following optimization program

$$\theta_i(\eta) = \sum_{i=1}^{n} \min_{f_i} \left(\int_0^{f_i} t_i(u)du - \eta_i f_i \right),$$

subject to

$$\sum f_i = F.$$

It is proved that equilibrium assignment problem for a network of parallel routes can be reduced to the fixed point problem and is expressed explicitly [15]. Now let us prove that the above mentioned bi-level program is really dual TAP.

3.2 Dual formulation of TAP for a network of general topology

Let us consider a network of general topology presented by oriented graph $G = (N, A)$. We introduce the following notation: W is set of OD-pairs, $w \in W$, $W \in N$; K^w is the set of routes connecting OD-pair w; F^w is traffic demand for OD-pair w, $F = \left(F^1, \dots, F^{|W|}\right)^{\mathrm{T}}$; f_k^w is traffic flow on the route $k \in K^w$, $\sum_{k \in K^w} f_k^w = F^w$; $f^w = \{f_k^w\}_{k \in K^w}$ and $f = \{f^w\}_{w \in W}$; x_a is traffic flow on the arc $a \in A$, $x = (\dots, x_a, \dots)$; $t_a(x_a)$ is the link travel cost on the arc $a \in A$; $\delta_{a,k}^w$ is the indicator: 1 if acr a is included in route k, 0 if otherwise.

User equilibrium on transportation network G is reached by such x^* that

$$Z(x^*) = \min_{x} \sum_{a \in A} \int_0^{x_a} t_a(u)du, \tag{9}$$

subject to

$$\sum_{k \in K^w} f_k^w = F^w, \quad \forall w \in W, \tag{10}$$

$$f_k^w \geq 0, \quad \forall w \in W, \tag{11}$$

with definitional constraints

$$x_a = \sum_{w \in W} \sum_{k \in K^w} f_k^w \delta_{a,k}^w, \quad \forall a \in A. \tag{12}$$

User equilibrium principle allows us to introduce t_w^*, that is equilibrium journey time for any OD-pair w.

Lemma. *t_w^* is the Lagrange multiplier in the optimization program (9)–(12) corresponding to the constraint (11).*

Proof. The Lagrangian of the problem (9)–(12) is

$$L = \sum_{a \in A} \int_0^{x_a} t_a(u)du + \sum_w \mu_w \left(F^w - \sum_{k \in K^w} f_k^w \right) + \\ + \sum_w \sum_{k \in K^w} \eta_k^w \left(-f_k^w \right),$$

where μ_w and $\eta_k^w \geq 0$ are Lagrangian multipliers, and differentiation of the Lagrangian yields:

$$\frac{\partial L}{\partial f_k^w} = \sum_{a \in k} t_a(x_a) - \mu_w - \eta_k^w = 0.$$

Note, that according to complementary slackness $\eta_k^w f_k^w = 0$. Thus, for $f_k^w > 0$ the following expression holds

$$\sum_{a \in k} t_a(x_a) = \mu_w, \quad \forall k \in K^w, w \in W. \tag{13}$$

Actually, left part of (13) is journey time on any used route ($f_k^w > 0$) between OD-pair r. Therefore, proposition is proved. □

Eventually, according to the Lemma the following equality is true:

$$t_w^* = \sum_{a \in k} t_a(x_a) \quad \forall k \in K^w, w \in W.$$

Theorem 2. *Dual mathemetical problem for a TAP, expressed by (9)–(12), is the following bi-level program:*

$$\max \theta(T) \tag{14}$$

where $\theta(T)$ is defined by

$$\theta(T) = \min_{f \geq 0} \left\{ \sum_{a \in A} \int_0^{x_a} t_a(s)ds + \right. \\ \left. + \sum_w t_w \left(F^w - \sum_{k \in K^w} f_k^w \right) \right\}, \tag{15}$$

subject to definitional constraints

$$x_a = \sum_{w \in W} \sum_{k \in K^w} f_k^w \delta_{a,k}^w, \quad \forall a \in A. \tag{16}$$

Proof. We assume that x^* is a solution of (9)–(12). We introduce multipliers for the flow conservation constraints (10) and (11). Indeed, according to Lemma, we can use t_w as Lagrangian multipliers for the constraints (10). The Lagrangian for the problem (9)–(12) is

$$L(x_a, t_w) = \sum_{a \in A} \int_0^{x_a} t_a(s)ds + \\ + \sum_w t_w \left(F^w - \sum_{k \in K^w} f_k^w \right). \tag{17}$$

If $L(x_a, t_w)$ has a saddle point (x_a^*, t_w^*) in admissible set, then x_a^* is the solution of the problem (9)–(11), and t_w^* is the solution of the following optimization problem [16]:

$$\max_T L(x_a^*, t_w)$$

in case of

$$x_a = \sum_{w \in W} \sum_{k \in K^w} f_k^w \delta_{a,k}^w, \quad \forall a \in A,$$

where $T = (t_1, \ldots, t_{|W|})^{\mathrm{T}}$.

This duality relation holds when the theorem of equivalence is satisfied [16]. □

4 OD-matrix estimation from plate scanning sensors

Link-flow counts provide the amount of vehicles on the links. Plate scanning sensors associated with the certain links identify plates of vehicles from link-flow. Thus, when any vehicle crosses a link with some sensor then sensor records its plate number and fixation time. Eventually, database consisting of {plate number, fixation time, number of sensor} is accumulated [3]. With the help of such database the travel time between any origin-destination pair can be evaluated directly. Indeed, one just has to know fixation time of the vehicle in origin and fixation time in destination to define t_w^* (as the difference between fixation time in the destination and fixation time in the origin) for any $w \in W$.

Therefore, the following bi-level optimization program can be formulated:

$$\min_F \left(\overline{F} - F \right)^{\mathrm{T}} U^{-1} \left(\overline{F} - F \right) + \\ + (T^* - T)^{\mathrm{T}} (T^* - T), \tag{18}$$

subject to

$$F \geq 0, \tag{19}$$

where T solves

$$\max \theta(T), \tag{20}$$

when $\theta(T)$ is defined by

$$\theta(T) = \min_{f \geq 0} \left\{ \sum_{a \in A} \int_0^{x_a} t_a(s)ds + \right.$$
$$\left. + \sum_w t_w \left(F^w - \sum_{k \in K^w} f_k^w \right) \right\}, \qquad (21)$$

subject to definitional constraints

$$x_a = \sum_{w \in W} \sum_{k \in K^w} f_k^w \delta_{a,k}^w, \quad \forall a \in A. \qquad (22)$$

Here, (18) is the generalized least squares estimation and $\overline{F}$ is the aprior volume of travel demand between all OD-pairs, and U is the weighting matrix.

5 Computational experiment

Let us consider Saint-Petersburg road network (fig. 3). We define seven origin-destination pairs with seven shortest routes from seven periphery origins {1,2,3,4,5,6,7} to the center destination {8}. According to STSI (State Traffic

Figure 3: Selected OD-pairs on the Saint-Petersburg road network with the shortest routes

Safety Inspectorate), nowadays there are 253 plate scanning sensors observing the Saint-Petersburg road network (fig. 4). Due to these sensors, we are able to identify travel time between chosen OD-pairs (table 1). The developed approach is based on user equilibrium principle, which suggests that value of travel time on the shortest route is travel time on any actually used route. Moreover, we are able to calculate aprior flows $\overline{F}$ using the gravity model [4].

Let us use these data as inputs for bi-level optimization program (18)–(22). MATLAB was employed to carry out the simulation. Results of simulation are presented in the table 2. Moreover, these results are available in comparison with aprior flows. Figure 5 gives a visualization of such a comparison. One can see that rough aprior estimation

Figure 4: Sensors location on the Saint-Petersburg road network

Table 1: Journey time obtained from plate scanning sensors

Route in OD-pair	**Travel time t^*/min**
1–8	89
2–8	80
3–8	83
4–8	78
5–8	45
6–8	57
7–8	36

of trip flows, obtained by gravity model, was adjusted by virtue of information about actual travel time between OD-pairs. Therefore, approach introduced in this paper seems to be quite useful.

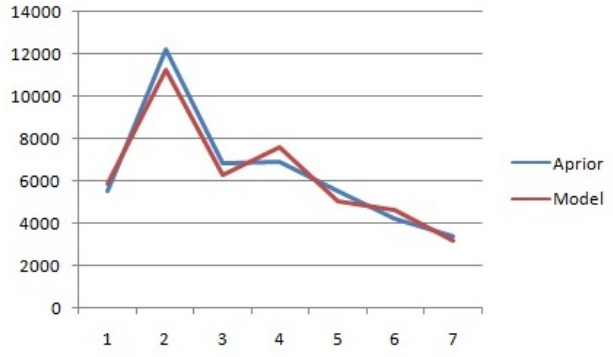


Figure 5: Comparison of model and aprior flows

6 Conclusion

The present paper was devoted to the problem of OD-matrix estimation. Original OD-matrix estimation technique based on a dual formulation of the traffic assign-

Table 2: Comparison of model and aprior flows

OD-pair	Aprior flow	Model flow
1–8	5523	5910
2–8	12232	11253
3–8	6827	6295
4–8	6938	7631
5–8	5534	5080
6–8	4254	4650
7–8	3395	3202

ment problem was offered. Due to this technique traffic demand estimation between certain OD-pair could be solely based on the information about journey time. Since journey time is easily obtained from plate scanning sensors, the developed technique has an obvious practical significance. Moreover, explicit relationship between traffic demand and journey time was obtained for the network of parallel routes with one OD-pair. Such a result gives a clear understanding of basis relationship between demand and journey time. Eventually, the developed approach was experimentally implemented to the Saint-Petersburg road network, that demonstrates its effectiveness.

References

[1] Hazelton M. Inference for origin-destination matrices: estimation, prediction and reconstruction // Transportation Research Part B. 2001. No 35. P. 667–676.

[2] Yang H., Sasaki T., Iida Y., Asakura Y. Estimation of origin-destination matrices from link traffic counts on congested networks // Transportation Research Part B. 1992. No 26 (6). P. 417–434.

[3] Zakharov V., Krylatov A. OD-matrix estimation based on plate scanning // 2014 International Conference on Computer Technologies in Physical and Engineering Applications (ICCTPEA). 2014. P. 209-210.

[4] Medina A., Taft N., Salamatian K., Bhattacharyya S., Diot C. Traffic matrix estimation: existing techniques and new directions // Computer Communication Review – Proceedings of the 2002 SIGCOMM conference. 2002. No 32. P. 161–174.

[5] Castillo E., Menedez J. M., Jimenez P. Trip matrix and path flow reconstruction and estimation based on plate scanning and link observations // Transportation Research Part B. 2008. No 42. P. 455–481.

[6] Bianco L., Cerrone C., Cerulli R., Gentili M. Locating sensors to observe network arc flows: exact and heuristic approaches // Computers and Operation Research. 2014. No 46. P. 12–22.

[7] Minguez R., Sanchez-Cambronero S., Castillo E., Jimenez P. Optimal traffic plate scanning location for OD trip matrix and route estimation in road networks // Transportation Research Part B. 2010. No 44. P. 282–298.

[8] Bierlaire M. The total demand scale: a new measure of quality for static and dynamic origin–destination trip tables // Transportation Research Part B. 2002. No 36. P. 282–298.

[9] Krylatov A. Yu. Optimal strategies for traffic flow management on the transportation network of parallel links // Vestnik of St. Petersburg State University. Series 10. Applied Mathematics. Computer Science. Control Processes. 2014. No 2. P. 121–130.

[10] Patriksson M. The Traffic Assignment Problem: Models and Methods. Utrecht, Netherlands: VSP, 1994. 223 p.

[11] Wardrop J. G. Some theoretical aspects of road traffic research // Proc. Inst. Civil Eng. 1952. Vol. 1, No 3. P. 325–362.

[12] Zakharov V. V., Krylatov A. Yu. Transit Network Design for Green Vehicles Routing // Advances in Intelligent Systems and Computing. 2015. Vol. 360. P. 449–458.

[13] Zakharov V. V., Krylatov A. Yu. Competitive routing of traffic flows by navigation providers // Automation and Remote Control, 2016. Vol. 77, No 1. P. 179–189.

[14] Shavidze G. G., Balykina Y. E., Lejnina E. A., Svirkin M. V. Mathematical Model of Ambulance Resources in Saint-Petersburg // Proceedings of the International Conference on Numerical Analysis and Applied Mathematics. 2015. Vol. 1738.

[15] Krylatov A. Yu. Network flow assignment as a fixed point problem // Journal of Applied and Industrial Mathematics. 2016. No 10 (2). P. 243-256.

[16] Belolipetskiy A.A., Gorelik V. A. Economic and mathematical methods Moscow, Russia: Academiya, 2010. 368 p.

Design of an Asynchronous Processor with Bundled-data Implementation on a Commercial Field Programmable Gate Array

Jukiya Furushima, Masamitsu Nakajima and Hiroshi Saito
University of Aizu, Aizu-Wakamatsu 965-8580, Japan
E-mail: {m5201118, m5191117, hiroshis}@u-aizu.ac.jp

Keywords: asynchronous circuits, FPGAs, processors

Received: November 18, 2016

In this paper, we propose a modeling method and a design flow to design asynchronous processors with bundled-data implementation on commercial Field Programmable Gate Arrays (FPGAs). The modeling method mainly concerns modeling of an asynchronous control circuit on commercial FPGAs. In addition to the use of a design environment provided by FPGA vendor, the design flow includes constraint generation, timing analysis, and delay adjustment to design asynchronous processor from a prepared model to FPGA programming. In the experiments, we design three asynchronous MIPS processors. Comparing with the synchronous counterpart, one of them reduces global cycle time which results in 13.8% performance improvement and another one reduces energy consumption 9.3% for a multiplication and 8.8% for a matrix multiplication.

Povzetek: Opisan je razvoj novega sinhronega procesorja na osnovi tržnega FPGA.

1 Introduction

Field Programmable Gate Arrays (FPGAs) are reconfigurable circuits where circuit structure can be changed by designers freely. Therefore, compared to Application Specific Integrated Circuits (ASICs), the lifetime of FPGAs is long. In addition, the design cost is low because FPGA vendors provide the design environment free of charge. Recently, due to the advance of the FPGA technology, FPGAs are well adopted in embedded systems and servers for data centers [1]. As there are a rich number of resources on FPGAs, we can accelerate performance by implementing Multi-Processor System-on-Chip (MPSoC). Current advanced FPGAs such as Altera Cyclone V include an ARM Cortex processor as a hard-macro to support MPSoC.

Most of commercial FPGAs are synchronous circuits. Circuit components in synchronous circuits are controlled by global clock signals. In synchronous circuits, clock skew, power consumption, and electromagnetic radiation will be significant problems when the semiconductor submicron technology is advanced more and more. In addition, generally, the power efficiency of FPGAs is worse than ASICs. Therefore, low power designs on FPGAs are very important.

Compared to synchronous circuits, circuit components in asynchronous circuits are controlled by local handshake signals. Due to the absence of global clock signals, asynchronous circuits are potentially low power consumption and low electromagnetic radiation. Therefore, asynchronous circuits may be useful for FPGAs where low power design is important. However, the design of asynchronous circuits is more difficult than the design of synchronous circuits. To represent circuit behaviors, circuit model including delay model, data encoding scheme, and handshake protocol should be considered. Based on the considered model, asynchronous circuits are designed. In addition, asynchronous circuit designs are also difficult for commercial FPGAs because the design environment provided by FPGA vendors is assuming synchronous circuit designs.

There are many approaches to design asynchronous circuits on commercial FPGAs [2, 3, 4, 5, 6]. Tranchero proposed a design method to design asynchronous circuits on commercial FPGAs in [2]. Ho et al. described to implement C-element [7] into a logic block on commercial FPGAs, showed that there is no hazards, and designed a 4-bit adder with the C-element in [3]. We proposed a floorplan method to place asynchronous logics to commercial FPGAs. All of these literatures address neither design constraint generation (e.g., the maximum delay constraints for paths) nor timing verification (i.e., whether correct timing to control resources is guaranteed or not). We also proposed a design method for asynchronous circuits with bundled-data implementation like this paper in [5]. However, it does not target asynchronous processors. As modeling, constraint generation, and timing verification of processor designs are different, we need a design method to implement asynchronous processors on commercial FPGAs. Minas, et. al., proposed an asynchronous processor with the concurrent error detection scheme to detect transient errors in [6]. It was implemented on an commercial FPGA. On the other hand, modeling, constraint generation, and timing verification described in this paper are not addressed in [6].

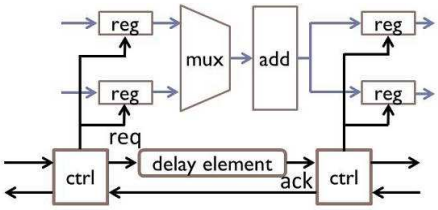


Figure 1: Circuit structure of asynchronous circuits with bundled-data implementation.

In this paper, we propose a modeling method and a design flow to design asynchronous processors with bundled-data implementation on FPGAs. We address how to implement an asynchronous control circuit on the commercial FPGAs, how to synthesize the asynchronous processor with the generation of design constraints, and how to carry out timing verification correctly. We design three pipelined MIPS processors using the proposed method and design flow to evaluate area, execution time, dynamic power, and energy consumption.

The rest of this paper is organized as follows. In section 2, we describe asynchronous circuits with bundled-data implementation. In section 3, we describe about FPGAs. In section 4, we describe the proposed modeling method and design flow. In section 5, we describe the experimental results by designing three MIPS processors. Finally, in section 6, we conclude this work.

2 Asynchronous circuits with bundled-data implementation

Asynchronous circuits with bundled-data implementation shown in Fig.1 are one of data encoding schemes in asynchronous circuits. Timing of data operations is guaranteed by delay elements on request signals. Therefore, the performance depends on the delay of the control circuit. Compared to other implementations such as dual-rail implementations [7] where one bit signal is represented by two wires and the completion detector is required, bundled-data implementation can be realized easily because we can use the same data-path resources as synchronous circuits. In addition, the circuit area and the power consumption become smaller and lower than other implementations.

2.1 Circuit model

Figure 2 represents a bundled-data implementation model used in this paper. It is a pipelined processor model with several pipeline stages i. The left side is the control circuit and the right side is the data-path circuit. The data-path circuit consists of Program Counter (PC), Memories (IMEM and DMEM), pipeline registers (pipereg), Decoder, Register File (RF), ALU, and delay elements $wd_{i,k}$ and hd_k. PC stores the address of the instruction memory. IMEM

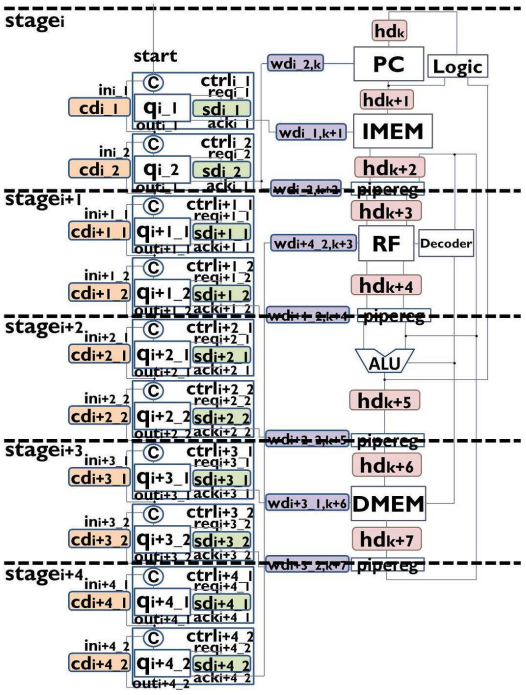


Figure 2: An asynchronous processor model with bundled-data implementation.

is a memory to store instructions. DMEM is a memory to store data. $piperegs$ are registers to separate pipeline stages. RF is a collection of registers. Data from DMEM and ALU are written into RF. $wd_{i,k}$ and hd_k are delay elements for registers or memories to guarantee simultaneous writing constraints and hold constraints.

The control circuit consists of control modules $ctrl_{i\_j}$ ($j = 1, 2$). A control module $ctrl_{i\_j}$ consists of a Q-module q_{i_j} [8], delay elements sd_{i_j} and cd_{i_j}, and a C-element c_{i_j} [7]. sd_{i_j} is used to guarantee setup constraints. cd_{i_j} is used to guarantee control initialization constraints. The C-element c_{i_j} is a synchronization component. The output of the C-element is 0 when all inputs are 0. The output is 1 when all inputs are 1. Otherwise, the output does not change. Logical 1 for the output of the C-element means that the execution at the previous control module and the initialization of the current control module finish.

There are two notes in the control circuit. First, compared to ordinal asynchronous pipelined circuits such as Micropipelines [9] where a feedback signal for the C-element is generated from the output of the C-element in the next control module, the feedback signal in this control circuit is generated from out_{i_j}. This is because to keep the same execution time in all pipeline stages. Second, we use two control modules $ctrl_{i\_1}$ and $ctrl_{i\_2}$ to control a pipeline stage i to hide the overhead caused by handshake signals.

Control modules $ctrl_{i\_j}$ operate as follows. When the execution at the previous control module and the initialization of the current control module finish, c_{i_j} asserts in_{i_j} to trigger the Q-module q_{i_j}. The Q-module q_{i_j} asserts

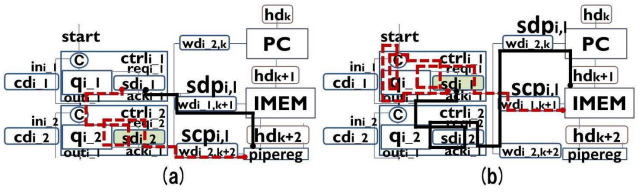


Figure 3: Data-path $sdp_{i,l}$ and control path $scp_{i,l}$ for setup constraints: (a) forward path and (b) backward path.

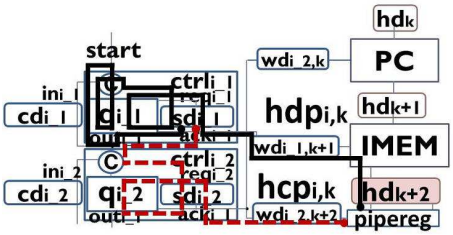


Figure 4: Data-path $hdp_{i,k}$ and control path $hcp_{i,k}$ for a hold constraint.

req_{i_j}. After the signal passes to sd_{i_j}, it is returned to q_{i_j} with the assertion of ack_{i_j}. Then, the Q-module q_{i_j} deasserts req_{i_j}. After the signal passes to sd_{i_j} again, it is returned to q_{i_j} with the deassertion of ack_{i_j}. The deassertion of ack_{i_j} asserts out_{i_j} to move the control to the next control module. Memories and registers in the data-path circuit are controlled by the output of sd_{i_j}. The initialization of control modules $ctrl_{i\_j}$ starts immediately after out_{i_j} is asserted. It is tuned to the deassertion of in_{i_j} and out_{i_j}. The next operation starts immediately after the execution at the previous control module finishes.

2.2 Timing constraints and cycle time

The bundled-data implementation model used in this paper must satisfy five types of timing constraints, setup constraints, hold constraints, control initialization constraints, and simultaneous writing constraints.

Setup constraints mean that input data of registers must be stable before writing to registers. Figure 3 represents paths related to setup constraints. $sdp_{i,l}$ (solid line) represents a data-path from sd_{i_1} to the destination register $pipereg$ where data is written through the source memory IMEM. $scp_{i,l}$ (dotted line) represents a control path from sd_{i_1} to the destination register $pipereg$ through the control module $ctrl_{i\_2}$. $t_{minscp_{i,l}}$, $t_{maxsdp_{i,l}}$, t_{setup_k}, and $sm_{i,l}$ represent the minimum delay of $scp_{i,l}$, the maximum delay of $sdp_{i,l}$, the setup time for the destination register $pipereg$, and the margin for $t_{maxsdp_{i,l}}$. The setup constraint can be represented by the following equation:

$$t_{minscp_{i,l}} > t_{maxsdp_{i,l}} + t_{setup_k} + sm_{i,l} \qquad (1)$$

If this constraint is violated, we need to adjust the delay element sd_{i_1} or sd_{i_2}.

There are two types of $sdp_{i,l}$. One is a forward path where the source register is controlled by a previous control module as shown in Fig.3(a) and the other is a backward path where the source register is controlled by a next control module as shown in Fig.3(b). We define local cycle time lct_i and global cycle time gct. The local cycle time lct_i is defined for each pipeline stage i in which is equal to the maximum delay of $scp_{i,l}$, $t_{maxscp_{i,l}}$, in pipeline stage i. The global cycle time gct is the maximum lct_i for all lct_i. The global cycle time with input data interval decides the throughput of asynchronous pipelined processors.

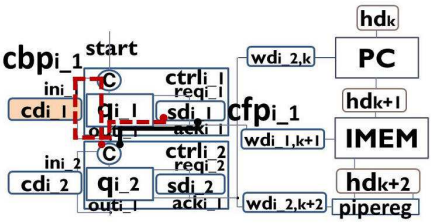


Figure 5: Forward path cfp_{i_1} and backward path cbp_{i_1} for a control initialization constraint.

Hold constraints mean that input data of registers must be stable during writing to registers. Figure 4 represents paths related to hold constraints. $hcp_{i,k}$ (dotted line) represents a control path from sd_{i_1} to the destination register $pipereg$ where data is written. $hdp_{i,k}$ (solid line) represents a data-path from sd_{i_1} to the destination register $pipereg$ through data-path resources. $t_{minhdp_{i,k}}$, $t_{maxhcp_{i,k}}$, t_{hold_k}, and $hm_{i,k}$ represent the minimum delay of $hdp_{i,k}$, the maximum delay of $hcp_{i,k}$, the hold time for the destination register $pipereg$, and the margin for $t_{maxhcp_{i,k}}$. The hold constraint can be represented by the following equation:

$$t_{minhdp_{i,k}} > t_{maxhcp_{i,k}} + t_{hold_k} + hm_{i,k} \qquad (2)$$

If this constraint is violated, we need to adjust the delay element hd_k.

Control initialization constraints mean that the initialization of control modules must be completed after the control signal by the assertion of out_{i_j} reaches to the next control module. Otherwise, the assertion is disabled. Figure 5 represents paths related to a control initialization constraint. cfp_{i_1} (solid line) represents a control path from sd_{i_1} to c_{i_2}. cbp_{i_1} (dotted line) represents a control path from sd_{i_1} to c_{i_2} through q_{i_1}. $t_{maxcfp_{i\_1}}$ represents the maximum delay of cfp_{i_1} and $t_{mincbp_{i\_1}}$ represents the minimum delay of cbp_{i_1}. cm_{i_1} represents the margin for $t_{maxcfp_{i\_1}}$. The control initialization constraint can be represented by the following equation:

$$t_{mincbp_{i\_1}} > t_{maxcfp_{i\_1}} + cm_{i\_1} \qquad (3)$$

If this constraint is violated, we need to adjust the delay element cd_{i_1}.

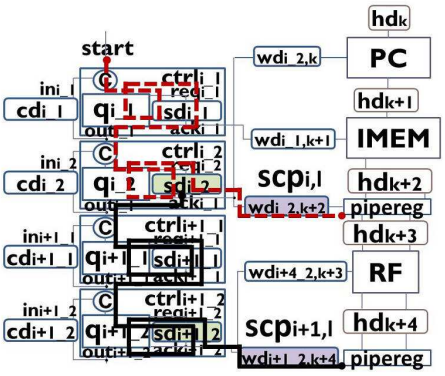


Figure 6: The maximum delays of two control paths $scp_{i,l}$ and $scp_{i+1,l}$ must be nearly equal to each other in simultaneous writing constraints.

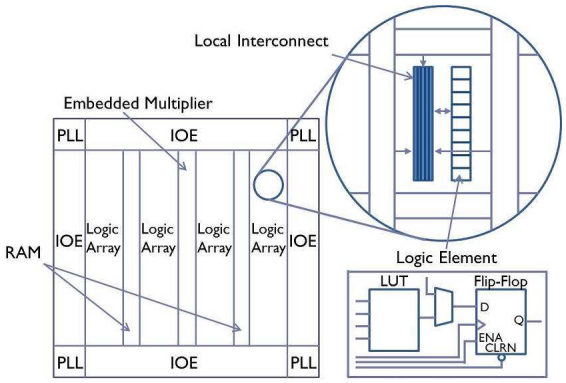


Figure 7: Structure of Altera Cyclone IV FPGA [10].

Simultaneous writing constraints mean that all of registers must be written to the same timing. In pipelined circuits, the throughput depends on the global cycle time gct. Therefore, to delay all of register writing timing until the global cycle time does not affect the throughput. In addition, as a difference of register writing timing may lead to setup/hold violations, to preserve these constraints reduces the occurrence of setup/hold violations. On the other hand, to satisfy simultaneous writing constraints results in behaviors like synchronous circuits. Different from synchronous circuits where global clock signals are used, registers are controlled by different control modules in this circuit model. Therefore, we expect low power consumption for designed asynchronous processors even though these constraints are preserved. Figure 6 represent two control paths $scp_{i,l}$ (dotted line) and $scp_{i+1,l}$ (solid line) for setup constraints. These constraints can be represented by the following equation:

$$t_{maxscp_{i,l}} \simeq t_{maxscp_{i+1,l}} \simeq gct \tag{4}$$

If the above relationship is violated, we adjust delay elements sd_{i_2} and $wd_{i\_2,k}$ for $t_{maxscp_{i,l}}$ and delay elements sd_{i+1_2}, and $wd_{i+1\_2,k}$ for $t_{maxscp_{i+1,l}}$.

3 Field programmable gate array

Field Programmable Gate Array (FPGA) is one of reconfigurable devices. FPGA has been used in many embedded systems because of the advantage such as lower design cost and flexibility to change circuit structure. Figure 7 shows the structure of Altera Cyclone IV FPGA.

The FPGA consists of Logic Array Blocks (LABs), Embedded Multipliers, Random Access Memories (RAMs), Input/output Elements (IOEs), and Phase Locked Loops (PLLs). A logic array consists of 16 logic elements (LEs). A logic element consists of a D Flip-Flop (DFF) and a 4-to-1 Look Up Table (LUT). Any logic function with four inputs can be implemented on an LUT based on a Static RAM. Most of commercial FPGAs has the similar structure like this FPGA.

We use two primitives in Altera FPGAs. One is LCELL and the other is DLATCH. LCELLs are used to implement delay elements such as sd_{i_j} and DLATCHes are inserted after C-elements to carry out static timing analysis correctly with the initialization of C-elements. Both of them are mapped to LUTs.

4 Design of asynchronous processor on commercial FPGAs

In this section, we describe the proposed modeling method and design flow. Even though we target Altera FPGAs in this paper, as there is a similar design environment, we think that we can design asynchronous processors on other FPGAs such as Xilinx FPGAs with the modification of the proposed modeling method and design flow.

4.1 Modeling method

As shown in Fig.2, bundled-data implementation used in this paper consists of a control circuit and a data-path circuit. We use the same data-path resources as the ones used in synchronous circuits. Therefore, we mainly describe modeling of the control circuit.

The proposed modeling method extends the method described in [11] where FPGAs are not considered. Initially, pipeline stages are modeled by a Finite State Machine (FSM) where nodes represent a pipeline stage and edges represent a control flow between pipeline stages. Figure 8(a) represents an FSM for a 5 stage pipelined processor. IF, ID, EX, MEM, and WB represent instruction fetch from the instruction memory, instruction decode, execution, data memory access, and write back to the register file.

Figure 8(b) represents a modeling flow. For each node in the FSM, we split it into two nodes and map control modules ($ctrl_{i\_1}$ and $ctrl_{i\_2}$) . Splitting of nodes is required to hide handshake overhead by two control modules. Then, delay elements (sd_{i_j}), C-elements (c_{i_j}), feedback loops

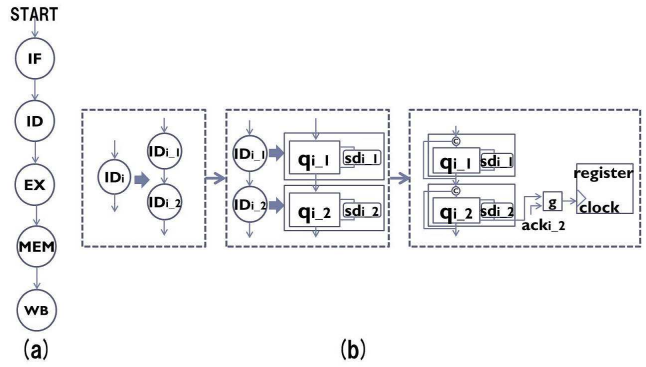


Figure 8: Modeling flow of control circuit: (a) FSM and (b) generation of a control circuit from FSM.

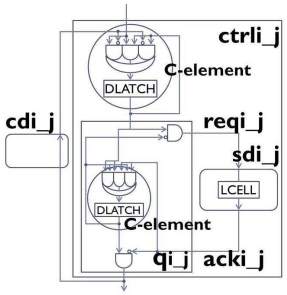


Figure 9: Internal structure of control modules.

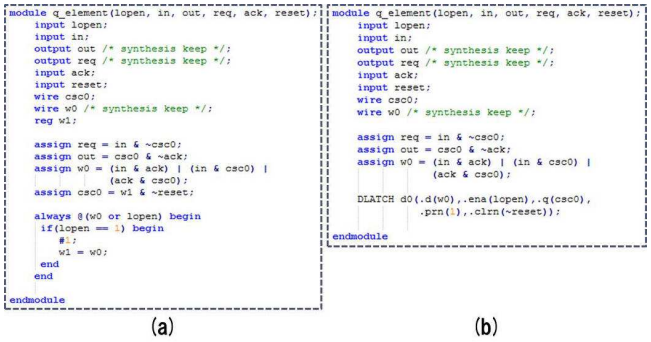


Figure 10: Verilog HDL models of an asynchronous processor: (a) simulation model and (b) synthesis model.

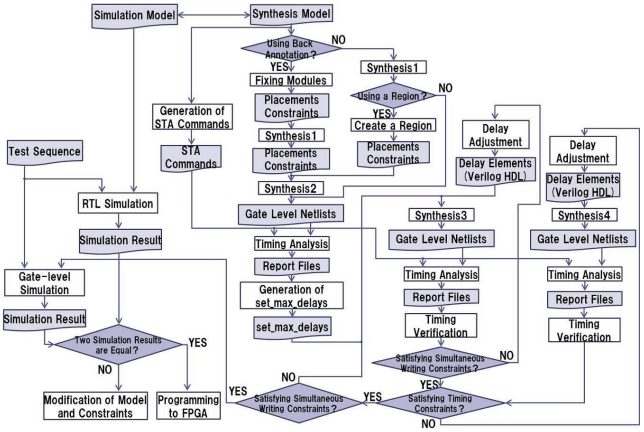


Figure 11: Proposed design flow.

from out_{i_j} to c_{i_j} are inserted. In the modeling, we just insert delay elements sd_{i_j} which consist of one LCELL. All other delay elements are inserted during delay adjustment after synthesis because other delay elements are required when timing constraints such as hold constraints are violated.

Resisters and memories are triggered by ack_{i_j} of corresponding control modules. We insert a glue logic to registers and memories to generate a local clock signal for them from ack_{i_j} and other conditional signals generated from the data-path circuit.

Figure 9 shows the structure of $ctrl_{i\_j}$ for Altera Cyclone IV FPGA. There are two DLATCHes in a $ctrl_{i\_j}$. One is after the C-element c_{i_j} and the other is after the C-element in the Q-module q_{i_j}. They are used to initialize the output of C-elements and to execute static timing analysis correctly. Delay elements sd_{i_j}, hd_k, cd_{i_j}, and $wd_{i\_j,k}$ consist of LCELLs. They work as buffers. To avoid renaming of the output signals of delay elements by the synthesis tool Altera Quartus Prime, we assign $synthesis\_keep$ commands to the output signals of delay elements. Note that we need to avoid logic optimization of control modules and all of delay elements by Quartus Prime. If they are optimized after synthesis by Quartus Prime, we need re-synthesis by assigning $design\_partition$ commands to control modules and delay elements.

Finally, we prepare two models of the bundled-data implementation by Verilog Hardware Description Language (HDL) which is a standard modeling language for FPGAs. The first model is for Register Transfer Level (RTL) simulation before synthesis and the latter model is for synthesis. As RTL simulation does not allow to involve primitive cells DLATCHes and LCELLs, we represent them using logic expressions. Figure 10 (a) and (b) represent the simulation model and the synthesis model for q_{i_j} in control module $ctrl_{i\_j}$ using Verilog HDL.

4.2 Design flow

The proposed design flow uses the design environment Altera Quartus Prime. To design asynchronous processors with bundled-data implementation on commercial FPGAs, we need to consider timing analysis, constraint generation, and delay adjustment for asynchronous processors which are not supported by the design environment.

Figure 11 represents the proposed design flow to implement asynchronous processors on Altera FPGAs. The inputs of the design flow are the simulation model and the synthesis model of an asynchronous processor.

The proposed design flow starts from RTL simulation to check functional correctness for the simulation model

with a test sequence. We use the ModelSim-Altera for logic simulation.

After RTL simulation, we extract all of paths related to setup, hold, and control initialization constraints (i.e., $sdp_{i,l}$, $scp_{i,l}$, $hdp_{i,k}$, $hcp_{i,k}$, cfp_{i_j}, cbp_{i_j}) in the synthesis model. To analyze path delay such as $t_{maxsdp_{i,l}}$ correctly by TimeQuest Timing Analyzer in the Quartus Prime, we generate $report\_timing$ commands and $report\_path$ commands. $report\_timing$ commands are used to analyze path delays between registers to obtain setup and hold times t_{setup} and t_{hold} for registers. $report\_path$ commands are used for other paths. Note that Altera recommends us to set start and end points of paths with primary inputs, registers (flip-flops and latches), and primary outputs. On the other hand, most of paths related to timing constraints in the bundled-data implementation starts or ends by other pins or nets through registers. For example, $sdp_{i,l}$ starts from the output of sd_{i-1_2} to the destination register through the source register. In such cases, we divide paths into sub-paths and prepare $report\_timing$ and $report\_path$ commands for divided sub-paths. For example of $sdp_{i,l}$, a $report\_path$ command is prepared to analyze from the output of sd_{i-1_2} to the source register and a $report\_timing$ command is prepared to analyze from the source register to destination register.

In the design flow, we synthesize bundled-data implementation without any constraints at first (Synthesis1). Then, we decide whether we generate placement constraints or not. There are two possibilities to generate placement constraints. First is to fix the locations of placed resources in the first synthesis (Back Annotate). Second is to prepare a region to place logics of a given processor model (Create a Region). If we use the placement constraints, we carry out the second synthesis (Synthesis2). From the static timing analysis result for the first or second synthesis, we analyze the global cycle time gct of the synthesized circuit. Then, we generate the maximum path delay constraints for all paths with the global cycle time gct so that the global cycle time of iterative synthesis results closes to gct. Then, with the generated constraints, we repeat synthesis and static timing analysis (STA) until simultaneous constraints are satisfied (Synthesis3). Then, we repeat synthesis and and STA until all other timing constraints are satisfied (Synthesis4). If some of timing constraints are violated, we carry out delay adjustment for corresponding delay elements. Finally, through the gate-level simulation for the synthesized processor, we program the synthesized processor on the target FPGA. In the rest of this sub-section, we describe the generation of constraints and the approach for delay adjustment.

4.2.1 Generation of design constraints

Generation of the Maximum Delay Constraints. We assign the maximum delay constraints to all paths related to setup constraints using gct obtained from the STA results by TimeQuest Timing Analyzer in Quartus Prime with

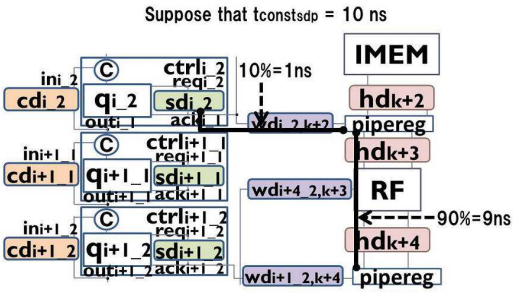


Figure 12: Generation of the maximum delay constraints.

$report\_timing$ and $report\_path$ commands.

From the STA results, first, we analyze local cycle time lct_i for each pipeline stage and global cycle time gct. Second, we decide the margin $sm_{i,l}$ for $t_{maxsdp_{i,l}}$. Third, we decide two parameters $scpmargin$ and $diff$. $scpmargin$ represents a margin between $t_{maxsdp_{i,l}}$ and $t_{minscp_{i,l}}$. Larger $scpmargin$ may result in that setup constraints seem to be satisfied easily. However, it degrades the performance of the synthesized processor because it may lengthen gct after the third synthesis. $diff$ represents the difference between $t_{maxscp_{i,l}}$ and $t_{minscp_{i,l}}$.

The maximum delay constraints for $scp_{i,l}$, $t_{constcp}$, are calculated by the following equation.

$$t_{constcp} = gct \tag{5}$$

The maximum delay constraints for $sdp_{i,l}$, $t_{constdp}$, are calculated by the following equation.

$$t_{constdp} = t_{constcp} - scpmargin - diff - sm_{i,l} \tag{6}$$

As same as $report\_timing$ and $report\_path$, we assign the maximum delay constraints to sub-paths of $scp_{i,l}$ and $sdp_{i,l}$ if these paths include several registers. From the STA results, we decide the ratio of delay for each sub-path. For example, suppose that $t_{constdp}$ for $sdp_{i,l}$ in Fig.12 is 10 ns and the ratio of delay from sd_{i_2} to the source register is 10% of $t_{maxsdp_{i,l}}$ obtained from STA. Then, the maximum delay constraint from sd_{i_2} to the source register becomes 1 ns and the maximum delay constraint from the source register to the destination register becomes 9 ns.

We use set_max_delay commands to represent the maximum delay constraints. We prepare a Synopsys Design Constraint (SDC) file which includes all of set_max_delay commands.

Generation of Placement Constraints. There are two approaches to generate placement constraints. The first approach is to make a region for placement. In the first synthesis report (Synthesis1), we can get the information about the number of used logic elements. From the number of logic elements, we decide a region of FPGA. The region is created by using $LogicLock$ in Quartus Prime.

The second approach is to fix the locations of placed resources in the first synthesis. To realize the second ap-

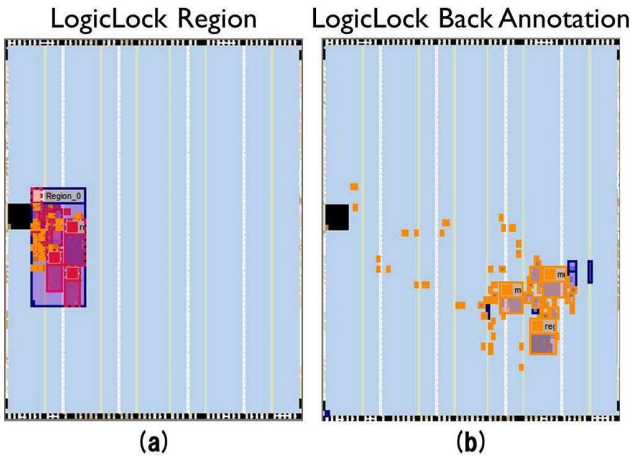


Figure 13: Effect of placement constraints: (a) based on a region and (b) based on the back annotation.

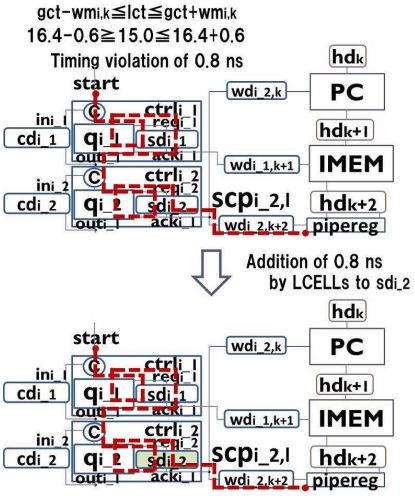


Figure 14: Delay adjustment for simultaneous writing constraints.

proach, we assign $LogicLock$ for each resource (i.e., datapath resources such as registers and control modules) in the synthesis model (Fixing Modules). Through the first synthesis with placement constraints by $LogicLock$, we back annotate the locations of used logic elements, pins, multipliers, and memories in the first synthesis result to a constraint file using Quartus Prime. The locations of resources are represented by $set\_location\_assignment$ commands in the constraint file.

As the second approach fixes the locations of the logic to the first synthesis result, it may reduce the number of delay adjustment. However, it may worse performance if more delay elements are required because the placed logic may affect the placement of newly introduced LCELLs for delay elements. On the other hand, the first approach may allow to place newly introduced LCELLs so that they can be placed freely inside the region. However, it may increase the number of delay adjustments. Figure 13(a) and (b) represent placed logics in the target FPGA when the first and the second approaches are used.

4.2.2 Delay adjustment

From the static timing analysis (STA) result with $report\_timing$ and $report\_path$ commands, simultaneous writing constraints are checked. For a given margin $gctmargin$ for the global cycle time gct, sd_{i_j} or $wd_{i\_j,k}$ are adjusted so that all of $t_{maxscp_{i,l}}$ are within $gct \pm gctmargin$. sd_{i_j} is adjusted if all of $t_{maxscp_{i,l}}$ are out of $gct \pm gctmargin$ while $wd_{i\_j,k}$ is adjusted if only corresponding $t_{maxscp_{i\_j,l}}$ is out of $gct \pm gctmargin$. We generate Verilog HDL models for adjusted delay elements. Figure 14 represents an example of delay adjustment for simultaneous writing constraints.

Next, we adjust setup, hold, and control initialization constraints. As the adjustment of hd_k affects to $sdp_{i,l}$, we adjust hold constraints at first and then we adjust setup constraints reflecting the added or removed delay for hd_k to $sdp_{i,l}$. The adjustment of cd_{i_j} does not affect to the paths related to setup and hold constraints. Therefore, there is no order for delay adjustment between setup (hold) constraints and control initialization constraints. From the STA result using $report\_timing$ and $report\_path$ commands, we assign the delays to both left side and right side of in the inequalities (1), (2), and (3) (see Section 2). By the subtraction of the right side value from the left side value, we add LCELLs to corresponding delay elements to satisfy the constraint if the subtraction result is a negative value (i.e., a timing violation). On the other hand, we remove LCELLs from corresponding delay elements if the left side value is larger than the right side value plus the margin $scpmargin$. Although that the left side value is larger than the right side value means no timing violation, the large left side value results in that gct becomes a large value. Therefore, we remove LCELLs if the left side value overs the right side value plus $scpmargin$. Figure 15 represents an example of delay adjustment for setup constraints.

After we generate Verilog HDL files for adjusted delay elements are generated, we repeat synthesis, STA, and delay adjustment until all of timing constraints are satisfied.

5 Experiments

5.1 Experimental results

In the experiments, we design asynchronous MIPS processor using the proposed modeling method and design flow. We refer to a synchronous MIPS processor in [12] for modeling of asynchronous MIPS processor. Figure 16 represents the block diagram of the MIPS processor. The execution of the synchronous MIPS processor is 5 stage pipeline (instruction fetch, instruction decode, execution,

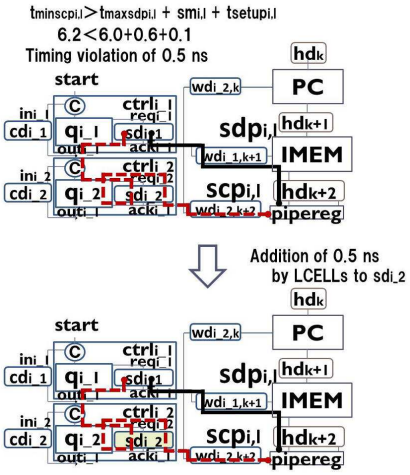


Figure 15: Delay adjustment for a setup constraint.

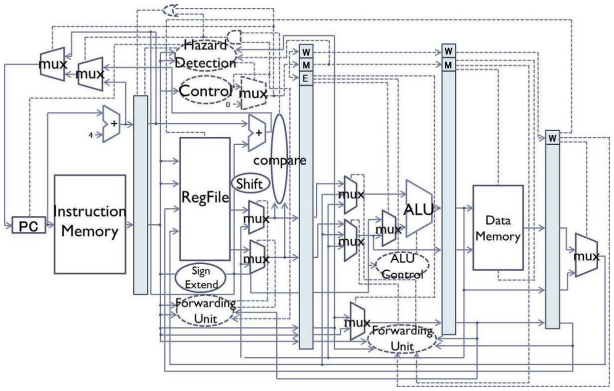


Figure 16: Block diagram of the MIPS processor in [12].

data memory access, and write back). The asynchronous MIPS processor supports 9 instructions (lw, sw, j, beq, add, sub, or, and, slt).

We compare the designed asynchronous MIPS processors with the synchronous MIPS processor in terms of area, execution time, dynamic power consumption, and energy consumption. The used synthesis tool and simulation tool are Altera Quartus Prime ver.15.1 and ModelSim-Altera ver.10.4b. TimeQuest timing analyzer in Quartus Prime is also used to analyze path delays. The target device is Altera Cyclone IV (EP4CE115F29C7).

Initially, we synthesize the synchronous MIPS processor "Sync" by changing clock cycle time so that the clock frequency is maximum. The clock cycle time of "Sync" is 16 ns. Then, we design three asynchronous MIPS processors. "Async1" is the one without placement constraints. "Async2" is the one that the asynchronous MIPS processor is placed inside a region represented by a placement constraint. "Async3" is the one that the locations of all used resources are fixed to the same locations as the first synthesis in the design flow. Table 1 represents parameters for three asynchronous MIPS processors. We decide

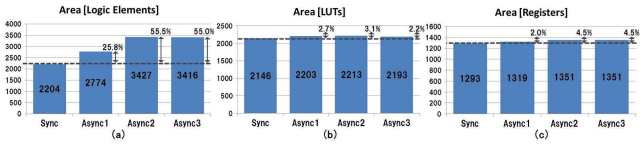


Figure 17: Area of MIPS processors: (a) the number of LEs, (b) the number of LUTs, and (c) the number of registers (DFFs).

Figure 18: Execution time of MIPS processors: (a) multiplication and (b) matrix multiplication.

$gctmargin$, sm_i, $scpmargin$, and $diff$ from the STA results for the first and second synthesis. gct is the value obtained by the STA result for synthesized circuit where all timing constraints are satisfied.

Table 2 represents the number of delay adjustments for three MIPS processors. "Simul" and "Others" represent the number of delay adjustments for simultaneous writing constraints and other timing constraints such as setup constraints.

Figure 17 represents the area of the MIPS processors. Figure 17(a), (b), and (c) represent the number of used LEs, LUTs, and registers (DFFs). These are reported by Quartus Prime. Comparing to "Sync", the increase of LUTs and registers in three asynchronous MIPS processors is less than 5%. However, the number of LEs is increased 25% for "Async1", 54.5% for "Async2", and 54.0% for "Async3". This is because LUTs and DFFs are separated to different LEs even though an LE has one LUT and one DFF.

Figure 18 represents the execution time obtained by gate-level simulation after the designs using ModelSim-Altera. We prepare two test benches, (a) a multiplication and (b) a matrix multiplication. In both cases, compared to "Sync", the execution time is increased 12.5% for "Async2" and 18.8% for "Async3" and decreased about 13.8% for "Async1". This is because the global cycle time of "Async2" and "Async3" is increased and the global cycle time of "Async1" is decreased compared to the cycle time of "Sync" (see Table 1).

Figure 19 represents the dynamic power consumption obtained by PowerPlay Power Analyzer in Quartus Prime assigning a value change dump (.vcd) file generated by gate-level simulation. Compared to "Sync", the dynamic power consumption of "Async1" is increased 22.0% for the multiplication and 22.7% for the matrix multiplication. On

Table 1: Used parameters for asynchronous MIPS processors ([ns]).

name	gct	$gctmargin$	sm_i	$scpmargin$	$diff$
Async1	13.5	1.2	0.6	0.6	1
Async2	17.0	1.2	0.6	0.6	0.9
Async3	18.9	1.2	0.6	0.6	1

Table 2: The number of delay adjustments.

name	Simul	Others
Async1	3	0
Async2	6	0
Async3	3	0

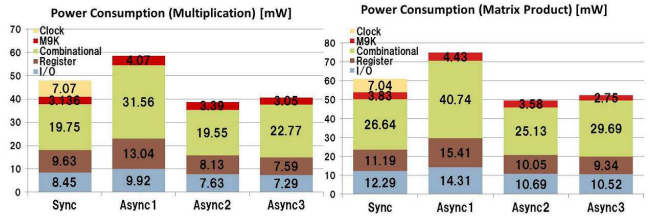


Figure 19: Dynamic power consumption of MIPS processors: (a) multiplication and (b) matrix multiplication.

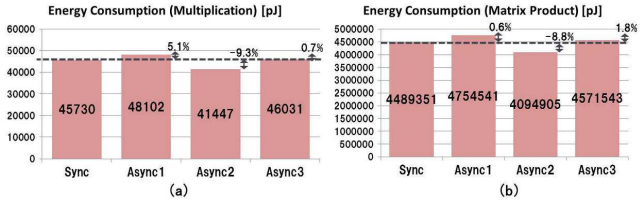


Figure 20: Energy consumption of MIPS processors: (a) multiplication and (b) matrix multiplication.

Table 3: Ratio of dynamic power consumption ([%]).

name	test bench	block	routing
Async1	multiplication	47.4	52.6
	matrix multiplication	46.9	53.1
Async2	multiplication	52.3	47.7
	matrix multiplication	50.9	49.1
Async3	multiplication	46.8	53.2
	matrix multiplication	46.3	53.7

the other hand, compared to "Sync", the dynamic power consumption of "Async2" and "Async3" is reduced 19.4% and 15.3% for the multiplication and 18.9% and 14.2% for the matrix multiplication. As dynamic power consumption depends on frequency which is a reciprocal of the clock cycle time, the longer global cycle time results in the lower dynamic power consumption. On the other hand, the dynamic power consumption caused by the global clock signals (Clock in Fig.19) is reduced in all of asynchronous MIPS processors.

Figure 20 represents the energy consumption which is obtained by the product of execution time and dynamic power consumption. Compared to "Sync", in both multiplication and matrix multiplication, the energy consumption is increased 5.1% and 0.6% for "Async1" and 0.7% and 1.8% for "Async3". On the other hand, compared to "Sync", in both multiplication and matrix multiplication, the energy consumption is reduced 9.3% and 8.8% for "Async2".

5.2 Discussion

The experimental results show that the proposed modeling method and design flow generate two possibilities of asynchronous processors on commercial FPGAs. First it to generate a high performance asynchronous processor like "Async1". As the global cycle time is smaller than the shortest clock cycle time, it increases throughput. To generate the high performance one, we should rely on commercial design environment without placement constraints. Second is to generate a low energy asynchronous processor like "Async2". To generate the low energy one, we should prepare a region to place the logics of processors.

In all of three asynchronous processor designs, there is no big difference for the number of delay adjustments. On the other hand, interestingly, to satisfy simultaneous writing constraints may reduce the possibilities of other timing violations.

To obtain more low power asynchronous processors on commercial FPGAs, we should reduce the number of used logic elements by packing LUTs and DFFs to the same LEs. As mentioned in Figure 17, LUTs and DFFs of three asynchronous processors are separated to different LEs compared to "Sync". This results in the increase of dynamic power consumption due to the use of routing resources such as switches among LABs in which consumes more power. In fact, in all of three asynchronous MIPS processors, the dynamic power consumption caused by routing resources is about half of the total dynamic power consumption as shown in Table 3. To pack LUTs and DFFs to the same LEs results in the reduction of the number of used LEs which in turn the reduction of dynamic power consumption by routing resources. We consider this issue in our future work.

6 Conclusions

In this paper, we proposed a modeling method and a design flow to implement asynchronous processors on commercial FPGAs. Using the proposed modeling method and design flow, we designed three asynchronous MIPS processors. Comparing with a synchronous MIPS processor, one of them reduced the global cycle time which results in 13.8% performance improvement and another one reduced the energy consumption 9.3% for a multiplication and 8.8% for a matrix multiplication.

In our future work, we are going to reduce the number of used logic elements to reduce the dynamic power consumption of routing resources. In addition, we are going to design different asynchronous processors to generalize the proposed method.

Acknowledgement

This work is partially supported by Grant-in-Aid for Scientific Research from Japan Society for the promotion of science (#15K00080).

References

[1] A. Putnam et al., "A Reconfigurable Fabric for Accelerating Large-Scale Datacenter Services", Proc. ISCA'14, pp.13–24, 2014.

[2] M. Tranchero and L. M. Reyneri, "Exploiting synchronous placement for asynchronous circuits onto commercial FPGAs", Proc. FPL, pp.622–625, 2009.

[3] Q. T. Ho et al., "Implementing Asynchronous Circuits on LUT Based FPGAs", Proc. FPL, pp.36–46, 2002.

[4] H. Saito et al., "A Floorplan Method for Asynchronous Circuits with Bundled-data Implementation on FPGAs", Proc. ISCAS, pp.925–928, 2010.

[5] K. Takizawa et al., "A Design Support Tool Set for Asynchronous Circuits with Bundled-data Implementation on FPGAs", Proc. FPL, pp.1–4, September 2014.

[6] Nikolaos Minas et al., "FPGA Implementation of an Asynchronous Processor with Both Online and Offline Testing Capabilities", Proc. Async, pp.128–137, 2008.

[7] Jens Sparso and Steve Furber, "Principles of Asynchronous Circuit Design: A Systems Perspective", Springer, 2001.

[8] F. U. Rosenberger et al., "Q-Modules:Internally Clocked Delay Insensitive Modules", IEEE Transaction of Computer, vol. C-37, no.9, pp. 1005-1018, 1988.

[9] I. E. Sutherland, "Micropipelines", Communications of the ACM, vol.32, issue 6, pp.720–738, 1989.

[10] Altera Cyclone IV FPGA, "https://www.altera.com/products/fpga/cyclone-series/cyclone-iv/overview.html"

[11] S. Iwasaki, "Design and Evaluation of a Low Power Asynchronous AVR Processor considering a Cycle Time Constraint", Master Thesis, the University of Aizu, 2014.

[12] D. A. Patterson and J. L. Hennessy, Computer Organization and Design", Morgan Kaufmann, 2013.

Performance Comparison of Featured Neural Network Trained with Backpropagation and Delta Rule Techniques for Movie Rating Prediction in Multi-criteria Recommender Systems

Mohammed Hassan
University of Aizu, Aizuwakamatsu, Fukushima, Japan
E-mail: d8171104@u-aizu.ac.jp

Mohamed Hamada
University of Aizu, Aizuwakamatsu, Fukushima, Japan
E-mail: hamada@u-aizu.ac.jp

Keywords: multi-criteria recommender systems, artificial neural network, prediction accuracy, backpropagation, delta rule

Received: November 22, 2016

Recommender systems are software tools that have been widely used to recommend valuable items to users. They have the capacity to support and enhance the quality of decisions people make when finding and selecting items online. Such systems work based on which techniques are used to estimate users' preferences on potentially new items that might be useful to them. Traditionally, the most common techniques used by many existing recommendation systems are collaborative filtering, content-based, knowledge-based and a hybrid-based which combines two or more techniques in different ways. The multi-criteria recommendation technique is a new technique used to recommend items to users based on ratings given to multiple attributes of items. This technique has been used and proven by researchers in industries and academic institutions to provide more accurate predictions than traditional techniques. However, what is still not yet clear is the role of some machine learning algorithms such as the artificial neural network to improve its prediction accuracy. This paper proposed using a feedforward neural network to model user preferences in multi-criteria recommender systems. The operational results of experiments for training and testing the network using two training algorithms and Yahoo!Movie dataset are also presented.

Povzetek: Opisana je primerjava več metod, tudi nevronske mreže, za napovedovanje uspešnosti filmov z večkriterijskim priporočilnim sistemom.

1 Introduction

Recommender systems are intelligent systems that play important roles in providing suggestions of valuable items to users. The types of suggestions given by the systems can be of different forms depending on the domain of recommendations. For example, in a movie recommendation problem such as Netflix[1], the systems can suggest the kinds of movies to watch. Similarly, music can be recommended to users in a music recommender systems like Pandora[2], or items to buy can be recommended in Amazon[3], or personalized online news recommender systems like Google-News[4] can recommend news for users to read [1, 2, 3, 4]. Recommender systems are classified based on the technique used during their design and implementation. Traditionally, collaborative filtering, content-based, knowledge-based, and a hybrid-based filtering are the commonly used techniques to design recommender systems. Therefore, knowing the recommendation techniques is at the heart of our understanding of recommender systems. Those techniques are sometimes called traditional techniques, and are increasingly becoming popular ways of building recommender systems [5].

However, despite their popularity and ability to provide considerable prediction and recommendation accuracies, they suffer from major drawbacks [6, 7, 8] because they work with just a single rating, whereas most of the time the acceptability of the item recommended may depend on several item's attributes [9]. Researchers have suggested that if ratings provided to those several characteristics of items would be considered during the prediction and recommendation process, it could help to enhance the quality of recommendations since complex opinions of users will be captured from various attributes of the item. Recent developments in this field have led to the existence of a new

This paper is based on Mohammed Hassan & Mohamed Hamada, Rating Prediction Operation of Multi-criteria Recommender Systems Based on Feedforward Network, published in the *Proceedings of the 2nd International Conference on Applications in Information Technology (ICAIT-2016)*.

[1] https://www.netflix.com/
[2] www.pandora.com
[3] https://www.amazon.com/
[4] https://news.google.com/

recommendation technique known as the multi-criteria recommendation technique [6, 9] that exploits multiple criteria ratings from various items' characteristics to make recommendations. This technique has been used for a wide range of recommendation applications such as recommending products to customers [11, 10], hotel recommendations for travel and tourism [12], and so on. Nevertheless, having considered multi-criteria techniques as the answer to some of the limitations of traditional techniques, it is also logical to look at various ways of modeling the multiple ratings to enhance the prediction accuracies and recommendation qualities. However, few researchers have been able to advance on systematic research into improving the prediction accuracy [13]. In addition, no previous research has investigated the effect of using artificial neural networks to model users' preferences in order to improve the prediction operations of multi-criteria recommender systems [9]. Therefore, as an attempt to investigate the effectiveness of applying neural network techniques for improving prediction accuracies of multi-criteria recommender systems, this study seeks to examine the performance of backpropagation and delta rule algorithms to train the network using a multi-criteria rating dataset for recommending movies to users based on four attributes of the movies. This paper has been divided into five sections including this introduction section. The second part of the paper gives a brief literature review. The experimental methodologies are contained in the third section while the fourth section displays the results and discussion and the final section is concerned with the conclusion and presenting future research work.

2 Related background

2.1 Multi-criteria recommender systems

To be able to understand the concept of recommender systems, some mathematical notations U, I, δ, and ψ to represent the set of users, the set of items, a numerical rating, and a utility function are introduced respectively. The notation δ is the measure of the degree to which a user $\mu \in U$ will like $\iota \in I$, while the utility function ψ is a mapping from a $\mu \times \iota$ pair to a number δ, written as $\psi : \mu \times \iota \mapsto \delta$. The value of δ is a number within a specifically defined interval such as between 1 and 5, 1 and 13, or it can be represented using non-numerical values such as "like", "don't like", . . ., "strongly like", true or false, and so on [14]. Therefore, recommender systems try to predict the value of $\delta\ \forall \iota \in I$ that have not been seen by μ and recommend those with a high value of δ.

The methods of prediction and recommendation explained in the above paragraph are the mechanisms followed essentially by traditional recommendation techniques. Moreover, a similar approach is followed by the multi-criteria recommendation technique with the distinction that it uses multiple values of δ for each $\mu \times \iota$ pair. In the multi-criteria technique, the utility function ψ can be generally defined using the relations in equation 1.

$$\psi : \mu \times \iota \mapsto \delta_0 \times \delta_1 \times \delta_2 \times ... \times \delta_n \quad (1)$$

It is important to note however, there are n ratings in the above equation with the additional rating δ_0 called the overall rating which needs to be computed based on the other n values as in equation 2.

$$\delta_0 = f(\delta_1, \delta_2, \delta_3, ..., \delta_n) \quad (2)$$

The technique can work even without taking δ_0 into account so that there is no overall rating, only ratings of other attributes will be used to undertake the operation process. However, evidence observed from many researchers confirmed the greater efficiency of considering the overall rating rather than ignoring it [6]. The two common approaches used to model multi-criteria rating recommenders are heuristic-based approach that uses certain heuristic assumptions to estimate the rating of an individual item for a user, and a model-based approach that learns a model to predict the utility and recommends unknown items. This classification leads to grouping the multi-criteria rating algorithms into model- and heuristic-based algorithms. For the sake of this experiment, we only need to understand one model among model-based approaches known as the aggregation function model, but nonetheless, for a detailed explanation of the two categories of multi-criteria rating algorithms readers can refer to [8, 9].

The aggregation function approach starts by selecting and training a function or model (such as a neural network) to learn how to predict the overall rating from the criteria ratings. Secondarily, the multi-criteria problem will be decomposed into traditional recommendation problems so that missing ratings for each criterion can be treated as a single rating problem. Finally, the system uses the trained model and the single rating recommenders to predict the overall rating as in equation 2.

2.2 Artificial neural network model

An artificial neural network is one of the most powerful classes of machine learning models that can learn complicated functions from a data to solve many optimization problems. It aimed to mimic the functions of biological neurons that receive, integrate, and communicate incoming signals to other parts of the body [15]. Similarly, the artificial neural network contains sets of connected neurons arranged in a layered style (see Figure 1), where the input layer consists of neurons that receive input from the external environment and the output layer neuron receives the weighted sums of the products of input values and their corresponding weights from the previous layer and sends its computational result to the outside environment.

The features x_1, x_2, x_3, and x_4 in Figure 1 are inputs presented to the input layer, the parameters $\omega_0, \omega_1, \omega_2, \omega_3$, and ω_4 are the synaptic weights for links between the input and output nuerons. $\sum$ is the weighted sum of $\omega_i x_i$ for $i \in [0, 4]$, x_0 is a bias, and f is an activation function that

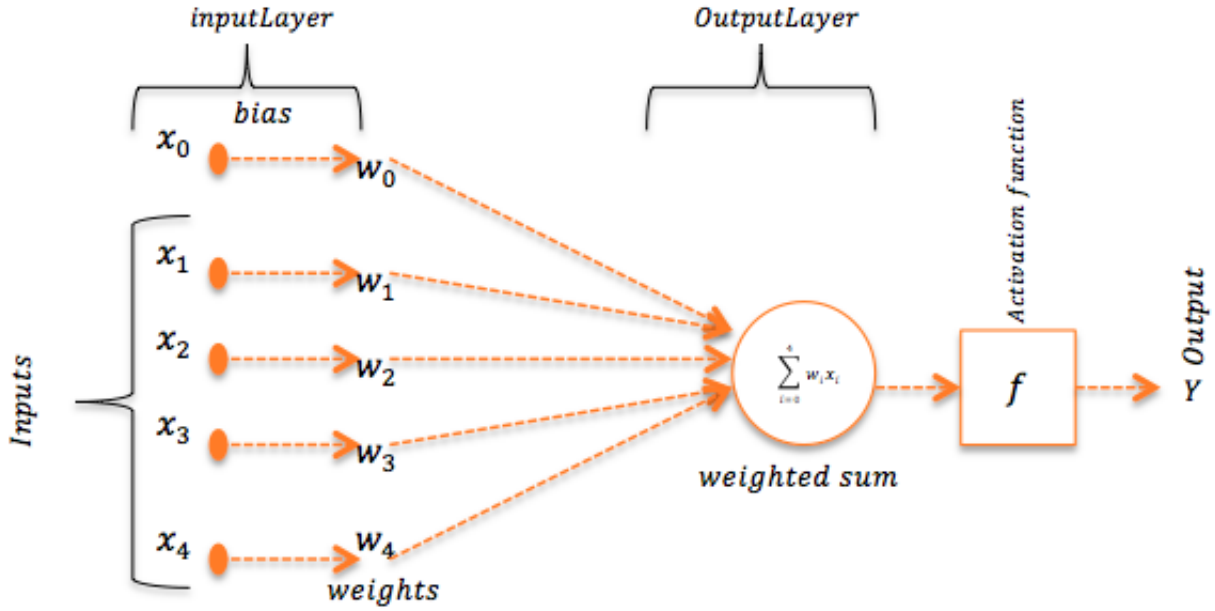


Figure 1: A Single Layer Neural Network

estimates the output Y written as $f(\sum_{i=0}^{4} \omega_i x_i)$. A feedforward network may contain more than two layers, where hidden layer(s) can be added between the input and output layers. The second experiment uses an extended version of the network presented in Figure 1 by adding one hidden layer between input and output layer. This is a brief explanation of how the neural network behaves; details of its learning process can be found in several machine learning books and articles [15, 16].

3 Experiments

The experiment was performed using Yahoo!Movies[5] datasets obtained from [10] for a multi-criteria movie recommendation system where movies are recommended to users based on four characteristics of movies, namely, action, story, direction, and visual effect of the movie which are represented as c_1, c_2, c_3, and c_4 respectively. In addition to those four criteria, an additional rating c_o, called overall rating criterion, was used to represent the final user's preference on a movie. The criteria values (ratings) in the dataset were initially presented using a 13-fold quantitative scale from A^+ to F representing the highest and the lowest preferences of the user respectively. In the same manner, we changed the rating representation to numerical form (13 to 1 instead of A^+ to F).

Table 1 consists of three parts: namely, *Original, Modified, and Normalized* datasets, where the first part displays the sample of the original dataset extracted, and the second part of the table is the same sample of the dataset modified into numerical ratings. Finally, for the network models to work faster and more efficiently, the numerically transformed dataset was normalized to real numbers between 0 and 1 through dividing each of the modified ratings by 13 (since 13 is the highest) as displayed in the last part of the same table. The dataset was well cleaned to avoid cases of uncompleted entries where ratings to some criteria will be missing, and also cases of users who rated few movies (less than five movies). Movies rated by a small number of users were removed completely from the dataset. This data cleaning process reduced the size of the dataset to a total of approximately 63,000 ratings sets. The dataset was divided into training and test data in a ratio of 75:25 for the two experiments. The target of the study was to use a feedforward network to learn how to estimate c_o from c_1, c_2, c_3, and c_4.

Two feedforward networks were developed using object oriented programming techniques in java [18] with learning capacities in delta rule and backpropagation. The Adaline network consists of an input and output layer as in Figure 1 with the input layer containing four neurons and a bias for passing the data to the output layer. The linear activation function f was used in the output neuron to process the weighted sum of the inputs x_i received from the input layer.

Furthermore, in addition to the two layers in the Adaline, a network containing an additional hidden layer with the same number of neurons as the input layers was used for backpropagation training with an additional activation function g (sigmoid function) that receives the weighted sum from the input layer and sends the result of its computation to the output neuron. For measuring the training and test error, mean square error in equation 3 for real output o_j (where $o_j = f(\sum_{i=0}^{5} x_i \omega_i)$) and the estimated output

[5]https://www.yahoo.com/movies/

	UserID	MovieID	Action c_1	Story c_2	Direction c_3	Visual c_4	Overall c_o
Original dataset	1	459	B	A^-	A	A^-	B^+
		554	A^-	A	A	A	A
		554	A^-	A^-	A	A^+	A^-
Modified dataset	1	459	9	11	12	11	10
		554	11	12	12	12	12
		554	11	11	12	13	11
Normalized dataset	1	459	0.692...	0.846...	0.923...	0.846...	0.769...
		554	0.846...	0.923...	0.923...	0.923...	0.923...
		554	0.846...	0.846...	0.923...	1.000	0.846...

Table 1: Sample of extracted and modified dataset

y_j, was used to compute the errors. Pearson correlation coefficient (PCC) presented in equation 4 was also used as a metric for measuring the relative relationship between the real and estimated output for the test data.

$$MSE = \frac{1}{2N}\sum_{j=1}^{N}(y_j - o_j)^2 \tag{3}$$

$$PCC = \frac{\sum(y_j - \overline{y})(o_j - \overline{o})}{\sqrt{\sum(y_j - \overline{y})^2}\sqrt{\sum(o_j - \overline{o})^2}} \tag{4}$$

4 Results and discussion

In each of the two algorithms, neurons' weights ω_i were initially generated at random and the network computes the outputs and the corresponding errors (as $\frac{1}{2}(y_j - o_j)^2$). Iteratively, the algorithms search for a set of weights ω_i $i = 0, 1, ..., 4$ that minimize the error. Since the two algorithms are based on gradient descent, the training begins at some points on the error function shown in equation 3 with defined ω_i, and tries to move to the optimal solution (global minimum) of the function. The rate of the movement is always determined by a parameter known as *learning rate* denoted by α which controls how much the $\omega_{i's}$ can be changed with respect to the observed training errors. Therefore, choosing the correct α is paramount since it can greatly influence the accuracy of the models. Deciding on the best value of α is not always obvious from the beginning of the experiment, as such, the study began by testing various values between 0.1 and 0.001 to find the one that could relatively produce the smallest error. The entire experiment was carried out using $\alpha = 0.007$, which produced the optimal error. The adaptive linear neuron (Adaline) network trained using delta rule shows a quick convergence within a few number of iterations (about 10 iterations) with a very good performance. On the other hand, the backpropagation algorithm prolongs the learning process where a large number of training cycles (epochs) have been used to monitor its performance and the result is presented in Figure 2. This figure shows the average MSE for the various numbers of training cycles. It shows that the convergence can only be attained at a very high number of iterations. However, for the purpose of comparison, the number of training cycles was set to 10,000 cycles (epoch = 10,000), the training error and correlations between the actual and estimated output of the test set for the two algorithms are shown in Table 2. Furthermore, to reaffirm the correlations between test results each of the two models and the actual values from the dataset, Figure 3 shows the curves, one for the actual values from the dataset, and the other two represent the corresponding predicted values by Adaptive linear neuron (Adaline)- and backpropagation (BPA)-based networks. The figure confirmed the accuracy of the Adaline network over the backpropagation-based network.

Table 2: **Performance Statistics**

Algorithm	Number of Iterations	Average Training MSE ($\times 10^{-3}$)	Percentage PCC
Adaline	10	5.34	94.4%
BPA	10,000	7.30	90.0%

5 Conclusion and future work

This study was carried out to investigate the relative performance of single layer and multilayer feedforward networks trained using delta rule and the backpropagation algorithm respectively.

The performance of each model was measured using MSE for the training and the percentage of the correct predictions were evaluated on the test data using Pearson correlation coefficient. From Figure 2 and Table 2, it can be seen that the backpropagation algorithm has a greater demand for longer training cycles to converge.

Moreover, the results indicate that the one layer network trained using the delta rule algorithm is more efficient than the two layer network which supports the tradi-

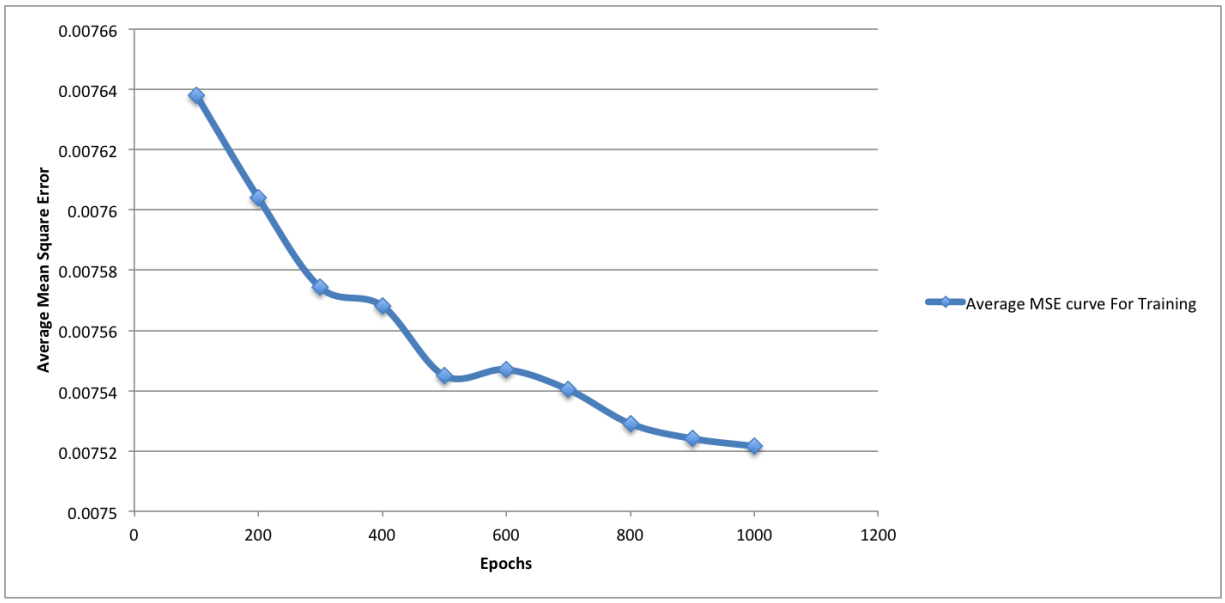


Figure 2: Average training MSE for Backpropagation

tional belief that a single layer network produces less error than a multilayered network [17]. Up to our last experiment with epochs of 10,000, backpropagation did not completely show final convergence, therefore, further investigation is recommended to estimate the approximate epochs required by the algorithm to converge and to know whether it will produce a better result than Adaline. The study confirmed the usefulness of training a neural network model with features of inputs obtained for predicting user preferences on items based on several characteristics of items in multi-criteria recommender systems. Future studies on the current topic are recommended to investigate the performance of more sophisticated neural network architectures and algorithms such as the restricted Boltzmann machine, deep neural networks, convolutional neural networks, and other similar neural networks. However, as the result of this study gives us a hint on the best network architecture and appropriate training algorithm to use, further work is required to extend this research by integrating the model with some popular collaborative filtering algorithms; such as the matrix factorization algorithm that can work on a single rating to predict individual criterion ratings to develop a complete multi-criteria recommender based on Adaline. Furthermore, as the scope of recommender systems covers many application domains like the domain of technology enhanced learning and e-commerce, investigating the effect of neural networks to improve their accuracies is a good direction for future research.

References

[1] Ricci, F., Rokach, L., & Shapira, B. (2015). Recommender systems: introduction and challenges. In *Recommender Systems Handbook* (pp. 1-34). Springer US.

[2] Mahmood, T., & Ricci, F. (2009, June). Improving recommender systems with adaptive conversational strategies, In *Proceedings of the 20th ACM conference on Hypertext and hypermedia* (pp. 73-82).

[3] Park, D. H., Kim, H. K., Choi, I. Y., & Kim, J. K. (2012). A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39(11), 10059-10072.

[4] , Beam, Michael A (2014). Automating the News How Personalized News Recommender System Design Choices Impact News Reception. *Communication Research 41(8)* Sage Publications. pp. 1019–1041.

[5] Hassan, M., & Hamada, M. (2016). Recommending Learning Peers for Collaborative Learning Through Social Network Sites. *IEEE ISMS, Intelligent Systems, Modeling and Simulation*.

[6] Adomavicius, G., & Kwon, Y. (2007). New recommendation techniques for multicriteria rating systems. *IEEE Intelligent Systems, 22(3)*, 48-55.

[7] Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering, 17(6)*, 734-749.

[8] Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web, 10(4),* 415-441.

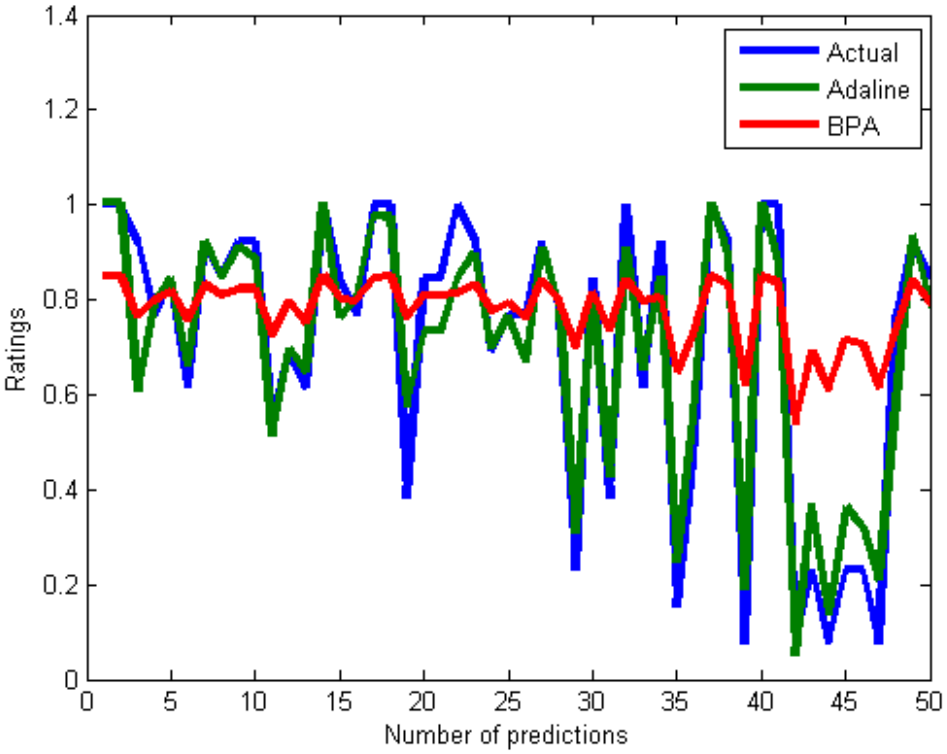


Figure 3: Curve of Actual and some testing results

[9] Adomavicius, G., Manouselis, N., & Kwon, Y. (2015). Multi-criteria recommender systems. *In Recommender systems handbook*, Springer US. (pp. 854–887).

[10] Lakiotaki, K., Matsatsinis, N. F., & Tsoukias, A. (2011). Multicriteria user modeling in recommender systems. *IEEE Intelligent Systems, 26(2)*, 64-76.

[11] Palanivel, K., & Sivakumar, R. (2011). A study on collaborative recommender system using fuzzy-multicriteria approaches. *International Journal of Business Information Systems, 7(4)*, 419-439.

[12] Jannach, D., Gedikli, F., Karakaya, Z., & Juwig, O. (2012). *Recommending hotels based on multi-dimensional customer ratings.* na.

[13] Jannach, D., Karakaya, Z., & Gedikli, F. (2012, June). Accuracy improvements for multi-criteria recommender systems. *In Proceedings of the 13th ACM Conference on Electronic Commerce* (pp. 674-689). ACM.

[14] Ning, X., Desrosiers, C., & Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods.*In Recommender systems handbook* Springer US.(pp. 37-76).

[15] Graupe, D. (2013). *Principles of artificial neural networks (Vol. 7)*. World Scientific.

[16] Wasserman, P. D. (1993). *Advanced methods in neural computing*. John Wiley & Sons, Inc...

[17] Souza, A. M., & Soares, F. M. (2016). Neural network programming with Java. Packt Publishing Ltd.

[18] Kendal, S. (2009). *Object Oriented Programming using Java*. Bookboon.

Agile Methodologies in Software Maintenance: A Systematic Review

Sandhya Tarwani
USICT, Guru Gobind Singh Indraprastha University
Sector-16C, Delhi-110078
E-mail: sandhya.tarwani@gmail.com

Anuradha Chug
USICT, Guru Gobind Singh Indraprastha University
Sector-16C, Delhi-110078
E-mail: anuradha@ipu.ac.in

Keywords: Agile methodology, scrum, extreme programming, software maintenance, quality

Received: November 8, 2016

Agile Methodologies has been gaining popularity since 2000. The Software Maintenance phase of software lifecycle is the most expensive and tedious in nature and use of Agile methodologies helps in maintaining software over time in flexible and iterative manner. This study reviews several papers with different case studies to evaluate the performance and quality of software using agile methodologies. In this study, more than 30 research studies are investigated which are conducted between 2001 and 2015 and have been categorized according to the publication year, datasets, tools, type of techniques etc. This will be the first review paper on the use of the Agile in software maintenance which will help the researchers and encourages companies and beginners to adopt these methodologies to gain software quality. It can be concluded that by adopting agile methodologies it is guaranteed that there will be continuous improvement, greater productivity and enhanced quality of the software. It will also help software development team to finish their work within real time constraints. This study would be helpful to professional academicians also so that they can identify the current trends and future gaps in the field of agile methodologies.

Povzetek: Podan je pregled agilnih metodologij za vzdrževanje programske opreme.

1 Introduction

Software maintenance is the most expensive phase of software development lifecycle. The maximum share of total project costs is being used to maintain software. Achieving the quality of software desired becomes difficult for developers as they often overstep budgetary constraints. Therefore, there is need to find appropriate solution that has the ability to minimize costs. Initially waterfall was used which was sequential in nature and this methodology dominated the world for longest period of time. This model was first cited by Winston W. Royce in 1970[1] when he divided the software development lifecycle in seven sequential and linear stages: Conception, Initiation, Analysis, Design, Construction, Testing and Maintenance. In the early days, waterfall was adopted by various large and small organizations. However, it is important to note that the model inherently has some drawbacks which includes, up-front requirements that increase the cost of maintenance as it become difficult to further change the software. Using this traditional model [2], 70% of the software could not achieve their objective. In a nutshell, the cost of maintenance phase has been tremendously increased as waterfall is sequential in nature making it difficult to move back to the previous stage in the course of maintenance.

Due to all these drawbacks, many organizations are moving towards agile methods for software process improvement. Agile Methodologies were first introduced in the Agile Manifesto [3] which was a summary written and signed in 2001 by Kent Beck also known as Agile Visionary.

Agile methods have gained tremendous success in the commercial industry since late 90's because of following advantages:

- They have up-front requirements.
- It focuses on the developers and customers relationship.
- It includes iterations so that product quality and performance enhances.
- It is iterative in nature which helps the organization to maintain their software in a more flexible and concise manner.
- Agile releases short prototypes after release planning so that users can review it and this continuous monitoring by users help in the maintenance phase.

As various companies had been using waterfall for software maintenance, it was difficult in the early phase to persuade their teams to adopt agile techniques. Wipro technologies were one of those to adopt agile methods. Initially it was difficult to change the mindset of team members but later as the advantages of agile methods came into focus, it becomes their prime focus.[4]. With the advancements in the technology, almost every organization, large and small, is adopting agile at their pace. Customers are more satisfied as the maintenance work consumes lesser time and cost as well as quality of the product has been enhanced. With the rapidness in the product delivery, the transition of maintenance from waterfall to agile methodology environment is increasingly faster. The use of extreme programming in the maintenance environment by Iona technologies [5], showed that by adopting its practices fully or partially in their Orbix project, there was an improvement of 67% although the team size reduced. Visibility was also a factor which plays an important motivational factor in this turnaround.

The aim of this systematic literature review is to summarize, analyze, plan and learn the following things:

(1) Various Agile methodologies for better performance in software maintenance
(2) Comparison of waterfall model and agile methodology lifecycle
(3) The switch from waterfall model to agile methodologies
(4) Various tools available for Agile methodologies
(5) Summarize the strength and weaknesses of Agile Methodologies.

Furthermore, there is a provision of future directions for practitioners.

The rest of the paper is organized as follows: Section 2 presents the research questions and the research criteria for the selection of the studies. It also provides an analysis of the number of research papers available per research questions. Section 3 provides the answer to the research questions identified in this literature survey. Section 4 provides the conclusion and future directions obtained from the systematic survey.

2 Review process

The planning, monitoring and reporting of the systematic literature review paper has been done as per the guidelines given by Kitchenham [6]. As shown in Fig. 1, planning has been done initially to identify the need of this literature review. A review was carried out in order to analyze the work in software maintainability with the help of Agile methodologies. Research studies with respect to their years, case studies, practices, tools used etc, have been investigated so that a trend can be established to find out the pace in this research area. During our survey, it was noticed that papers can be categorized on the basis of year, datasets, tools, techniques, etc. For example, many researchers use private datasets in their studies which make it difficult for the comparison of their performance. After figuring out the needs, keywords were searched for the formation of literature review of Agile methodologies in software maintenance. This was an important step as this is the first review paper in this field.

In the next step, the process of including and excluding case studies was identified. The fourth step involved the formation of the research questions that are being involved and answered in this literature review. In the fifth step, we have analyzed the data.

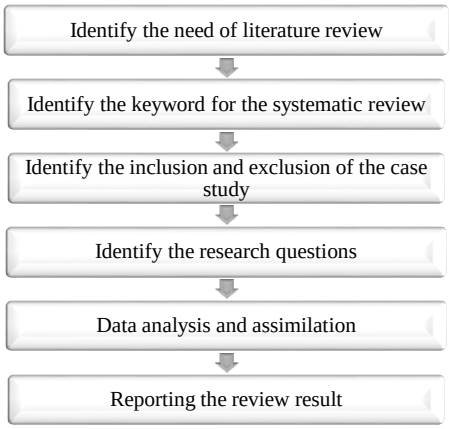


Figure 1: Systematic review process.

2.1 Search keywords strategy

We formed the search terms by using the Boolean expression ‘OR’ and combining main search terms using ‘AND’. The following general search terms were used to collect the data and to form the basis of this literature review:

Agile AND (software OR development OR tool OR testing) AND (XP OR scrum OR lean) AND software (Maintenance OR Maintainability OR Quality OR complexity) AND (quality factors OR reliability OR effects OR refactoring OR metrics). In the table 1, subject along with the search terms are available which can be used to search the various papers including this research review.

Subject	Search terms
Agile methodologies	Agile software, agile development, agile tools, agile testing, XP agile case, agile in small medium companies, agile scrum, agile in software maintainability, extreme programming effects
Software maintenance	Software maintenance, software quality, software complexity, software reliability, software maintenance maturity level, quality factors, refactoring, metrics

Table 1: Search term along with the subjects.

The terms on agile methodologies in software maintenance were derived from textbooks and various research papers. After the identification of the search terms, digital portals were selected and were not restricted only at the home university. The primary source of the literature survey is Google scholars which extracted data from various databases including IEEE Xplore, Wiley online library, ICSR, Science digest, SpringerLink, World Scientific and Digital library.

2.2 Inclusion and exclusion of the study

After identifying the search terms, primary study needed to be selected. There are various studies available in these fields but we needed to apply the inclusion and exclusion so that only important and primary studies would be looked at for the purpose of this review. 30 primary case studies we considered for this review process. The studies were selected after following the inclusion and exclusion criteria given below:

Inclusion criteria:

- Empirical studies using the agile methodologies.
- Empirical study comparing the waterfall and agile methodologies.
- Empirical study combining agile methodologies and Data mining.
- Empirical study using extreme programming, scrum and test driven development.

Exclusion criteria:

- Studies without empirical results of agile methodologies.
- Review studies.
- Web links
- Studies without validation of data.

These inclusion and exclusion criterion helped in the identifying our 30 primary case studies.

2.3 Research Questions

The main focus on the systematical literature review is to answer some of questions which were raised. Table 2 presents 10 research questions addressed during the course of review survey. Firstly, the basic definition of the agile methodologies was identified (RQ1). The second question explains the strengths and weaknesses of agile methodologies (RQ2). RQ3, RQ4 and RQ5 explain the most dominant journal, kind of dataset used and percentage of publications during these years respectively. The issue of transition from waterfall to agile methodologies has been raised in RQ6. The different type of agile methodologies has been analyzed in RQ7. The improvements in software maintenance using extreme programming has being identified in RQ8 and the suitable project size along with the various tools available in market for agile methodologies have been discussed in RQ9 and RQ10.

Research Question	**Main Motivation**
RQ1: What is Agile software development?	Identify the definition of agile in software development
RQ2: What are the strengths and weaknesses of agile methodologies?	Identify the importance of agile methodologies along with their limitations
RQ3: Which journal is dominant in software maintainability using agile methodology?	Identify the most important agile methodologies and its effect on the software maintenance journal
RQ4: What kind of datasets are the most used in various journals?	Identify whether private or public datasets are being used by the researches
RQ5: What is the percentage of publications published during these years?	Identify whether paper published during these years represent large portion of papers in literature or not
RQ6: How does Software Maintenance Team switched from Waterfall to Agile?	Identify team progress from traditional sequential model to more iterative model
RQ7: What are the various sub parts of agile methodologies?	Identify different types of agile methodologies in use today
RQ8: How Extreme Programming practices help to improve performance of software maintenance?	Identify the extent of using extreme programming helps in improving performance
RQ9: Which size project is suitable for the Agile methodologies?	Identify the complexity of introducing agile methods in large, small and medium size
RQ10: What are the various tools available for agile methodologies?	Identify whether the tools available are open or commercial

Table 2: Research questions.

3 Review results

3.1 Agile software development (RQ1)

When Kent Beck investigated the cost of change curve of Barry Boehm [7], Agile Visionary, he observed that curve was more flattened in case of Agile[8,9]. Agile methods are iterative in nature. They uses design-code-test loop which is implemented once a day. Agile mainly balances four variables: Cost, Schedule, Requirements

and Quality. Velocity is also being introduced which is the amount of effort calculated.

Software development is a very tedious job because of evolving technology and there is always a need to develop high quality product [10]. Therefore, Agile methodologies were introduced which minimizes development life cycle. Agile methodologies have various advantages and are easy to learn and implement. Its most important advantage is its light weight characteristic which mainly concentrates on the delivery of high quality product. Extreme programming, one of the most acceptable and widely used agile methodologies, helps the small team organization to change requirements, tight schedules and meet high quality demands [11].

Agile methodology when mapped with the complex adaptive systems and its three dimensions results in the best possible practices [12]. The three dimensions are people, process and product which are not completely independent from each other and hence require identification of all metrics incorporated in all three dimensions. Agile methodology mainly focuses on the rapid iterations and small releases [13] so that users can bring change requirement to notice more rapidly. Serena [13] describes these two methodologies: extreme programming which focuses on the development aspect of software lifecycle rather than managerial aspect and scrum which has its focus on both managerial and development aspects.

Agile architecture can be divided into Product owners and sprintable form [14]. The product owners consist of Up front planning in which architecture is being designed and Story boarding structures the business need. Sprintable form has sprints which build the working software and its functionality by conducting the meetings which reviews and delivers software on time, very frequently. The most important factor influencing agile adoption was personal initiative [15]. As agile requires up front gathering of the requirements therefore turnaround time, software complexity and stability of requirements help in the decision of using agile approaches in business environment. With its challenges and limitations, agile software development has great future scope.

There is always conflict between the formal methods and agile software developments methods [16] because of lack of communication and understanding. Therefore interaction is needed to extract the best practices from both methods.

3.2 Strengths and weaknesses of agile methodologies (RQ2)

The main strength of agile due to which it had gained popularity over traditional and sequential waterfall model is that it is based on the concept of iterations [17]. The user will be able to get the working version of their respective project after each iteration. Based on this it becomes easy for the user to add requirements during the development phase and hence it enhances the flexibility. Even after the design phase has started and user wants to add or change the requirement, they can do so. This is what differentiates it from the waterfall model. In waterfall model, all the requirements have to be submitted at the start of project only. Testing is very important phase of the software life cycle and it needs to be done on a regular basis. With agile methodologies, it becomes easy to continuously integrate and test after every iterations as this method has the provision of continuous integration and constant testing. Developers are in direct contact with the customer which helps them understand the project placing communication at the centre of agile methodologies. With the involvement of the customer, teams inevitably gain motivation and this goes on to enhance the quality [18]. As the number of people in agile team is small, therefore coordination among the team members is easy. Main reason for the introduction of the agile methodologies is that the project needs to be completed on time and stay within the allocated budget something which is granted to the use of this solution.

Figure 2: Strengths of agile methodologies.

Although it has been established that agile methodologies comes with lots of advantages, but it has various weaknesses which must be considered before going head first into software development, so that the quality is not compromised. The major advantage of agile methodologies is the active participation of the customer or user throughout the development lifecycle but this can also leads to the major weaknesses [17]. Sometimes Customer do not have the time to interact. Agile methodologies generally use small teams to develop their projects which can sometimes make it challenging to complete large projects. Team members need to be at the same location throughout their work, but this can get difficult as it is not possible for those teams that work on the different projects and are far away from each other to come together and work at the same physical location. This makes coordination difficult. Also as requirements can be added any time, this will lead to the never ending project.

Miscommunication is the major factor that leads to the problem during the implementation of the agile methodologies in the software development lifecycle. Testing is conducted throughout the software development lifecycle therefore it requires the testers to be at the same place during the lifespan of the project development. This will unnecessarily increase the

resources of the project and increases the overall cost [19]. It becomes difficult to find the pace for the software development. The overall weaknesses of agile are summarized in fig. 3.

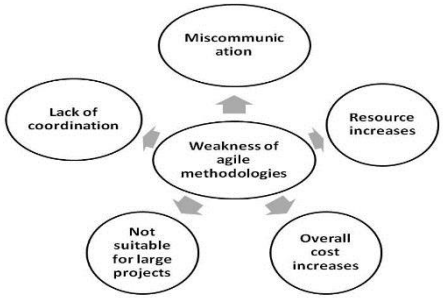


Figure 3: Weaknesses of agile methodologies.

3.3 Dominant journals in this research field (RQ3)

We used more than 35 research studies on software maintainability using agile methodologies. The most dominant research studies along with their ranks are categorised in Table 3.

Rank	Journal	Author
1	IEEE Transactions on Software Engineering	Poole [1]
2	International Journal of computer application	Agarwal [20]
3	International Journal of database theory and application	Upadhyay [21]
4	International Journal of computer Applications	Kumar [22]

Table 3: Most important software maintainability journals using agile methodologies.

3.4 Analysis of datasets used (RQ4)

The biggest difficulty faced during this analysis was the use of unknown and private data sets. Many of the research studies have been written by private firms that used their proprietary data from the analysis work. Papers have been divided into four subsections according to the type of data used for the analysis work: public, private, virtual and unknown.

Public datasets were mostly extracted from the interviews and questionnaire. These includes various case studies like student scientist, maintenance managers etc. They are located at CVS (Concurrent version system) repositories. Various companies volunteered for the analysis providing their projects and case studies and associated private datasets belong to these private companies and not available publicly which includes CSoft developed by Norwegian Software Company[33], projects from Samsung electronics, Orbix projects developed at Iona technologies[5] etc. Virtual datasets have been created by the researchers on their own so as to provide an analysis and proper understanding of the topic. These have been created by SPEM tool and EPF composer editor [23]. These are not included in the public repositories. If no information is available about the datasets, then they have been classified as unknown datasets. As shown in Fig. 4, 34% of papers utilise private datasets. This is what makes them not repeatable and verifiable. On the other hand, 43% of papers have used public datasets. And Only 13% have used virtual datasets.

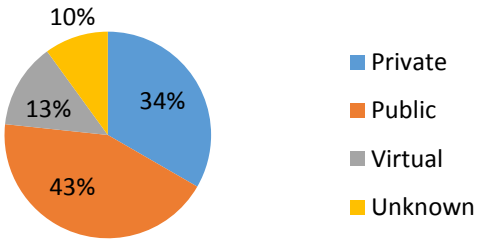


Figure 4: Distribution of datasets.

In table 4, all the information about the case studies and papers considered is provided that have used either private or public or virtual or unknown datasets.

Type of Datasets	Number of paper	Author name
Private	11	Poole [5], Hayes [24], Llieva [11], Szalvay [2], Zanker [25], Serena [13], Sureshchandra [4], Succi [26], Jeanette [37], Jeon [28], Dagnino [10], Hanssen [33],
Public	13	Reyes [29], Abrahamsson [30], Vijayasarathy [15], Saiedian [31], Christensen [32], Mattson [34], Hinchey [16], Thong [35], Mirza [36], Knipper [37], Chakka [38], Qureshi [39]
Virtual	4	Svensson [40], Singh [41], Beeson [42], Piattini [23]
Unknown	3	Huo [43], Jakobsen [44], Choudhari [45]

Table 4: Papers per datasets used.

3.5 Distribution of papers (RQ5)

We examined papers according to their publication year. Papers have been classified into two groups: Paper published before 2008 and paper published after 2008. Fig. 5 shows the distribution of papers regarding publication date.

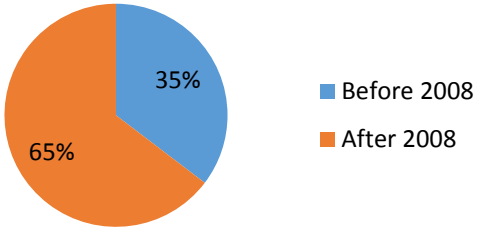


Figure 5: Distribution of papers.

Sixteen papers have been published before year 2008 and twenty nine papers have been published after that. Maximum papers are being published in 2008, IEEE Agile Conference. Fig. 6 shows the type of papers which were published till now. It can be categorized into three groups: journals, book chapters and conference.

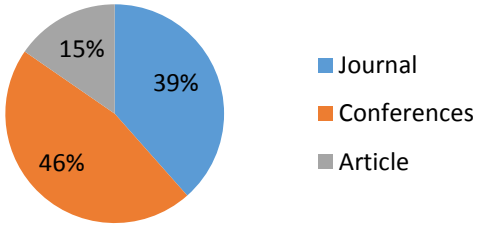


Figure 6: Distribution of type of paper.

3.6 Switching from waterfall to agile (RQ6)

Before 2000, software companies found it convenient to use the traditional waterfall method for the development of software. Due to its shortcomings, the development of software can often not completed on time. Even the maintenance cost associated with the use of this method was increasing. To address these issues, Agile methods in which all the phases overlap and the requirements are gathered in an iterative manner, were introduced. Under this methodology, all the requirements are reexamined at the beginning after each iteration. This minimizes the shortcomings of waterfall model and hence improves the software development process and is more cost efficient. With agile, maintaining software becomes quite easy which enhances the quality as well as reduces the cost. Agile is based on its four factors which include: Cost, Schedule, Requirements and Quality.

The biggest challenge in the world of software development was the conversion of waterfall team into an agile one. The mindset of teams had been set and it was becomes difficult to steer them away from traditional to more modern methods [4]. Basically, the entire project is divided into iterations and product is released iteration by iteration. These projects have phases which are implemented weekly. In the first week, team focuses mainly on the analysis and design. Second week consist of designing and unit testing. The subsequent third and fourth week consists of coding, unit and integration testing of the system. Fifth and sixth phase also involve integration and unit testing of the system. Prior to the start of every phase and iteration, testing is intrinsically considered important. However, what was clear that making the team's bend more towards agile methodologies could not be done without prior planning. Team members who were not able to feel comfortable in this new version of developing software could not continue to be included in the team.

Agile methods provide a faster delivery of product in short span of time and ensure high level of software quality at the same time. This is what plays a crucial role in preferably [43]. With agile practices, the quality of the software also enhances. Agile methods mainly relay on the feedback from the onsite customer who is involved throughout the development process. Pair programming refactoring is used continuously to enhance the productivity, an upgrade from the waterfall method. To check on the quality process, acceptance testing is being used regularly to achieve results successfully. Fig. 7 shows the comparison between waterfall and agile lifecycle methods.

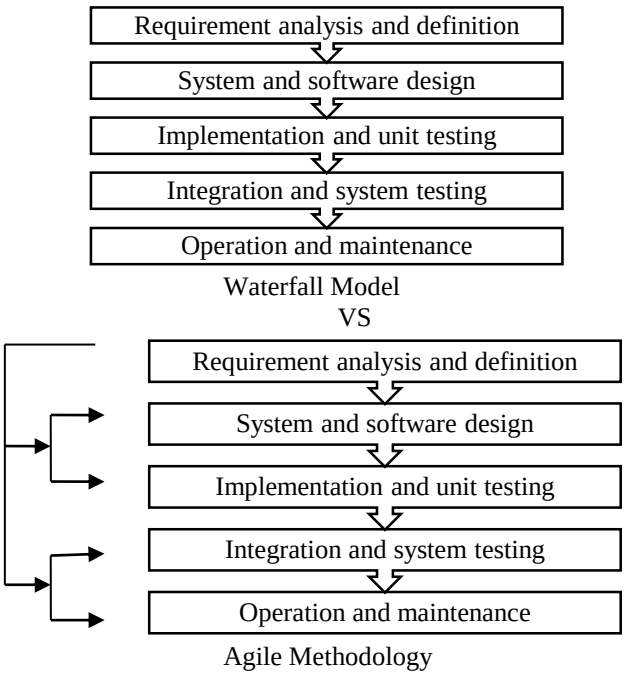


Figure 7: Comparison of waterfall and agile methodologies.

3.7 Types of agile methodologies (RQ7)

Many agile methodologies have been developed so far which helps in developing and maintaining the software at a lower cost. Fig. 8 shows the type of agile methodologies along with their founder name.

Extreme programming

Extreme programming is one of the most widely adopted agile methodologies which were created by Kent Beck. It primarily focuses on the development phase rather than the managerial aspect of software projects [13]. XP was mainly designed so that companies and firms could

comfortably accept some of the agile methodologies. A release plan is developed initially. User writes user stories to describe what they want and is part of the developer team. This ensures that all the requirements are being added in accordance and in presence of user. Team breaks the task into iterations and at the end of it; acceptance testing is being performed to satisfy the user.

Scrum

Scrum was applied in 1990's by Ken Schwaber and Mike Beedle. It is agile method which is incremental and iterative and focuses not only on development but at managerial aspects also[13, 20]. In scrum, work is divided into cycles of work called sprints. During each sprint, requirements are prioritised and are also called as user stories. This is done to develop the highest value requirement first for the user[46].

Test driven development

Test driven development relies on the repetition of very short development cycle. Test cases are being generated to provide improvement and limited code is generated to pass the test successfully so that refactoring can be done easily and code can be sent for the acceptable standard[31]. So therefore it is a quality-first approach where developers test cases are written before the functional code itself [2].

Lean

Lean is a production practice that primarily focuses on the expenditure of resourses. It is mainly used to preserve value for the end users who consumes a product or service with less work. Dynamic system development methodology gained popularity to provide a standard for agile framework that was called as Rapid application development. It revolves around the nine principles that includes active user involvement, frequent delivery, integrated testing etc.

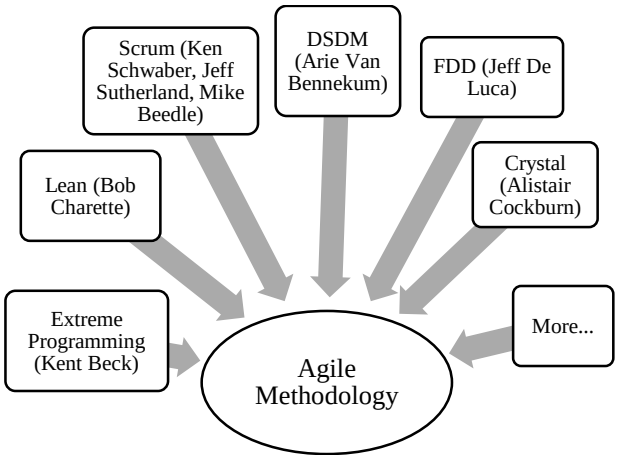


Figure 8: Types of agile methodologies.

Crystal

Crystal is the most light weight agile methodology that consist of agile family such as crystal clear, crystal orange which can be characterised according to the team size and priority.

3.8 Extreme programming practices (RQ8)

Extreme programming is the most successful agile methodology that focuses on the development aspect. XP consists of various practices that help in improving the software in a maintenance phase. Companies adopt full or partial practices of extreme programming to improve the quality of their software. Iona technologies partially use extreme programming practices in their environment [5]. Planning game and simple design is adopted partially. There were small releases of product which provides good productivity and enables the completion of the project within time and budget which was followed religiously. Pair programming in which two developers come together to write code was sparingly adopted. All other practices are fully adopted from starting. Because of the use of extreme programming practices, 67% of improvement is witnessed along with the improvement in the visibility.

Analysis says that when extreme programming practices along with the personal software experts i.e., eXPERT approach is used then there has being significant improvement in productivity, defect rate and effort spent [11]. The introduction of extreme programming in maintenance environment has a positive impact on the project [40]. All the twelve practices could not be able to introduce successfully and those which are assimilated into the project are adapted according to development team environment. Pair programming is one of the most important principles of extreme programming [10]. Pair programming contrast some of the results of empirical evidence as pair programming style has no higher productivity and in many cases it is low in coding standards. Therefore, it is not always necessary to have positive results from the extreme programming principles.

3.9 Suitable project size for agile methodologies (RQ9)

Small companies have a prime focus on the maintenance process; hence Agile_MANTEMA [23] was introduced to help small organizations in the maintenance of their product and to provide services to the customer. Extreme programming was traditionally used in the small organizations but was later extended to be used in medium and large organization [39]. Postmortem analysis and fault rate per KLOC is what is used as a basis for the comparison of traditional and extended extreme programming. The quality of the extended extreme programming is much better because of less fault rate per KLOC.

Agile practices are more easily adapted in less complex organizations [40]. As complexity increases, it becomes mandatory to redesign various methodologies and practices in order to fit in the existing environment.

3.10 Tools available (RQ10)

There are various tools available for agile software development in the industry. Open sources as well as commercial tools are available that helps in the proper implementation of all the advantages of agile methodologies. Free or open sources are agilefant, agile manager, fire scrum, ICEscrum, LeanKit KAnban, Xplanner, etc. These helps the small and medium companies to use these tools and figure out the burndown chart, to make user stories and to estimate the effort left.

Some of the description of open source tools is provided in Table 5.

Tools	Description
Agilefant	With agilefant, management work improves by formation of project burnup and iteration burndown chart. These tools provide three levels of backlogs i.e., product, project and iterations.
Agile manager	It provides designing management also along with the management work.
Fire Scrum	It is based on rich internet application which helps in project management.
ICEscrum	It's a J2EE based tool that helps in the scrum management.
LeanKit	It helps in the customer value and satisfaction.
Xplanner	It is a web based planning and tracking tool and implements with the help of java, jsp etc.

Table 5: Distribution of open tools.

Many companies have built their own agile software which are available to others as well but are paid. Hence, vendors have to buy them according to their needs and demands. These help the organization to make improvements and help others to avail the benefits of these products. These tools are version One, Agile Log, Agilo, ExtremePlanner, etc. Some of the description of these tools is provided in Table 6.

Tools	Description
Version One	It is a simple project management tool which focuses on the centralised version of the management.
Agile Log	It is loaded with tool to effectively manage project and efficiently manages cost.
Agilo	It is a robust platform for managing project.
ExtremePlanner	It has east to use interface that helps the teams to coordinate among themselves easily.

Table 6: Distribution of proprietary tools.

In order to explore further the features of open and proprietary tools, one example of each type is taken and compare in Table 7.

Feature	Agilefant	Agilo
Platform	It works on the Java and MYSQL	It works on Python and RDBMS
Ranking vs prioritization	Priority of 1-5 scale is being given to user stories.	Ranks as well as drag and drop function is available here
Story points	This functionality is not available	It is available here
Iteration burn down chart	Very poor functionality of the chart which decreases the motivation of team	Chart is developed in a good manner, hence enhanced motivation
Epics	Many user stories cannot be encompassed due to lack of this feature	User stories can be encompassed
Portfolio management	Real time overview of budget as well as progress of the project is possible	This feature is not available here
Story themes	It is available in the form of story labels	This feature is not available here
User role	No direct interaction as well as involvement of user	Users are part of teams
Reports	They are available in timesheets only	It saves the reports in customized query
Pros	Story hierarchy is available along with good iteration planning.	It delivers robust platform for the team and helps in coordination with the help of Scrum-teams
Cons	Customization is very poor and external systems integration is not possible	Management of people along with the user interface is not possible

Table 7: Comparison of open source(agilefant) vs proprietary(agilo) tools for implementing the agile methodologies.

4 Analysis

Sixteen journals and twenty four conference proceedings have been evaluated in this review. These were published during the years 2001 to 2015. Fig. 9 shows the curve for a particular year and number of publications developed during its course. It plots year on y-axis and number of publications on x-axis. After reviewing the papers, it was found that maximum papers are published in 2008. There were gradual increases in the beginning of the time period selected till 2008, after which the slope declined for years 2009 and 2010. To date, the work is being continually pursued in this research field to bring improvements in the performance.

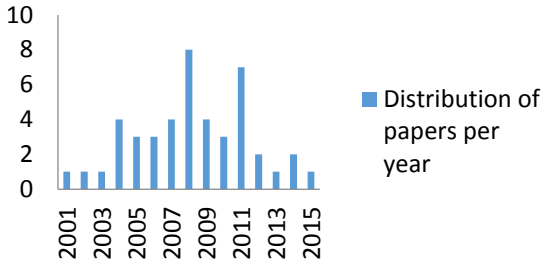


Figure 9: Distribution of papers per year in review.

Fig. 10 and Table 8, describes the number of papers which were used in explaining the answers for the research questions. It is clear that maximum papers have been published to describe what actually agile software development is. Initially, it was stated that design-code-test loop is implemented daily [2]. Agile support high quality delivery of product with upfront gathering of requirements [10]. Small team uses extreme programming in time constraints project [11]. Complex adaptive systems along with the agile software development help in identification of metrics that provides benefits [12]. Serena industries integrated with the agile software development to describe extreme programming and crystals [13]. Agile architecture interactions help in the delivery of working software on time [14]. The factor influencing agile usage and adoption is explained in detail [15]. The basic integration problem between the formal methods and agile methods are explained and extraction of best is done for best practices [16].

[5] Describes to the most dominant paper in this field. Public and private datasets are the closest in number in these research papers. 41% of data used belongs to the private datasets therefore it becomes difficult to compare various case studies. There are various different papers published during these years. 65% of the papers are being published after 2008. Waterfall was used initially but because of these shortcomings, agile was introduced [2]. With agile, quality as well as productivity enhances. The mindset of people was set that's why it was difficult to switch from traditional to iterative methods [4]. Agile provides better productivity and quality to the software as compare to waterfall model [43].

Extreme programming was one of the most acceptable methodologies of agile software development [13]. User stories and acceptance testing is done to provide the better productivity and quality. In scrum methodologies, sprints are being introduced in which daily meetings are held to discuss various requirements [46]. With test driven development, refactoring becomes an easy job [2, 31]. Different extreme programming practices like pair programming, continuous integration etc. are being used fully or partially to improve the productivity [5, 11, 40, 10]. Agile_MANTEMA was introduced to provide benefits to the small companies [23]. Extreme programming was initially meant for small companies but later it was extended for medium and large organization [39]. Less complex systems also makes use agile methodologies [40].

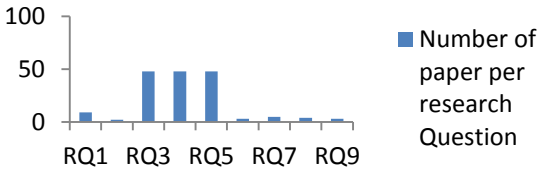


Figure 10: Number of research paper per research question.

Research questions	**Number of papers**	**Authors**
RQ1	9	The Agile Manifesto[3], Szalvay[2], Ilieva [11], Meso[12], Serena [13], Vijayasarathy [15], Hinchey [16], Dagnino[10], Madison [14]
RQ2	2	Mohammad[17], Koch [18]
RQ3	All are included	All are included
RQ4	All are included	All are included
RQ5	All are included	All are included
RQ6	3	Szalvay [2], Sureshchandra[4], Huo[43]
RQ7	5	Serena[13], Majumdar [20], Meng*[46]*, Saiedian [31], Szalvay [2]
RQ8	4	Huisman [5], Ilieva[11], Svensson [40], Dagnino [10]
RQ9	3	Svensson[40], Piattini [23], Qureshi [39]

Table 8: Number of papers per research questions.

5 Conclusion and future directions

Our current survey study is the review of 30 research studies. After observing the evidences from the research studies, it was observed that by introducing agile software development methodologies there has been a continuous improvement in the field of software development. Various methodologies have been used and practiced by practitioners. Agile uses product backlog, sprint backlog and carries work in iterations. The small products are released after every iteration to help the customers to add more requirements according with their needs. As maintenance is very tedious job and is the most expensive phase of the software lifecycle, this has always been a concern under the traditional waterfall approach and is something that's the introduction of agile methodology has addressed in terms of visibly reduced cost.

This helps the organizations to minimize the cost and concentrate on the provision of greater productivity and quality. Extreme programming is the most practiced and used methodology and provides productivity not only in small but also in medium as well as large organizations. There is, however, more research that is required in this field to provide clear path of implementation of agile methodologies in software maintenance.

Some of the future works which can be done in this field are:

- As per the analysis, author's observed that improvement in the pair programming will help the programmer to make up for theory lack of training. Although this is an advantage but still it needs to be incorporated in a company so that it become part of its fabric.
- In future we are planning to compare the quality of a product that can be achieved through the use of waterfall alongside that of agile methodologies.
- To the best of author's knowledge, analysis of detailed metrics should be done with the help of agile methodologies and this analysis could be extended to consider not only the number of defects but also severity.
- There is a strong need for the Private case studies to be replicated with the general cases so that results can be verified.
- Authors are also planning to focus on the refinement of the extreme programming process model with the help of different case studies.
- As far as the author's knowledge is concerned, Comparison of the number of hours required for maintaining the software by the developers using agile as well as some traditional lifecycle modeling has not yet conducted.
- There is a strong need for creation of Automated tools for agile which can be prepared for future refinements in the projects.
- The Authors observed that formalized validation of the data is needed needs so that projects can be validated easily.

References

[1] Walker W. Royce, "Managing the Development of Large software system" *in Proc. IEEE WESTCON*, Los Angeles, IEEE Computer Society Press, 1970, pp. *328-338*.

[2] Victor Szalvay (2004). *An Introduction to Agile Software Development* [Online]. Available: http://www.danube.com/system/files/CollabNet_IntroToAgile_wp_0710.pdf

[3] Kent Beck et al. *Mainifesto for Agile Software Development* [Online]. Available: http://www.agilemanifesto.org/

[4] K. Sureshchandra and J. Shrinivasavadhani, "Moving from Waterfall to Agile" in Agile Conference, [2008] © IEEE. doi: 10.1109/Agile.2008.49

[5] C. Poole and J.W. Huisman, "Using Extreme Programming in a Maintenance Environment," *Proc. IEEE J. Software* vol. 18, Issue 6, pp. 42-50, Nov. 2001.

[6] B. Kitchenham et al., "Guidelines for performing systematic literature review in software engineering," © 2008 Elsevier B.V, doi:10.1016/j.infsof.2008.09.009

[7] B.W. Boehm, in *Software Engineering Economics*, 1st ed. USA: Prentice Hall PTR Upper Saddle River, NJ, 1981.

[8] K.Beck, in *Extreme Programming explained: embrace change*, USA: Addison-Wesley Longman Publishing Co., Inc. Boston, MA, 2000.

[9] S.W. Ambler ," Examining the cost of change curve," in *The Object Primer: Agile Model-Driven Development with UML 2.0*, 3rd ed. USA: Cambridge University Press, 2004.

[10] A.Dadnino ,"An Evolutionary lifecycle Model with Agile practices for software development at ABB" in *ICECCS '02 Proc. 8th Int. Conf. on Engineering of complex computer systems*,© IEEE computer Society, USA, pp. 215.

[11] S. IIieva et al., "Analyses of an agile methodology implementation," in Proc. *30th EUROMICRO Conference*, ©IEEE, 2004, pp. 326-333.

[12] P. Meso, "Agile Software Development: Adaptive Systems principles and Best practices," *Information System Management*, doi: 10.1201/1078.10580530/46108.23.3.20060601/93704.3.

[13] Serena software Inc., *An introduction to Agile Software Development*,[Online], Available: http://www.serena.com/docs/repository/solutions/intro-to-agile-devel.pdf.

[14] J. Madison, " Agile Architecture Interactions," *IEEE Software*, ©IEEE Computer Society,[2010], Vol. 27, no. 2, doi: http://doi.ieeecomputersociety.org/ 10.1109/MS.2010.27.

[15] L.R. Vijayasarathy and D.Turk, "Agile Software Development: A survey of early adopter," *Journal of Information Technology Management*, Vol. 11, no. 2, 2008, pp. 1-8.

[16] S. Black et al., "Formal Versus Agile: Survival of the fittest,", ©IEEE Computer Society, [2009], Vol. 42, no. 09, pp. 37-45.

[17] A.H. Mohammad et al.," Agile Software Methodologies: Strength and Weakness,", *Int. J. of Engineering Science and Technology*, Vol. 5, No. 03, March 2003, pp. 455-459.

[18] A.Koch, " 12 Advantages of Agile Software Development,", ©Global knowledge Training LLC. , [2011], pp. 1-10.

[19] K. Waters, Disadvantages of Agile Development [Online], Available: http://www.allaboutagile.com/disadvantages-of-agile-development/.

[20] M. Agarwal and R. Majumdar," Software Maintainability and Usability in Agile Environment,", *Int. J. of Computer Application*, Vol. 68, No. 4, 2013, pp. 30-36.

[21] P. Upadhyay," Modeling software Maintainibility and Quality Insurance in Agile Environment,", Int. J. of Database Theory and Application, Vol. 7, No. 3, 2014, pp. 83-90.

[22] B. Kumar," The Sway of agile Processes over Software Maintainability,", *Int. J. of Computer Application*, Vol. 109, No. 1, 2015, pp. 25-29.

[23] F.J. Pino et al. ," A software maintenance methodology for small organizations: Agile_MANTEMA,", *J. Software Maintenance and evolution*, ©John Wiley & Sons, Ltd., 2011, pp. 851-876.

[24] J.H. Hayes et al.,"Observe-mine-adopt(OMA): an agile way to enhance software maintainability,", *J. Software Maintenance and evolution*, ©John Wiley & Sons, Ltd.,2003, pp. 297-323.

[25] M. Zanker and S. Gordea," Measuring, monitoring and controlling software maintenance efforts,", *Proc. 13th Int. Symp. on Temporal Representation and Reasoning*, ©2006 IEEE.

[26] M. Najafi and L. Toyoshiba," Two case Studies of User Experience Design and Agile Development,", *IEEE Agile Conf.*, ©2008 IEEE, pp. 531-536.

[27] J.Heidenberg," Towards increased productivity and quality in software development using Agile,Lean and Collaborative Approaches,",Ph. D Dissertation, Dept, of Information Technology, Abo Akademi Univ., Turku, Finland, 2011.

[28] S. Jeon et al.," Quality Attribute driven Agile Development,", *9th Int. Conf. on Software engineering Research, Management and Applications*,©2011 IEEE, doi:10.1109/SERA.2011.24.

[29] W.Reyes, "Agile Approaches to Software Maintenance: An exploratory Study of Practitioner Views,", *Managing worldwide Operations& communications with Information Technology*, ©2007, Idea Group Inc.

[30] H.Hulkko and P.Abrahamsson, "A Multiple Case Study on the Impact of Pair Programming on Product Quality,",*ICSE*,©2005 ACM, pp. 495-504.

[31] D.S.Janzen and H.Saiedian, "Does Test Driven Development Really Improve Software Design Quality?," *IEEE Software*, ©2008 IEEE, Vol. 25, No. 2.

[32] K.R.Schougaard et al., " SA@Work A field Study of Software Architecture and Software Quality at work,",*Proc. 15th Asia-Pacific Software engineering Conf.*, ©IEEE Computer society, pp. 411-418.

[33] G.K.Hanssen, "Maintenance and Agile Development: Challenges, Opportunities and Future Directions,", *Proc. ICSM, Canada*, ©!EEE, pp. 487-490.

[34] M.K.Mattsson and J.Nyfjord, "A Model of Agile Evolution and Maintenance Process,",*Proc. 42nd Hawaii Int. Conf. on System Sciences*,©2009 IEEE.

[35] F.K.Y. Chan and J.Y.L. Thong, "Acceptance of Agile Methodologies: A critical Review and conceptual framework,",*J. Decision Support Systems*, Vol. 46, No. 4, 2009, pp. 803-814.

[36] A. Ahmed et al., "Agile Software Development, impact on Productivity and quality,", *Int. Conf. on Management of Innovation and Technology*,2010©IEEE, doi: 10.1109/ICMIT.2010.5492703.

[37] D.Knippers, "Agile Software Sevelopment and Maintainability,", *15th Twente Student Conf.*,2011,the Netherlands,©University of Twente.

[38] K.N.Rao, " A study of the agile software development methods, applicability and implications in industry,"*Int. J. of Software Engineering and its Applications*,Vol. 5, No. 02,2011, pp. 35-46.

[39] M.R.J.Qureshi, "Agile software development methodology for medium and large projects,", *IET Software*, Vol. 6, No. 4,©2012 The Institution of engineering and technology, pp.358-363.

[40] H.Svensson and M. Host," Introducing an agile process in a software maintenance and evolution organization",*Proc. 9th European Conf. on Software Maintenance and Reengineering*,©2005 IEEE.

[41] M.Singh," U-Scrum: An agile methodology for promoting usability,",*Agile Conf.*, ©2008 IEEE, doi: 10.1109/Agile.2008.33.

[42] K.Opelt and T. Beeson, "Agile teams require agile QA: how to make it work, An experience Report,",*Agile Conf.*,©2008 IEEE,doi: 10.1109/Agile.2008.59.

[43] M.Huo et al., "Software Quality and Agile methods,", *Proc. 28th Annual Int. Computer Software and Applications Conf.*, ©2004 IEEE.

[44] C.R.Jakobsen and K.A.Johnson, "Mature agile with a twist of CMMI in Agile,", *Agile Conf.*,©2008 IEEE,doi: 10.1109/Agile.2008.10.

[45] J.Choudhari and U. Suman, "Iterative Maintenance Life Cycle using Extreme programming,", *Int. Conf. on Advances in Recent Technologies in Communication and Computing*,©2010 IEEE, doi:10.1109/ARTCom.2010.52.

[46] X.Meng et al., "A process pattern Language for Agile Methods,", *14 th Asia-Pacific Software Engineering Conference.*

[47] R.Moser et al.,"Does XP deliver quality and maintainable code?,", *8th Int. Conf., XP*, Como, Italy, June 18-22, 2007.
[48] L.Williams, "A Survey of Agile Development Methodologies", ©Laurie 2007.
[49] T.Mens and T. Tourwe," A survey of software refactoring in Software Engineering,",*IEEE transaction on Software Engineering*,Vol. 30,No. 2, ©IEEE Computer Society,2004,doi: 10.1109/TSE.2004.1265817.
[50] A.Sampaio et al.," Towards reconciling Quality and agility in web application development,", *ICWE Workshops'04*.
[51] J.Prochazka,"Agile support and Maintenance of IT services,",Information Systems Development ©Springer Sciences,doi:10.1007/978-1-4419-9790-6_48.
[52] G.Concas et al., "An empirical study of software metrics for assessing the phase of an agile project,",*Int. J. of Software Enineering and Knowledge engineering*,Vol.22,no. 4,2012,doi: 10.1142/S0218194012500131.
[53] J.Verdugo et al., "Using Agile methods to implement a laboratory for software product quality evaluation,",©Springer International Publication Swizerland 2014, pp. 143-156.
[54] S.Nerur et al., "Challenges of migrating to agile methodologies,",Magazine Communications of the ACM-Adaptive Complex enterprises, Vol. 48,No. 5,pp. 72-78.
[55] R.Moser et al., "Does refactoring improve reusability?,", Proc.9th Int. Conf. on Reuse of Off-the-Shelf Components,©2006 Springer, pp.287-297.
[56] 70+Comprehensive Agile Project Management Tool list,[Online], Available: http://www.softwaretestingclass.com/70-comprehensive-agile-project-management-tools-list/.
[57] Top Agile and Scrum Tools, [online], Available: http://agilescout.com/best-agile-scrum-tools.

Optimizing Parameters of Software Effort Estimation Models using Directed Artificial Bee Colony Algorithm

Thanh Tung Khuat, My Hanh Le
The University of Danang, University of Science and Technology, Danang, Vietnam
thanhtung09t2@gmail.com, ltmhanh@dut.udn.vn

Keywords: Software effort estimation, directed artificial bee colony, optimization, swarm intelligence, estimation models

Received: December 5, 2016

Effective software effort estimation is one of the challenging tasks in software engineering. There have been various alternatives introduced to enhance the accuracy of predictions. In this respect, estimation approaches based on algorithmic models have been widely used. These models consider modeling software effort as a function of the size of the developed project. However, most approaches sharing a common thread of complex mathematical models face the difficulties in parameters calibration and tuning. This study proposes using a directed artificial bee colony algorithm in order to tune the values of model parameters based on past actual effort. The proposed methods were verified with NASA software dataset and the obtained results were compared to the existing models in other literature. The results indicated that our proposal has significantly improved the performance of the estimations.

Povzetek: S pomočjo algoritma umetne čebelje kolonije so optimirani parametri za oceno potrebnega softverskega dela.

1 Introduction

Software effort estimation is the process of predicting the most realistic amount of effort which is usually expressed in terms of person-hours required to develop or maintain software based on incomplete, uncertain and noisy input. This activity has become a crucial task in software engineering and project management. Effort estimation at the early stages of software development is a challenge due to the lack of understanding the requirements and information regarding to the project. Both underestimated and overestimated effort are harmful for projects under development. Underestimation results in a situation where commitments of the project cannot be accomplished because of a shortage of time and/or resources. In contrast, overestimation can lead to the rejection of a project proposal or cause the allocation of an excess of resources to the project [1].

Several techniques for cost and effort estimation have been proposed over the last few decades, and they can be classified into three main categories [2]. These categories comprise:

1. **Expert judgement** [3]: a technique widely used, relies on an expert's previous experience on similar projects to gather, evaluate, discuss, and analyze data concerning a target project to generate an estimation.

2. **Algorithmic models** [4]: also known as parametric models attempt to represent the relationship between effort and characteristics of project. The main cost driver of these models is the software size, usually measured by Kilo Line of Code (KLOC) or function point. This is still the most popular technique in the literature [26]. These models include COCOMO I [6], COCOMO II [7], SLIM model [8], and SEER-SEM [9].

3. **Machine learning**: In recent years, machine learning approaches have been used in conjunction or as alternative to the above two techniques. These techniques in this group consist of fuzzy logic models [10], neural networks [11], case-based reasoning [2], and regression trees [12].

However, none of the aforementioned approaches are complete and can be appropriate in all situations [13]. This study focuses on the algorithmic models and deals with their difficulties in parameters calibration and tuning in order to enhance the accuracy of software effort predictions. The objective of this study is to focus on improving the algorithmic models which were proposed in the literatures of Sheta [14] and Uysal [15] by using the directed artificial bee colony algorithm with several modifications.

Swarm intelligence is the discipline that collectives behaviors from the local interactions of the individuals with each other and with their environment. This discipline also models swarms that are able to self-organize [16]. Examples of systems studied by swarm intelligence are behaviors of real ants [17], schools of fish, flocks of birds [18]. Artificial bee colony (ABC) algorithm [16] is one of the most-studied swarm intelligence algorithms. There have been a lot of improved variants of ABC algorithm used to tackle a wide range of optimization problems. Among these studies, the directed artificial bee colony (DABC) al-

gorithm [19] is a new version of basic ABC. This algorithm is better than the original ABC in terms of solution quality and convergence characteristics [19]. This work, therefore, applies the DABC in order to optimize parameters of algorithmic models for software cost estimation.

Our contributions in this paper include:

- We combine a control parameter and direction information for each dimension of food source position to update position of the current food source in the state-of-the-art of the Directed Artificial Bee Colony Algorithm. We also use a new boundary constraint-handling mechanism for the DABC if the position of the food source exceeds the boundaries of variables.
- We apply the DABC to improve the effort estimation models introduced in recent literature.
- We evaluate the efficiency of the proposed approaches compared with original models.

The rest of this paper is organized as follows. Section 2 briefly represents software effort estimation models in general and algorithmic models in particular. Section 3 shows the DABC algorithm. Experimental results are presented in Section 4 and Section 5 concludes the obtained results of the study.

2 Software effort estimation models

Estimation methods based on algorithmic models are common. Researchers have attempted to derive algorithmic models and formulas to present the relationship between size, cost drivers, methodology used in the project and effort. As a result, an algorithmic model can be expressed as follows:

$$Effort = f(x_1, x_2, ..., x_n) \quad (1)$$

where $\{x_1, x_2, ..., x_n\}$ denote the cost factors. In additon to the software size, there are many other cost factors proposed and used by Boehm *et al* in the Constructive Cost Model (COCOMO) II [20]. These cost factors can be divided into four categories:

- **Product factors**: required software reliability, database size, product complexity.
- **Computer factors**: execution time constraint, main storage constraint, virtual machine volatility, and computer turnaround constraints.
- **Personnel factors**: analyst capability, application experience, programmer capability, virtual machine experience, and language experience.
- **Project factors**: multi-site development, use of software tool, and required development schedule

The existing algorithmic models differ in two aspects: the selected cost factors, and the function f used.

2.1 Algorithmic models

The simplest formula of relationship between effort and input factors is a linear function, which means that if size increases then effort also rises at a steady rate. Linear models have the form:

$$Effort = a_0 + \sum_{i=1}^{n} a_i * x_i \quad (2)$$

where the coefficients $a_0, a_1, ..., a_n$ are selected to best fit the completed project data.

The linear model, nevertheless, is not appropriate for estimates of non-trivial projects in large and complicated environments. Therefore, more complex models were developed. These ones reflected the fact that costs do not normally increase linearly with project size. In the most general form, an algorithmic estimation for software cost can be represented as:

$$Effort = A * Size^B * M \quad (3)$$

where *A* is a constant factor depending on local organizational practices and the kind of software that is developed. *Size* might be either the size of the software or a functionality estimation expressed in function or object points. The value of exponent *B* is normally between 1 and 1.5. *M* is a multiplier formulated by combining process, product and development attributes.

COCOMO which was introduced by Boehm [21] is one of a very famous software effort estimation models using general formula presented in Eq. 3. The COCOMO model is an empirical model that was derived by collecting data from 63 software projects. These data were analyzed to construct a formula that was the best fit to the observations. The formula of the basic COCOMO is shown in Eq. 4.

$$E = A * (KLOC)^B \quad (4)$$

where *E* shows the software effort computed in person-months. *KLOC* stands for Kilo Line of Code. The values of the parameters *A* and *B* depend mainly on the type of software project. There were three classes of software projects that were classified based on the complexity of projects. They are Organic, Semidetached and Embedded models [21].

1. For simple, well-understood applications (Organic): *A* = 2.4, *B* = 1.05
2. For more complex systems (Semidetached): *A* = 3.0, *B* = 1.15
3. For Embedded systems: *A* = 3.6, *B* = 1.2

COCOMO model ignores requirements and documentation, customer skills, cooperation, knowledge, hardware issues, personnel turnover levels at all. Therefore, with regard to the complex projects, the estimated results using COCOMO model are not accurate. Extensions of COCOMO, such as COMCOMO II [20], enhanced the quality

of software estimates. However, this paper does not take into consideration this model.

Another method to improve the quality of the COCOMO model is to complement a methodology (ME) factor used in the software project into the equation to estimate effort. It was also found that adding the ME factor similar to the classes of regression models assists to stabilize the model and reduce the influence of noise in measurements [14]. Software effort estimation model is changed to:

$$E = f(KLOC, ME) \tag{5}$$

where f is a nonlinear function in terms of KLOC and ME.

Sheta [14] presented two various versions for function f as follows:

– **Sheta's Model 1**:

$$E = A * (KLOC)^B + C * ME \tag{6}$$

where $A = 3.1938, B = 0.8209, C = -0.1918$.

– **Sheta's Model 2**:

$$E = A * (KLOC)^B + C * ME + D \tag{7}$$

where $A = 3.3602, B = 0.8116$,
$C = -0.4524, D = 17.8025$.

Uysal [15] developed Sheta's models and proposed two new models as the following functions:

– **Uysal's Model 1**:

$$E = A * (KLOC)^B + C * ME^D + E \tag{8}$$

where $A = 3.3275, B = 0.8202$,
$C = -0.0874, D = 1.6840, E = 18.0550$.

– **Uysal's Model 2**:

$$\begin{aligned} E = A * (KLOC)^B + C * ME^D \\ +E * ln(ME) + F * ln(KLOC) + G \end{aligned} \tag{9}$$

where $A = 3.8930, B = 0.7923$,
$C = -0.2984, D = 1.3863, E = 2.8935$,
$F = -1.2346, G = 15.5338$.

This study uses the DABC algorithm with several modifications to optimize the parameters of four aforementioned models in order to enhance the accuracy of estimates.

2.2 Measuring estimation quality

The approaches which are widely used to evaluate the quality of software effort estimation models encompass:

– The Mean Magnitude of Relative Error (MMRE) [22]

– The Median Magnitude of Relative Error (MdMRE) [23]

– The Prediction at level *N* (PRED(*N*)) [24]

The Mean Magnitude of Relative Error is probably the most widely employed evaluation criterion for appraising the performance of software prediction models [26]. The MMRE is defined as Eq. 10.

$$MMRE = \frac{1}{T}\sum_{i=1}^{T} MRE_i \tag{10}$$

where T is the number of observations, *i* expresses each observation for which effort is predicted and MRE is Magnitude of Relative Error, which is computed as:

$$MRE_i = \frac{|ActualEffort_i - EstimatedEffort_i|}{ActualEffort_i} \tag{11}$$

Conte *et al.* [24] indicated that $MMRE \leq 0.25$ is acceptable for effort estimation models. Given two data sets A and B, suppose that data set A includes small projects whereas B contains large projects. Given everything else is equal and MMRE(B) is smaller than MMRE(A). As a result, a prediction model assessed on data set B will be considered as better than a competing model evaluated on data set A.

Unlike the mean value, the median always shows the middle value *m*, given a distribution of values, and assures that there is the same number of values above *m* as below *m*. Therefore, the median of MRE values for the number of observations called the MdMRE is an alternative to evaluate the performance of software prediction models. Similar to MMRE, the value of MdMRE less than or equal to 0.25 is acceptable for effort estimation models.

Another method which is commonly used is the Prediction at level *N* known as PRED(*N*). It is the percentage of projects for which the predicted values fall within $N\%$ of their actual values. For instance, if PRED(25) = 85, this indicates that 85% of the projects fall within 25% error ranges. Conte *et al* [24] claimed that *N* should be set at 25% and a good estimation system should offer this accuracy level to 75% of the effort. Eq. 12 illustrates the way to compute the value of PRED(*N*):

$$PRED\,(N) = \frac{100}{T} * \sum_{i=1}^{T} \begin{cases} 1, & \text{if } MRE_i \leq \frac{N}{100} \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

Although MMRE and MRE were frequently used for assessing the accuracy of effort estimation, Shepperd and MacDonell [25] criticized that the use of these criteria is biased. For instance, we have two projects where the first project is an over-estimate and the second project is an under-estimate. The actual and estimated values of the effort of project 1 are 20 and 100 respectively. Project 2 has the actual effort value being 100, and the estimated value is 20. Both estimates have identical absolute residual with 80, but the MMRE values differ by an order of magnitude. Consequently, MMRE will be biased towards prediction systems that under-estimate [25]. Therefore, Shepperd and MacDonell proposed a novel measure called mean absolute

residual (MAR), and it is shown in Eq. 13.

$$MAR = \frac{\sum_{i=1}^{T} |ActualEffort_i - EstimatedEffort_i|}{T} \quad (13)$$

This paper uses all four approaches above to assess the accuracy of the various software effort estimation models presented. Next section shows the DABC algorithm with some modifications to optimize the parameters of prediction models.

3 Directed artificial bee colony algorithm

The original ABC algorithm was proposed by Karaboga [16] based on simulating intelligent behavior of real honey bee colonies. One half of the artificial bees population contains the employed bees, while the other one includes the onlookers and scouts. In this algorithm, the total number of food sources (solutions) is equal to number of employed bees. The employed bees search the food around the food source and then give their information about the quality of the food sources to the onlookers. On the basis of information obtained, the onlooker bees make a decision, which food source to visit, and further search the foods around the chosen food sources. When the food source is exhausted, the corresponding employed and onlooker bees become scouts. These bees will abandon the food source and search for a new food source randomly. The general structure of ABC algorithm is shown in Algorithm 1.

Algorithm 1 General framework of Artificial Bee Colony Algorithm

```
Initialization Phase
while Cycle < Maximum Cycle Number (MCN) do
    Employed Phase
    Onlooker Phase
    Scout Phase
    Memorize the best solution achieved so far
end while
```

There are four control parameters in the original ABC. They are the maximum cycle number (MCN), the size of the population (SP) (the sum of numbers of employed and onlooker bees), the number of trials for abandoning food source *limit* and the number of scout bees (usually chosen as 1) [16].

In the initialization phase, the population of solutions is randomly produced in the range of parameters by using Eq. 14:

$$x_{ij} = Lb_j + r * (Ub_j - Lb_j), \quad i = 1, 2, ..., SP/2,\ j = 1, 2, ..., D \quad (14)$$

where *D* is the number of decision variables of the problem (also the number of variables need to be optimized in software effort estimation models), x_{ij} is the j^{th} dimension of the i^{th} food source which will be assigned to the i^{th} employed bee. Lb_j and Ub_j are the lower and upper bounds of the j^{th} dimension respectively, *r* is a random number in the range of [0, 1].

After the food sources are generated, their qualities are measured by using Eq. 15.

$$fit_i = \begin{cases} \frac{1}{1+f_i}, & \text{if } f_i > 0 \\ 1 + |f_i|, & \text{otherwise} \end{cases} \quad (15)$$

where fit_i is the fitness of the i^{th} food source, and f_i is the corresponding cost function value for the optimization problem. This study uses f_i as the value of MMRE on N projects of the training dataset using parameter values of the i^{th} food source, $f_i = MMRE_i$.

In the employed bee phase of the original ABC algorithm, every solution x_i (i = 1,...,SP/2) is updated according to the following equation:

$$v_{ij} = x_{ij} + \varphi * (x_{ij} - x_{kj}) \quad (16)$$

where v_i is the candidate food source position generated for food source position x_i, x_{ij} denotes the j^{th} parameter of x_i, *j* and *k* are random indexes ($j, k \in \{1, ..., SP/2\}$), φ is a uniform random number in the range of [-1, 1], x_k presents the other solution chosen randomly from the population.

It can be seen that only one dimension of the food source position is updated by the employed bees. This leads to a slow convergence rate. In order to overcome this issue, Akay and Karaboga [27] introduced a control parameter called modification rate (MR). In this improved algorithm, whether a dimension will be updated is decided by using the predefined MR value which is a number in the range of [0,1]. Eq. 16 is modified as follows:

$$v_{ij} = \begin{cases} x_{ij} + \varphi * (x_{ij} - x_{kj}), & \text{if } r_{ij} < MR \\ x_{ij}, & \text{otherwise} \end{cases} \quad (17)$$

where r_{ij} is a random number generated in the range of [0, 1] for the j^{th} parameter of x_i. If r_{ij} is less than MR, the dimension j is changed and at least one dimension is updated by using Eq. 16, otherwise the dimension j is remained [27].

Kiran *et al.* [19] claimed that the search process around the current food source in the basic ABC is fully random in terms of direction because φ is a random number in [-1, 1]. Therefore, they proposed adding direction information for each dimension of food source position and Eq. 16 is changed as follows:

$$u_{ij} = \begin{cases} x_{ij} + \varphi * (x_{ij} - x_{kj}), & \text{if } d_{ij} = 0 \\ x_{ij} + r * |x_{ij} - x_{kj}|, & \text{if } d_{ij} = 1 \\ x_{ij} - r * |x_{ij} - x_{kj}|, & \text{if } d_{ij} = -1 \end{cases} \quad (18)$$

where u_i is the candidate food source position generated for food source position x_i, d_{ij} is the direction information for j^{th} dimension of the i^{th} food source position and while

φ is a random number in the range of [-1, 1], r is a number generated randomly in the range of [0, 1].

This paper uses the following function to update position of the current food source:

$$v_{ij} = \begin{cases} u_{ij}, & \text{if } r_{ij} < MR \\ x_{ij}, & \text{otherwise} \end{cases} \quad (19)$$

where r_{ij} is a random number generated in the range of [0, 1], u_{ij} is presented in Eq. 18.

Algorithm 2 The pseudo code of the DABC algorithm

Input:
- the maximum cycle number: MCN
- the size of the population: SP
- the number of trials for abandoning food source: $limit$
- the modification rate: MR
- the dimensionality: D

Output: The best individual in the population: $\vec{x}_{best} = \{x_1, x_2, ..., x_D\}$.

Initialize the population solutions x_{ij}, i = 1,...,SP/2, j = 1,...,D.
Compute fitness value for each x_i by using Eq. 15
$cycle$ = 1
while $cycle \leq MCN$ **do**
  for $i = 1$ to $SP/2$ **do**
    - Generate a new solution v_i for the employed bee x_i by using Eq. 19
    - Apply the boundary constraint-handling mechanism for the created solution v_i by using Eq. 20
    - Compute fitness value for each x_i by using Eq. 15
    - Apply the greedy selection process
  end for
  for $i = 1$ to $SP/2$ **do**
    - Compute the probability value p_i for the solution x_i by Eq. 21
  end for
  Formulate the set of potential solutions S by using the roulette-wheel selection mechanism to select $SP/2$ solutions in the population based on the probability value p_i
  for each solution x_i in *S* **do**
    - Generate a new solution v_i for the employed bee x_i by using Eq. 19
    - Apply the boundary constraint-handling mechanism for the created solution v_i by using Eq. 20
    - Compute fitness value for each x_i by using Eq. 15
    - Apply the greedy selection process
  end for
  for $i = 1$ to $SP/2$ **do**
    if value *litmit* of solution x_i is reached **then**
      Produce a random solution and replace x_i with this solution
      break;
    end if
  end for
  Memorize the best solution achieved so far
  $cycle = cycle + 1$
end while

At the beginning of the algorithm, the direction information for all dimensions is equal to 0. If the new solution obtained by Eq. 18 has fitness value better than old one, the direction information will be updated. If prior value of the dimension is less than current value, the direction information of this dimension is set to -1; otherwise it is set to 1. If the fitness of candidate food source is worse than old one, the direction information of the dimension is assigned to 0. This way will help to improve the local search capability and enhance the convergence rate of the algorithm [19].

After generating the candidate food source position, if this position exceeds the boundaries of the variables then boundary constraint-handling mechanism is used and a diverse set of parameter values is produced, which helps to maintain diversity in the population. This paper applies the mechanism proposed in the Kukkonen and Lampinen work [28]. This mechanism is presented as follows:

$$v_{ij} = \begin{cases} 2 * Lb_j - v_{ij}, & \text{if } v_{ij} < Lb_j \\ 2 * Ub_j - v_{ij}, & \text{if } v_{ij} > Ub_j \\ v_{ij}, & \text{otherwise} \end{cases} \quad (20)$$

where v_{ij} is the j^{th} variable of the candidate solution v_i, Lb_j and Ub_j are the lower and upper bounds of the variable v_{ij}. By using Eq. 20, if there are a lot of solutions focused on the extreme values of the search space, the boundary constraint-handling mechanism will help algorithm to avoid getting stuck in the local minimum. After boundary constraint-handling mechanism is applied to the new solution, if the fitness of candidate food source is better than the old one, the new food source position will be memorized and trial counter of the food source is reset; otherwise the trial counter of the food source is increased by 1. This task is called the *greedy selection process*.

In the onlooker phase, each onlooker bee chooses an employed bee based on the probability value associated with food source of the employed bee and improves the quality of the food source chosen by using roulette-wheel selection mechanism [29] with the probability value given as follows:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SP/2} fit_j} \quad (21)$$

where p_i is the being selected probability of the i^{th} employed bee by an onlooker bee. Thereafter, the onlooker bee searches around the food source position chosen and the update process for the current food source position in the onlooker bee phase is the same as in the aforementioned employed bee phase.

In the scout phase, solutions that do not change over a certain number of trials are again initialized by using Eq. 14. In our study, each cycle has a maximum of one scout bee. The details of DABC algorithm are shown in Algorithm 2.

4 Experiments and results

4.1 Experimental dataset

Experiments in this paper have been conducted on a dataset represented by Bailey and Basili [30]. The dataset includes two variables, which are the Kilo Line of code (KLOC) and the Methodology (ME). The measured effort is described in person-months. The dataset is shown in Table 1.

Project No.	KLOC	ME	Measured Effort
1	90.2	30.0	115.8
2	46.2	20.0	96.0
3	46.5	19.0	79.0
4	54.5	20.0	90.8
5	31.1	35.0	39.6
6	67.5	29.0	98.4
7	12.8	26.0	18.9
8	10.5	34.0	10.3
9	21.5	31.0	28.5
10	3.1	26.0	7.0
11	4.2	19.0	9.0
12	7.8	31.0	7.3
13	2.1	28.0	5.0
14	5.0	29.0	8.4
15	78.6	35.0	98.7
16	9.7	27.0	15.6
17	12.5	27.0	23.9
18	100.8	34.0	138.3

Table 1: NASA software project data

In [14] and [15], authors utilized the data of first thirteen projects to optimize parameters of the estimation model, and five remaining projects were used for testing the performance after optimizing the parameters. To ensure comparison with these studies, we also took first thirteen projects as a learning set, and remaining ones are the testing set.

4.2 Experimental setup

The parameters of software effort estimation models are real numbers. With regard to Sheta's Model 1 and Sheta's Model 2, this paper used the range of parameters as presented in [14], and shown in Table 2. Table 3 presents parameter settings for Uysal's Model 1, and Table 4 shows configuration parameters for Uysal's Model 2.

Parameter	Minimum Value	Maximum Value
a	0	10
b	0.3	2
c	-0.5	0.5
d	0	20

Table 2: Configuration parameters for Sheta's Model 1 and Sheta's Model 2

In [14], Sheta used the maximum generation for genetic algorithms to optimize parameters of Sheta's Model 1 and Sheta's Model 2 is 100. This paper, therefore, also used the value of 100 for the maximum cycle number of the DABC algorithm when optimizing parameters of models. The remainder settings of the DABC algorithm to optimize parameters for four software effort estimation models were as follows:

Parameter	Minimum Value	Maximum Value
a	0	10
b	0.3	2
c	-0.5	0.5
d	0	5
e	0	20

Table 3: Parameter settings for Uysal's Model 1

Parameter	Minimum Value	Maximum Value
a	0	10
b	0.3	2
c	-0.5	0.5
d	0	5
e	0	5
f	-5	5
g	0	20

Table 4: Parameter settings for Uysal's Model 2

- The size of the population: 10
- The number of trials for abandoning food source: 2
- The modification rate: 0.7

4.3 Results and empirical evaluations

The model parameters which presented by Eq. 6 were optimized using the DABC algorithm as bellow. This model after optimizing parameters using the DABC was called as **Model 1**.

$$A = 5.4507, \qquad B = 0.7082, \qquad C = -0.3184$$

The model represented by Eq. 7 was named as **Model 2** after its parameters was optimized by using the DABC algorithm. The optimal values of parameters of Model 2 were as bellow:

$$A = 1.7319, \qquad B = 0.966, \qquad C = -0.5, \qquad D = 11.5731$$

This paper calls the software effort estimation model presented by Eq. 8 after optimizing parameters using the DABC as **Model 3**. The optimal values of parameters in Model 3 were as follows:

$$A = 2.3003, \qquad B = 0.8982, \qquad C = -0.0164, \qquad D = 2.0623, \qquad E = 14.1862$$

Model shown by Eq. 9 after optimizing parameters using the DABC was called as **Model 4**. Model 4 got the optimal

values as bellow:

$$A = 4.4442, \quad B = 0.7826, \quad C = -0.373,$$
$$D = 1.2756, \quad E = 2.4425,$$
$$F = -4.9198, \quad G = 17.0804$$

Table 5 shows the actual effort, and predicted effort values using Model 1, Sheta's Model 1, Model 2, Sheta's Model 2, and simple Regression model. Table 6 presents measured data, predicted values of Model 3, Uysal's Model 1, Model 4, Uysal's Model 2, and simple Regression model.

First of all, we assess the accuracy of models using criteria MMRE, MdMRE, and PRED(25). Table 7 shows the obtained results of nine estimation models on 18 NASA projects. The results indicated that Model 1 could not improve the MMRE of Sheta's Model 1, while Model 2 significantly enhanced the MMRE of Sheta's Model 2 by 46.5%. After applying the DABC algorithm, the MMRE of Uysal's Model 1 decreased by more than 5.8% and an additional 5.6% improvement over Uysal's Model 2 was obtained. Model 2, Model 3, and Model 4 produced the better results compared with simple regression with regard to MMRE. This means that the efficiency of software cost estimation models except for Model 1 has been significantly improved after using the DABC algorithm to optimize parameters. With regard to MdMRE, Model 4 gave the lowest result, the second one belonged to Model 3, while Sheta's Model 2 produced the highest value. In general, the values of MdMRE for models with parameters optimized using the DABC are lower than those using original models.

With respect to PRED(25), Model 4, Model 3, and Regression produced the highest value, while the lowest value belonged to Sheta's Model 2. An improvement of 5.5% was achieved in PRED(25) on Uysal's models after applying the DABC algorithm for optimizing parameters. Three out of four improved models gave the values of PRED(25) higher than 75%. This proved that the improved models have been good effort estimation systems. Meanwhile, the values of PRED(25) of both models proposed by Sheta were less than 75%. These results indicated that the models of Sheta have not been suitable for making effort estimates.

In general, the improved models using the DABC algorithm to optimize parameters presented the good prediction accuracy, and were better than the original models in terms of all evaluation criteria. It is also seen that Model 4 gave the most accurate predictions.

As mentioned above, the MMRE criterion is biased, so MAR is used for evaluating the performance of predicted models. The values of MAR of each model on 18 projects are reported in Table 8.

Based on the results of the experiments, it is seen that all improved models outperformed their original ones in terms of MAR. It is also found that the Model 4 overcame all models, while the Model 3 was ranked second. Although Sheta's models were enhanced by using DABC to optimize parameters, they could not show better performance when compared with Uysal's models and the simple regression model. These results continue to prove that Sheta's models are not suitable for software effort estimation.

To enrich the study of the MAR results of the Model 4, we carried out statistical tests to see whether the Model 4 is statistically different from other models in datasets. We also applied statistical tests to see whether the Model 3, the second winning model, is statistically different from other models. We used a normality test on the results obtained and identified that data were not normally distributed. For this reason, we employed the Wilcoxon test, which is a nonparametric test; this type of tests should be used when the distribution is not normal. Table 9 gives the results of the Wilcoxon test based on the 95 % confidence interval (CI). Bold text indicates that the model is statistically different at 95% CI. If the p-value is less than or equal to 0.05, then we conclude that the current model is statistically different from the other models at 95% CI. Otherwise, models are not statistically different at 95% CI. Based on Table 9, we can see that the Model 4 is statistically different from the model 3, and Sheta's Model 2, but it fails to be statistically different from remaining models. Model 3 is statistically different from the Model 2 and Sheta's Model 2, but it is not also statistically different from the remaining models.

5 Conclusion and future work

In this paper, we studied the problem of optimizing parameters for software effort estimation models. The directed artificial bee colony algorithm with several modifications was used to tackle this optimization problem. The models after optimizing parameters produced the results whose accuracy was considerably improved in comparison with the original models in terms of all evaluation criteria such as MMRE, MdMRE, PRED(25), and MAR. In general, the accuracy of software effort is enhanced by more than 5.5% by applying the improved models.

The future work focuses on employing the DABC algorithm to optimize parameters for other effort estimation models. We also use other algorithms of Swarm Intelligence for four models presented in this paper and carry out the comprehensive assessment in terms of the performance of algorithms for the cost estimation problem.

References

[1] Ochodek, M., Nawrocki, J., Kwarciak, K. (2011). Simplifying effort estimation based on Use Case Points, *Information and Software Technology*, 53(3): 200-213.

[2] Mendes, E., Mosley, N., Watson, I. (2002). A comparison of case-based reasoning approaches. *Proceedings of the 11th international conference on World Wide Web*, Hawaii, USA, pp. 272-280.

Project	Measured Effort	Model 1	Sheta's Model 1	Model 2	Sheta's Model 2	Regression
1	115.8	122.617	124.8585	130.6186	134.0202	126.5132
2	96	75.9229	74.8467	71.8103	84.1616	78.6798
3	79	76.6194	75.4852	72.7508	85.0112	80.5612
4	90.8	86.1377	85.4349	83.9645	94.9828	90.4495
5	39.6	51.0323	50.5815	41.9945	56.658	35.4272
6	98.4	98.4043	99.0504	98.378	107.2609	95.7798
7	18.9	24.8788	24.148	18.9008	32.6461	22.5813
8	10.3	17.992	18.0105	11.3608	25.0755	7.6717
9	28.5	38.002	37.2724	29.6205	44.3086	27.6381
10	7	3.8676	4.5849	3.7394	14.4563	8.8264
11	9	9.0108	8.9384	9.0007	19.9759	20.5784
12	7.3	13.4767	13.5926	8.6707	21.5763	8.2111
13	5	0.303	1.51	1.1195	11.2703	4.4963
14	8.4	7.8061	8.2544	5.2715	17.0887	7.1526
15	98.7	108.7481	110.5249	111.4279	118.0378	102.7839
16	15.6	18.648	18.2559	13.6236	26.8312	16.7294
17	23.9	24.0082	23.369	17.9404	31.6864	20.6999
18	138.3	132.1635	135.4825	143.8064	144.4587	135.7203

Table 5: Measured data, predicted values according to Model 1, Sheta's Model 1, Model 2, Sheta's Model 2, and Regression model

Project	Measured Effort	Model 3	Uysal's Model 1	Model 4	Uysal's Model 2	Regression
1	115.8	127.1478	124.794	125.3071	124.3563	126.5132
2	96	78.2194	81.6608	77.743	81.6143	78.6798
3	79	79.4325	83.1941	79.1179	83.1781	80.5612
4	90.8	89.7282	92.8603	89.2478	92.757	90.4495
5	39.6	39.5325	39.0238	39.5424	39.6279	35.4272
6	98.4	98.3011	98.0132	97.2828	97.8566	95.7798
7	18.9	23.3181	23.8838	21.3727	23.8446	22.5813
8	10.3	9.5814	7.7948	8.5967	8.2993	7.6717
9	28.5	30.8556	30.8864	29.6235	31.0829	27.6381
10	7	6.96	5.3694	6.441	5.7918	8.8264
11	9	15.4219	16.4089	14.9203	16.7359	20.5784
12	7.3	9.223	7.6178	7.75	7.8978	8.2111
13	5	2.8414	0.2631	3.3478	0.9986	4.4963
14	8.4	6.9379	5.1496	5.6857	5.4465	7.1526
15	98.7	105.0602	102.5719	104.7675	102.7935	102.7839
16	15.6	17.2108	17.0202	15.2793	17.0407	16.7294
17	23.9	21.7401	21.9803	19.8065	21.9698	20.6999
18	138.3	135.5447	131.2398	133.8053	130.9554	135.7203

Table 6: Measured data, predicted values according to Model 3, Model 4, Uysal's Model 1, Uysal's Model 2, and Regression model

Model	MMRE(%)	MdMRE(%)	PRED(25)
Sheta's Model 1	23.79	14.5	61.11
Sheta's Model 2	63.64	49.27	38.89
Uysal's Model 1	20.04	8.2	77.78
Uysal's Model 2	18.80	8.63	77.78
Regression	17.20	10.31	83.33
Model 1	26.03	14.86	61.11
Model 2	17.13	11.48	77.78
Model 3	14.20	8.65	83.33
Model 4	13.21	7.07	83.33

Table 7: Results for techniques based on criteria MMRE, MdMRE, and PRED(25)

Model	MAR
Sheta's Model 1	5.71
Sheta's Model 2	11.26
Uysal's Model 1	4.00
Uysal's Model 2	3.92
Regression	3.94
Model 1	5.69
Model 2	5.25
Model 3	3.51
Model 4	3.45

Table 8: Results for techniques based on MAR

	p-value at 95% CI
Model 4 vs. Model 3	**0.00072**
Model 4 vs. Model 2	0.61708
Model 4 vs. Model 1	0.18352
Model 4 vs. Uysal's Model 1	0.34722
Model 4 vs. Uysal's Model 2	0.25014
Model 4 vs. Sheta's Model 1	0.20054
Model 4 vs. Sheta's Model 2	**0.0002**
Model 4 vs. Regression	0.28462
Model 3 vs. Model 2	**0.0002**
Model 3 vs. Model 1	0.5892
Model 3 vs. Uysal's Model 1	0.37346
Model 3 vs. Uysal's Model 2	0.5287
Model 3 vs. Sheta's Model 1	0.5892
Model 3 vs. Sheta's Model 2	**0.0002**
Model 3 vs. Regression	0.32708

Table 9: Wilcoxon test for the Model 4 and the Model 3

[3] Jorgensen, M. (2004). A review of studies on expert estimation of software development effort, *Journal of Systems and Software*, 70(1–2): 37-60.

[4] Khalifelu, Z.A., Gharehchopogh, F.S. (2012). Comparison and evaluation of data mining techniques with algorithmic models in software cost estimation, *Procedia Technology*, 1: 65-71.

[5] Briand, L.C., Wieczorek, I. (2002). Resource Estimation in Software Engineering, *Encyclopedia of Software Engineering, John Wiley & Sons*, 2: 1160-1196.

[6] Chen, Z., Menzies, T., Port, D., Boehm, B (2005). Feature subset selection can improve software cost estimation accuracy. Proceedings of the 2005 workshop on Predictor models in software engineering, ACM, pp. 1-6.

[7] Attarzadeh, I., Ow, S.H. (2010). A Novel Algorithmic Cost Estimation Model Based on Soft Computing Technique, *Journal of Computer Science*, 6(2): 117-125.

[8] Putnam, L.H. (1978). A General Empirical Solution to the Macro Software Sizing and Estimating Problem, *IEEE Transactions on Software Engineering*, SE-4(4) 345-361.

[9] Galorath, D.D., Evans, M.W. (2006). *Software sizing, estimation, and risk management*, Auerbach Publications, Boston, MA.

[10] Mittal, A., Parkash, K., Mittal, H. (2010). Software cost estimation using fuzzy logic, *ACM SIGSOFT Software Engineering Notes*, 35(1): 1-7.

[11] Gharehchopogh, F.S. (2011). Neural networks application in software cost estimation: A case study. International Symposium on Innovations in Intelligent Systems and Applications (INISTA), pp. 69-73.

[12] Selby, R.W., Porter, A.A. (1988). Learning from examples: generation and evaluation of decision trees for software resource analysis, *IEEE Transactions on Software Engineering*, 14(12): 1743-1757.

[13] Boehm, B., Abts, C., Chulani, S. (2000). Software development cost estimation approaches — A survey, *Annals of Software Engineering*, 10(1-4): 177-205.

[14] Sheta, A.F. (2006). Estimation of the COCOMO Model Parameters Using Genetic Algorithms for NASA Software Projects, *Journal of Computer Science*, 2(2): 118-123.

[15] Uysal, M. (2010). *Estimation of the Effort Component of the Software Projects Using Heuristic Algorithms*, In RAMOV, B. (ed.). New Trends in Technologies, Rijeka, Croatia, InTech.

[16] Karaboga, D., Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, 39(3): 459-471.

[17] Dorigo, M., Maniezzo, V., Colorni, A., Maniezzo, V. (1991). Positive feedback as a search strategy. Technical Report, Politecnico di Milano, Italy.

[18] Kennedy, J., Eberhart, R. (1995). Particle swarm optimization. Proceedings of IEEE International Conference on Neural Networks, pp. 1942-1948.

[19] Kiran, M.S., Findik, O. (2015) A directed artificial bee colony algorithm, *Applied Soft Computing*, 26: 454-462.

[20] Boehm, B., Clark, B., Horowitz, E., Westland, C., Madachy, R., Selby, R. (1995). Cost Models for Future Software Life Cycle Processes: COCOMO 2.0, *Annals of Software Engineering*, 1(1): 57-94.

[21] Boehm, B.W. (1984). Software Engineering Economics, *IEEE Transactions on Software Engineering*, SE-10(1): 4-21.

[22] Olatunji, S., Selamat, A. (2015). Type-2 Fuzzy Logic Based Prediction Model of Object Oriented Software Maintainability, *Intelligent Software Methodologies, Tools and Techniques*, 513: 329-342.

[23] Valdes, F., Abran, A. (2010). Comparing the Estimation Performance of the EPCU Model with the Expert Judgment Estimation Approach Using Data from Industry, *Software Engineering Research, Management and Applications*, 296: 227-240.

[24] Conte, S.D., Dunsmore, H.E., Shen, V.Y. (1986). *Software engineering metrics and models*, Benjamin-Cummings Publishing Co., Inc.

[25] Shepperd, M., MacDonell, S. (2012). Evaluating prediction systems in software project estimation, *Information Software Technology*, 54(8): 820–827

[26] Briand, L., Wieczorek, I. (2002). *Resource Modeling in Software Engineering*. Encyclopedia of Software Engineering, Wiley.

[27] Akay, B., Karaboga, D. (2012). A modified Artificial Bee Colony algorithm for real-parameter optimization, *Information Sciences*, 192: 120-142.

[28] Kukkonen, S., Lampinen, J. (2006). Constrained Real-Parameter Optimization with Generalized Differential Evolution, *IEEE Congress on Evolutionary Computation*, pp. 207-214.

[29] Blickle, T., Thiele, L. (1996). A comparison of selection schemes used in evolutionary algorithms, *Evolutionary Computation*, 4(4): 361-394.

[30] Bailey, J.W., Basili, V.R. (1981). A meta-model for software development resource expenditures. *Proceedings of the 5th international conference on Software engineering*, pp. 107-116.

A Secure and Fast Chaotic Encryption Algorithm Using the True Accuracy of the Computer

Jean De Dieu Nkapkop and Joseph Yves Effa
University of Ngaoundéré, Department of Physics, P.O. Box 454, Ngaoundéré, Cameroon
E-mail: golby01@yahoo.fr, effa_jo@yahoo.fr

Monica Borda
Technical University of Cluj-Napoca, Department of Communications,
26-28 Baritiu Street, 400027, Cluj-Napoca, Romania
E-mail: Monica.Borda@com.utcluj.ro

Laurent Bitjoka
University of Ngaoundéré, Department of Electrical Engineering, Energetics and Automatics,
P.O. Box 454, Ngaoundéré, Cameroon
E-mail: lbitjoka@univ-ndere.cm

Alidou Mohamadou
University of Maroua, Department of Physics, P.O. Box 814, Maroua, Cameroon
E-mail: mohdoufr@yahoo.fr

Keywords: secure encryption, fast cryptosystem, chaotic sequences, permutation-diffusion scheme

Received: January 8, 2016

A secure and fast cryptosystem for image encryption based on chaotic generators is proposed. The principle of the method is to use the permutation-diffusion scheme to create computationally secure encryption primitives using the true accuracy of the computer. In the permutation step, integer sequences obtained by the sorting of the solutions of chaotic Logistic map by descending order is used as the permutation key to shuffle the whole image. This stage substantially reduces the correlation between neighbouring pixels. After, in order to increase the entropy of encrypted image, the iteration of the chaotic Skew Tent map is applied, with an exclusive-or scheme, to change the value of the entire pixel. Moreover, to further enhance the security of the cryptosystem, the keystream used in diffusion process is updated for each pixel and the computed encrypted pixel values depends on both the previously encrypted pixels and the random keystream. We proved that the cipher sequence of the algorithm is random and truly random by applying the NIST tests batteries and $\chi 2$-test respectively. Hence, the proposed algorithm can resist the statistical attacks. The extensive cryptanalysis has also been performed and results of our analysis indicate that the scheme is satisfactory in term of the superior security and high speed as compared to the existing algorithms, which makes it a very good candidate for real-time of multimedia data encryption applications.

Povzetek: Prispevek opiše nov način kriptiranja slik s pomočjo kaotičnih generatorjev.

1 Introduction

In today's world, the extension of multimedia technology in which image covers the highest percentage, has promoted digital images to play a more significant role than the traditional texts. The Internet banking, e-business, e-commerce, etc., are the major fields where security is most important. So it is necessary to encrypt image data before transmission over the network to preserve its security and prevent unauthorized access. For this end, most of the conventional encryption algorithms such as Advanced Encryption Standard (AES) [1] are designed with good properties [2, 3]. However, due to bulk volume of data, high correlation among adjacent pixels, high redundancy and real time requirement [4], these ciphers may not be the most desired candidates for image encryption, particularly for fast and real-time communication applications [5]. To meet this challenge, the chaos-based encryption has suggested a new and efficient way to deal with the intractable problem of fast and highly secure image encryption [6]. The properties of chaos such as high sensitive dependence on initial conditions and control parameter, quasi-randomness, ergodicity, unpredictability, mixing, etc. [7], which are analogous to the confusion and diffusion properties of Shannon [8], have granted chaotic dynamics as a promising alternative for the traditional cryptographic algorithms, and also for generating keystream.

Depending on the type of key used in the encryption algorithms, chaos-based cryptosystems are either symmetric or asymmetric. Symmetric encryption, in which the decryption key is identical to the encryption

key, is the oldest method in cryptology and is still used today. By contrast, asymmetric cryptosystems use different keys for decryption and encryption. We consider here typical (symmetric encryption) chaos-based image encryption techniques which rely on two processes: pixel permutation and pixel substitution [9, 10]. The first one, also call pixel confusion is needed to scramble the pixels. But, due to the strong correlation between adjacent pixels of the images, this stage does not guarantee a good level of security [11]. The diffusion stage is thus used to modify the pixel values in order to increase the entropy of the entire image.

Several image encryption algorithms based on this structure are already available in the literature [10, 12, 13, 14]. Each of them has its own strength and limitations more or less in terms of security level and computational speed. Accordingly, some of them have been cryptanalyzed successfully [15, 16, 17, 18, 19]. The common characteristic of these algorithms are: their chaotic generators need to be discretized to the finite sets of integers and that is time consuming and destroyed also their chaotic behaviors. Also, the keystream in the diffusion stage of these algorithms depends on the key only and that is less secured because an attacker can obtain that keystream by known/chosen plaintext attack [16]. So, to enhance the security, in [20], the keystream in the diffusion step depend on both the key and original image. Another method to obtain a high immunity to resist the differential cryptanalysis is to design strong substitution Boxes (S-Boxes) based on chaotic map or strong diffusion properties based on the combination of chaotic function and other techniques [21, 22, 23].

To improve the computational performance and to resist statistical, differential, brute-force attacks, this paper continues the same pursuit with further improvement, in which a one round chaos-based image encryption scheme based on the fast generation of large permutation key with a good level of randomness and a very high sensitivity on the keys is proposed. We use the integer sequences obtained by the descending sorting of the Logistic map as a secret key in the permutation stage. This technique avoids the excess digitization of chaotic values. As consequence, the sensitivity to small changes of the initial condition or control parameters is increased, as the true accuracy of the computer is exploited by using integer sequences. In diffusion process, at first, a random code is generated to get integer numbers from real numbers generated by Skew Tent map. Then, with that numbers, the exclusive-or is performed on the permuted image to computed the cipher image. The proposed approach can be easily implemented and is computationally simple.

The remaining of the paper is organized as follows. The chaotic maps are described in Section 2. In Section 3, the proposed encryption scheme is discussed in detail. Simulation results and security analysis are presented in Section 4 to show the efficacy and validity of the algorithm. Finally, conclusions are drawn in the last Section.

2 Chaotic maps

Chaotic maps are nonlinear maps that exhibit chaotic behavior. The chaotic maps generate pseudo-random sequences, which are used during encryption process [24]. Many fundamental concepts in chaos theory, such as mixing and sensitivity to initial conditions and parameters, actually coincide with those in cryptography [25]. The only difference in this concern is that encryption operations are defined on finite sets of integers while chaos is defined on real numbers. The main advantage using chaos lies in the observation that a chaotic signal looks like noise for the unauthorized users. Moreover, generating chaotic values is often of low cost with simple iterations, which makes it suitable for the construction of stream ciphers. Therefore, cryptosystem can provide a secure and fast means for data encryption, which is crucial for data transmission in many applications. The proposed scheme uses Logistic and Skew Tent maps and they are both discussed hereafter.

2.1 Logistic map

The Logistic map is a very simple non-linear dynamical and polynomial equation of degree two with x output and input variable, one initial condition x_0 and one control parameter λ and can be described as follows:

$$x_{n+1} = \lambda x_n (1 - x_n) \tag{1}$$

Where $x_n \in (0,\ 1)$ is the state of the system, for $n = 0,1,2,...$, and $\lambda \in (0,\ 4)$ is the control parameter. For different values of parameter λ, the Logistic sequence shows different characteristics [26]. For $3,58 \le \lambda \le 4$, the Logistic map Eqn. (1) has a positive Lyapunov exponent and thus is always chaotic. So all the (x_0, λ) where $x_0 \in (0,\ 1)$ and $3,58 \le \lambda \le 4$ can be used as secret keys.

2.2 Skew tent map

The Skew Tent chaotic map [27] can be described as follows:

$$y_{n+1} = \begin{cases} y_n/\alpha, & \text{if } y_n \in [0,\alpha] \\ (1\text{-}y_n)/(1\text{-}\alpha), & \text{if } y_n \in [\alpha,1] \end{cases} \tag{2}$$

Where α is controllable parameter for chaotic maps, y_i and y_{i+1} are the i-th and the $i+1$-th state of chaotic maps. For $\alpha \prec 1$ the system converges to 0 for all initial conditions. If $\alpha = 1$, then all initial conditions less than or equal to 0.5 are fixed points of the system, otherwise for initial conditions $y_0 \succ 0.5$ they converge to the fixed point $1 - y_0$. So all the y_0, $\alpha \in (0,\ 1)$, can be used as secret keys.

3 Proposed encryption scheme

The encryption algorithm consists of two stages: permutation and diffusion of pixels of the entire image as shown in Figure 1.

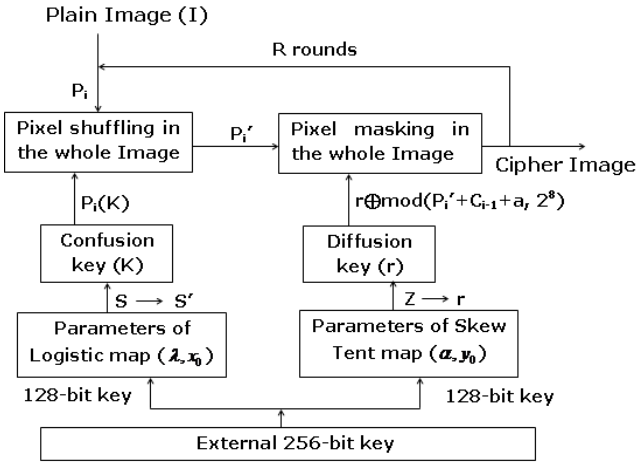


Figure 1: Synoptic of the proposed scheme.

In the proposed algorithm, we use one round (R=1) of confusion and diffusion for encryption.

3.1 Confusion

In this stage, the position of the pixels is scrambled over the entire image without change their values and the image becomes unrecognizable. The purpose of confusion is to reduce the high correlation between adjacent pixels in the plain image. To enhance the degree of randomness and the level of security, the Logistic map described in subsection 2.1 is used in order to generate pseudorandom key stream $S=\{x_1, x_2, ..., x_{M\times N}\}$ as the same size of the plain-image. Let I be a gray original image of size $M\times N$, containing M rows and N columns, and the gray values ranges from 0 to 255. Transform I to a one-dimensional vector $P=\{P_1, P_2, ..., P_{M\times N}\}$, where P_i is the i-th pixel value. Then sort S by descending order, and note $S'=\{x_j, ..., x_8, ..., x_1\}$ with $x_1 \prec ... \prec x_8 \prec ... \prec x_j$, the sorted chaotic values. The positions of sorted chaotic values in the original chaotic sequence are found and stored in $K=\{j, ..., 8, ..., 1\}$. Now, the next step is to scramble the total one-dimensional vector with K by using the following formula:

$$P' = P(K) \tag{3}$$

Where P' is the permuted image and K the permutation key. The reconstruction of P cannot be made unless the distribution of K is determined. The inverse transform for deciphering is given by:

$$P'(K) = P \tag{4}$$

This technique avoids the excess digitization of chaotic values. As consequence, the sensitivity to small changes of the initial condition or control parameters is increased, as the true accuracy of the computer is exploited and the computational time necessary for the generation of large permutation is reduced.

After obtaining the shuffled image, the correlation among the adjacent pixels is completely disturbed and the image is completely unrecognizable. Unfortunately, the histogram of the shuffled image is the same as that of the plain-image. Therefore, the shuffled image is weak against statistical attack and known plain-text attack. As a remedy, we design diffusion next to improve the security.

3.2 Diffusion

The total image is again encrypted with different chaotic numbers. Skew Tent map system shown in section 2.2 is applied here to produce that numbers: $Z=\{y_1, y_2, ..., y_{M\times N}\}$. The masking process is employed to modify the gray values of the image pixels, to confuse the relationship between the plain image and the encrypted image in order to increase the entropy of the plain image by making its histogram uniform. The diffusion function is also used to ensure the plain image sensitivity so that, a very little change in any one pixel of plain image should spread out to almost all pixels in the whole image. Diffusion is performed by using following equation:

$$C_i = r \oplus \mathrm{mod}\left(P_i' + C_{i-1} + a, 2^8\right) \tag{5}$$

Where C_i and C_{i-1} are the value of the currently and previously masking pixel respectively; C_0 can be set as a constant; P_i' is the permuted pixels; $\oplus$ is bitwise XOR operation; a is a positive integer and r is a random code obtained according to the following formula:

$$r = \mathrm{mod}\left(floor\left(y_n \times 2^{20}\right), 256\right) \tag{6}$$

Where, *mod* (x, y) returns the remainder after division and y_n is the state value corresponding to the n-th iteration of the skew tent map from initial state value y_0 and α.

A random code r is computed to get integer numbers from real numbers generated by Skew Tent map.

The key formula in decryption procedure is as follows:

$$P_i' = \mathrm{mod}\left(r \oplus C_i - C_{i-1} - a, 2^8\right) \tag{7}$$

To compute the first encrypted pixel, equation 8 is used.

$$C_1 = r \oplus \mathrm{mod}\left(P_1' + C_0 + a, 2^8\right) \tag{8}$$

Where r is evaluated by using Skew Tent map parameters below for i=1 to generate y.

$$\begin{cases} y_0 = \left(C_{i-1} + a\right) / \left(255 + a + b\right) \\ \alpha = \left(P_i' + a\right) / \left(M \times N + a + b\right) \end{cases} \tag{9}$$

With $a, b, C_0 \succ 0$.

For the security to be strengthened, the keystream r is updated for each pixel and the computed encrypted pixel values C_i depends on the previously encrypted pixels and the keystream, hence algorithm shows resistance to the differential attacks such as known plaintext attack, chosen-plaintext attack, known cipher-text attack and so on.

3.3 Encryption scheme

3.3.1 Encryption algorithm

The encryption algorithm is composed of thirteen steps.
Step 1: Reshape the plaintext image I into 1-D signal P and choose x_0 and λ in (0, 1) and (3.58, 4) respectively;
Step 2: Iterate the Logistic map given in equation 1 for T times to get rid of transient effect, where T is a constant;
Step 3: Continue to iterate the Logistic map for $M\times N$ times, and take out the state $S=\{x_{1+T}, x_{2+T}..., x_{M\times N+T}\}$;
Step 4: Sort S and get S' then, generate the permutation keys K as explained in subsection 3.1;
Step 5: Shuffle the pixels of the whole 1-D signal P with K using equation 3 and get P';
Step 6: Give C_0, choose a, b and evaluate y_0 and α in (0, 1) respectively as shown in equation 9;
Step 7: Iterate the Skew Tent map T times by using equation 2 and get the random code r;
Step 8: Compute the first cipher-pixel C_1 using equation 8 for i=1;
Step 9: Set i=i+1 and update y_0 and α in (0, 1) and get a new chaotic sequence y_T;
Step 10: Evaluate the new random code r by using equation 6;
Step 11: Compute cipher-pixel C_i according to the formula 5;
Step 12: Repeat step 9 to 11 until i reaches $M\times N$, the length of the whole 1-D signal;
Step 13: Reshape the 1-D signal into the 2-D image and get ciphered image.

The decryption involves reconstructing gray levels of the original image from the encrypted image. It is a simple inverse process of the proposed encryption algorithm.

3.3.2 Key schedule

A key of 128 bits or 256 bits is required for symmetric-key cryptosystems for more security [28]. We used an external 256-bit key ($E_1E_2 \cdot \cdot \cdot E_i \cdot \cdot \cdot E_{32}$, where E_i are ASCII symbols) to derive initial conditions and control parameters of the chaotic systems. The key is divided into two blocks of 16 ASCII symbols for the determination of the system control parameter and the initial condition respectively. For each block of 128 bits (corresponding to 16 ASCII symbols), we defined:

$$W=\sum_{i=0}^{15} 2^{\frac{i}{i+1}} P_i \tag{10}$$

where P_i are values (0-255) of ASCII symbols E_i and W is the value from which the control parameters and initial conditions are deduced, depending on the chaotic system. By considering the possible maximum value of ASCII symbols equal to 255 and the upper limit of the weight coefficient $2^{\frac{i}{i+1}}$ equal to 2, the value of W presents an upper limit Wr = 8160, which is used for its normalization.

The flowchart of the proposed encryption algorithm is then described in Figure 2.

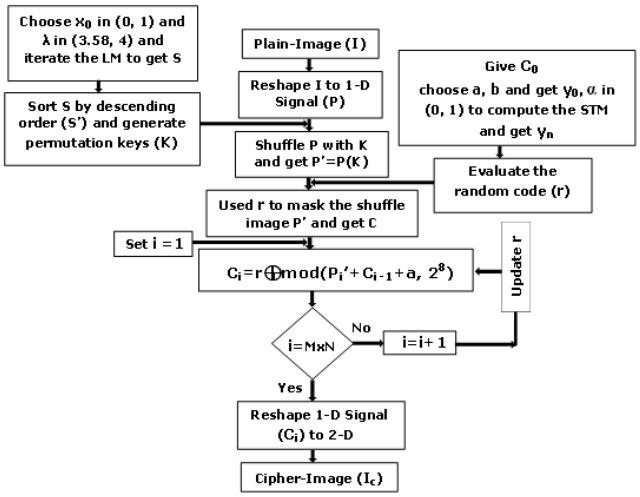


Figure 2: Flowchart of the encryption algorithm.

4 Experiments and security analysis

In this section, the proposed image cryptosystem is analyzed using different security measures. These measures consist of statistical analysis, sensibility analysis, differential attack analysis and speed analysis. Each of these measures which are widely used in the literature in the field of chaos-based cryptography [4, 10, 22, 23, 25, 29, 30, 31] is described in detail in the following subsections.

4.1 Statistical analysis

4.1.1 Randomness test

The National Institute of Standards and Technology (NIST) of the U.S. Government designed a set of fifteen tests to evaluate and quantify the randomness of binary sequences produced by either software or hardware based random or pseudo-random number generators for cryptographic applications [32]. The NIST has adopted two approaches: the examination of the proportion of sequences that pass a statistical test and the distribution of P-values to check for uniformity.

In our experiment, we used m = 2000 different keystreams, each sequence having a length of n = 1000,000 bits which are generated using our scheme. The acceptance region of the passing ratio is given by equation (11), where m represents the number of samples tested (m=2000) and P the probability corresponding to the significance level 0.01 (P = 0.99). In this case, we obtained the confidence interval [0.983, 0.996]. We have summarized the results obtained after applying the NIST test suite on the binary sequences produced by the proposed pseudo-random bit generator in the second and fourth column of Table 1. The proportion for each test computed on the Lena and Baboon encrypted images lies inside the confidence interval. So, the tested binary sequences generated by the proposed chaotic generator are random with respect to all tests of NIST suite with a confidence of 99%. In order to show that binary sequences tested are truly random, the χ2-test is used

[32]. According to that test, the P-values must be greater than 0.0001 to ensure that they could be considered uniformly distributed. The results from the third and fifth column of Table 1 lead us to the conclusion that P-values, for each statistical test, are well uniformly distributed.

These results show the quality of the produced sequences with the pseudo-random number generator. So the proposed map has perfect cryptographic properties.

$$\left[p-3\sqrt{(p(1-p))/m} \ , \ p+3\sqrt{(p(1-p))/m,} \right] \quad (11)$$

4.1.2 Visual test

In this subsection, we perform visual test using Lena and

Test name	Lena		Baboon		Result
	Passing ratio of the test	Uniformity P-value	Passing ratio of the test	Uniformity P-value	
Frequency	0.9907	0.053181	0.9902	0.216399	Good
Block frequency	0.9921	0.214936	0.9862	0.197340	Good
Cumulative sums	0.9886	0.183142	0.9819	0.179402	Good
Runs	0.9895	0.492088	0.9900	0.618276	Good
Longest run	0.9898	0.625312	0.9788	0.498233	Good
Rank	0.9876	0.429910	0.9876	0.429910	Good
FFT	0.9882	0.103644	0.9950	0.112419	Good
Non-overlapping template	0.9891	0.999925	0.9682	0.968566	Good
Overlapping template	0.9880	0.580201	0.9909	0.682005	Good
Universal	0.9869	0.273197	0.9898	0.293941	Good
Approximate entropy	0.9911	0.991018	0.9899	0.969684	Good
Random excursions	0.9934	0.544220	0.9897	0.510334	Good
Random excursions variant	0.9887	0.898591	0.9606	0.699372	Good
Serial	0.9901	0.682476	0.9953	0.708396	Good
Linear complexity	0.9918	0.733601	0.9868	0.599271	Good

Table 1: The statistic results of cipher images by 2010 revised version of NIST statistic test.

Black images of size 512 × 512 encrypted using parameters x_0=0.75, λ=3.393695629843 for the permutation and a=7, b=4 and C_0=23 for the diffusion. As shown in Figure 3(*b*) and (*b*'), the encrypted image is non recognizable in appearance, unintelligible, incomprehensible, random and noise-like image without any leakage of the original information. This demonstrates that the proposed algorithm can be used to protect various images for diverse protections. The decrypted images are exactly same as the original images (Figure 3(*c*) and (*c*')).

4.1.3 Histogram analysis

The histogram of the plain-image and cipher image is in Figure 3(*e*) and (*f*) respectively. We found that the histogram of the ciphered image has approximately a uniform distribution. For instance, the histogram in Figure 3(*f*') which corresponds to the ciphered Black image highlights the effectiveness of the algorithm, as all the 256 gray-levels present the same probability. To confirm this result, we measured the entropy for the ciphered image and we found that it has the value 7.9996 which is close to the ideal value 8.

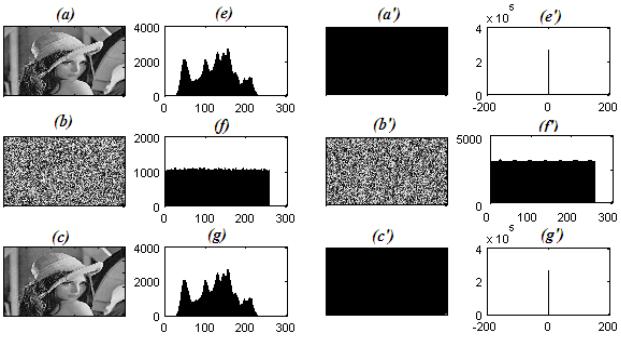


Figure 3: Histogram. (a), (a') original image; (b), (b') ciphered image of (a), (a'); (c), (c') decrypted image of (b), (b'); (e), (e'), (f), (f'), (g), (g') histogram of (a), (a'), (b), (b'), (c), (c') respectively.

4.1.4 Key space analysis

The key space is the total number of different keys that can be used in the encryption/decryption procedure. For an effective cryptosystem, the key space should be sufficiently large enough to resist brute-force attacks. In the proposed algorithm a 256-bit key corresponding to 32 ASCII symbols is considered and the key consists of the initial value x_0, a, b and the parameter λ, where $x_0 \in (0, 1)$, $\lambda \in (3.58, 4)$ et $a, b>0$. In hexadecimal representation, the number of different combinations of secret keys is equal to 2^{256}. Accordingly, the theoretical key space is not less than 2^{256}, which is large enough to resist brute-force attack [10].

4.1.5 Correlation analysis

The proposed chaotic encryption system should be resistant to statistical attacks. Correlation coefficients of adjacent pixels in the encrypted image should be as low as possible [28]. A thousand pairs of two adjacent pixels are selected randomly in vertical, horizontal, and diagonal direction from the original and encrypted images. And then, the correlation coefficient was computed using the formulas below and the results are shown in Table 2 and the visual testing of the correlation distribution of two horizontally adjacent pixels of the plain image and the cipher image produced by the proposed scheme is shown in Fig. 4. It is clear from the results of the sixth row of Table 2 and Fig. 4 that the proposed approach is resistant to statistical attacks. We can also find in Table 2 that the proposed encryption algorithm has much better statistic properties than those in [4, 22, 25, 29, 30, 31], using respectively the same standard gray scale image Lena with size 512x512.

$$r_{xy} = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{y})^2\right)}} \quad (12)$$

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \quad (13)$$

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \quad (14)$$

Where x_i and y_i are greyscale values of i-th pair of adjacent pixels, and N denotes the total numbers of samples.

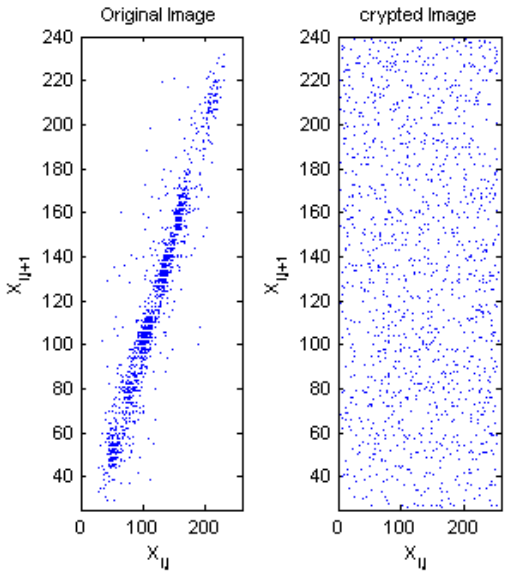


Figure 4: Correlation of horizontally adjacent pixels of the image Lena.

4.1.6 Information entropy

The information entropy can be calculated by:

$$H = -\sum_{i=1}^{2^M} p(m_i)\log_2\left(p(m_i)\right) \quad (15)$$

where M is the number of bits to represent a symbol; $p(m_i)$ represents the probability of occurrence of symbol mi and log denotes the base 2 logarithm so that the entropy is expressed in bits. It is known that if the information entropy is close to 8, the encryption algorithm is secure upon the entropy attack. The results of the third row of Table 2 show that, our scheme is better in the aspect of the information entropy than the other encryption schemes ([4, 22, 25, 29, 30, 31]).

4.2 Key sensitivity analysis

An ideal image encryption procedure should have not only a large key space but also a high key sensitivity. Key sensitivity implies that the small change in the secret key should produce entirely different encrypted image. It means that a slight change in the key should cause some large changes in the ciphered image [28]. This property makes the cryptosystem of high security against statistical or differential attacks. Fig. 5 shows key sensitivity test result. Where the plain Lena image is firstly encrypted using the test key (x_0=0.75, λ=3.393695629843, a=9, b=2). Then the ciphered image is tried to be decrypted using five decryption keys:

(i) x_0=0.75, λ=3.393695629843, a=9, b=2;
(ii) x_0=0.74, λ=3.393695629843, a=9, b=2;
(iii) x_0=0.75, λ=3.393695629842, a=9, b=2;
(iv) x_0=0.75, λ=3.393695629843, a=8, b=2;
(v) x_0=0.75, λ=3.393695629843, a=9, b=3.

It can be observed that the decryption with a slightly different key fails completely. Therefore, the proposed image encryption scheme is highly key sensitive.

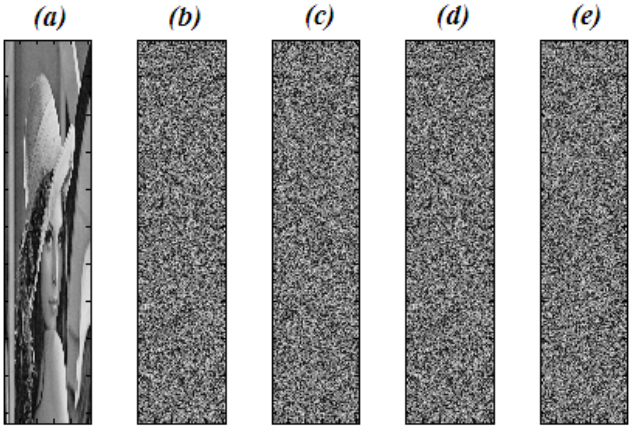


Figure 5: Key sensitivity test: (a) Deciphered image using key (i); (b) Deciphered image using key (ii); (c) Deciphered image using key (iii); (d) Deciphered image using key (iv); (e) Deciphered image using key (v).

4.3 Differential attack analysis

In general, a desirable property for an encrypted image must be sensitive to the small changes in plain-image. An opponent may make a slight change, usually one pixel, in the plain image and compare the cipher images (corresponding to very similar plain images and obtained by the same key) to find out some meaningful relationship between plain image and cipher image, which further facilitates in determining the secret key. If one minor change in the plain image can be effectively diffused to the whole ciphered image, then such differential analysis as known plaintext attack, chosen-plaintext attack, known cipher-text attack and so on, may become inefficient and practically useless.

The diffusion performance is commonly measured by means of two criteria, namely, the Number of Pixel Change Rate (NPCR) and the Unified Average Changing Intensity (UACI). *NPCR* is used to measure the percentage of different pixel numbers between two images. The *NPCR* between two ciphered images A and B of size $M \times N$ is [28]:

$$NPCR_{AB} = \frac{\sum_{i=1}^{M}\sum_{j}^{N} D(i,j)}{M \times N} \times 100 \quad (16)$$

Where,

		Chaos-based encryption algorithm						
Reference		**Proposed**	**[22]**	**[29]**	**[4]**	**[25]**	**[30]**	**[31]**
Entropy		7.9996	7.9971	7,9993	7.9992	7.9994	7.9992	7.9880
NPCR		99.693	99.621	99,603	99.6201	99.639	-	-
UACI		33.621	33.434	33,456	33.4006	33.554	-	-
Correlation coefficients of permuted image	H	0.0002	0.0097	-0,0010	0.0026	0.000707	-0.0155	0.0368
	V	-0.0030	0.0136	-0,0016	0.0034	0.002165	0.0199	-0.0392
	D	-0.0008	0.0178	0,0010	-0.0019	0.014886	0.0244	0.0068
Number of rounds		1	2	10	1	2	-	2
Key space		2^{256}	2^{167}	2.27×10^{57}	2.27×10^{57}	-	2^{451}	2^{153}
Key sensitivity		99.69 percent difference	99.62 percent difference	99,628 percent difference	99.61 percent difference	99.622 percent difference	high	99.61 percent difference

Table 2: Comparison of different chaos-based encryption algorithm.

$$D(i,j)=\begin{cases}1 & A(i,j)\neq B(i,j)\\ 0 & otherwise\end{cases} \quad (17)$$

The expected *NPCR* for two random images with 256 gray levels is 99.609 %.

The second criterion, *UACI* is used to measure the average intensity of differences between the two images. It is defined as [27]:

$$UACI_{AB}=\frac{100}{M\times N}\sum_{1}^{M}\sum_{1}^{N}\frac{|A(i,j)-B(i,j)|}{255} \quad (18)$$

For a 256 gray levels image, the expected *UACI* value is 33.464 %.

The *NPCR* and *UACI* test results are shown in Table 2. The proposed cryptosystem achieves high performance by having $NPCR \succ 0.99609$ and $UACI \succ 0.33464$ and can well resist the known-plaintext, the chosen-plaintext and the known cipher-text attacks. Also, the results of the seventh row of table 2 show that the proposed scheme requires fewer permutation and diffusion rounds than the other algorithms. Indeed, the proposed scheme requires few chaotic numbers for the generation of complex permutation and diffusion keys, which contributes to the raise of the speed performance as compared to the other algorithms ([4, 22, 25, 29, 30, 31]).

4.4 Efficiency analysis

Running speed of the algorithm is an important aspect for a good encryption algorithm, particularly for the real-time internet applications. In general, encryption speed is highly dependent on the CPU/MPU structure, RAM size, Operating System platform, the programming language and also on the compiler options. So, it is senseless to compare the encryption speeds of two ciphers image. We evaluated the performance of encryption scheme by using Matlab 7.10.0. Although the algorithm was not optimized, performances measured on a 2.0 GHz Pentium Dual-Core with 3GB RAM running Windows XP are satisfactory.

The average running speed depends on the precision used for the quantization of chaotic values. For *P*=8, the average computational time required for 256 gray-scale images of size 512x512 is shorter than 100 *ms*. By comparing this result with those presented in Ref. [28], the scheme can be said high-speed as we only used a 2.0 GHz processor and the Matlab 7.10.0 software. Indeed, the modulus and the XOR functions are the most used basic operations in our algorithm. Also, the comparison between the simulations times required at the permutation stage shows that the computational time required in our experiment is three times less than that of Chong Fu et al's. [34].

This means that the actual computational times of our scheme could be at least smaller if implemented in the same conditions than the Chong Fu et al's. algorithm. So, referring to actual fast ciphers [22, 23, 34], our proposed algorithm has a fast running speed. Such a speed is promising for real time applications of multimedia data encryption.

5 Conclusion

In this paper, we proposed a new secure and fast chaos-based algorithm for image encryption using the true accuracy of the computer. In the proposed scheme, the permutation-diffusion design based on the fast generation of large permutation and diffusion key with a good level of randomness and a very high sensitivity has been investigated. This procedure allows to use the true accuracy of the computer by using integer sequences obtained by the descending sorting of the Logistic map as a secret key in the permutation stage. This technique avoids the excess digitization of chaotic values. As consequence, the sensitivity to small changes of the initial condition or control parameters is increased. In the diffusion stage, in order to avoid the known/chosen plaintext attack, we have proposed to link keystream with both the key and original image to mask the whole image. By using these techniques, the spreading process is significantly accelerated contrary to that of Chong Fu et *al*. [34]. According to NIST randomness tests, the image sequences encrypted by the proposed algorithm have no defect and pass all the statistical tests with high P-values. Also, we proved the very good cryptographic performances of the proposed image scheme through an

extensive analysis, performed with respect to the latest methodology from this field. As a result, one round of encryption with the proposed algorithm is safe enough to resist exhaustive attack, differential attack and statistical attack. The new scheme has higher security and faster enciphering/deciphering speeds. This makes it a very good candidate for real-time image encryption applications.

Acknowledgement

J.D.D Nkapkop gratefully acknowledges the Erasmus Mundus – Action 2 for their financial support.

References

[1] H. Dobbertin, V. Rijmen and A.Sowa (2005). Advanced Encryption Standard-AES, *4th International Conference*, AES 2004, Bonn, Germany, May 10-12, Lecture Notes in Computer Science 3373.

[2] F. Riaz, et al. (2012). Enhanced image encryption techniques using modified advanced encryption standard. *Communications in Computer and Information Science*. 281, 385-396.

[3] S. Dey (2012). SD-AEI: An advanced encryption technique for images. *In Proceedings of Digital Information Processing and Communications*, Second International Conference, Lithuania, 68-73.

[4] J. S. A. Eyebe Fouda, et al. (2014). A fast chaotic block cipher for image encryption. *Communications in nonlinear science and numerical simulations*. 19 (3), 578-588.

[5] S. Li, et al. (2007). On the design of perceptual MPEG-video encryption algorithms. *In: IEEE Transactions on Circuits and Systems for Video Technology*, 214-223.

[6] S. Mohammad, S. Mirzakuchaki (2012). A fast color image encryption algorithm based on coupled two-dimensional piecewise chaotic map. *Signal processing*. 92(5), 1202-1215.

[7] A. A. Abd El-Latif, et al. (2012). Digital image encryption scheme based on multiple chaotic systems. Sensing and Imaging, *An international journal on continuing subsurface sensing technologies and applications*. 56(2), 67-88.

[8] C. Shannon (1949). Communication theory of secrecy systems. *Bell System Technical Journal*. 28, 656-715.

[9] M. Demba and O. M. Abu Zaid (2013). A Proposed Confusion Algorithm Based on Chen's Chaotic System for Securing Colored Images. *International Journal of Signal Processing Systems*. 1(2), 296-301.

[10] J-Y. Wang, G. Chen (2015). Design of a Chaos-Based Digital Image Encryption Algorithm in Time Domain. *In: IEEE International Conference on Computational Intelligence and Communication Technology* (CICT), 26-28.

[11] S. Li, et al. (2008). A general quantitative cryptanalysis of permutation-only multimedia ciphers against plaintext attacks. *Signal Processing: Image Communication*. 23, 212-223.

[12] S. Sam, P. Devaraj and R. S. Bhuvaneswaran (2012). A novel image cipher based on a mixed transformed logistic maps. *Multimedia tools and applications*, 56 (2), 315-330.

[13] X. Di, L. Xiaofeng and W. Pengcheng (2009). Analysis and improvement of a chaos-based image encryption algorithm. *Chaos Solitons and Fractals*. 40 (5), 2191-2199.

[14] A. A. Abd El-Latif, L. Li and X. Niu (2014). A new image encryption scheme based on cyclic elliptic curve and chaotic system. *Multimedia tools and applications*. 70 (3), 1559-1584.

[15] P. Jagadeesh, P. Nagabhushan and R. Pradeep Kumar (2012). A New Image Scrambling Scheme through Chaotic Permutation and Geometric Grid based Noise Induction. *International Journal of Computer Application*. 78 (4), DOI: 10.5120/13481-1181.

[16] G. Alvarez, S. Li (2009). Cryptanalyzing a nonlinear chaotic algorithm (NCA) for image encryption. *Commun Nonlinear Sci Numer Simul*. 4 (11), 3743-3749.

[17] R. Rhouma, E. Solak and S. Belghith (2010). Cryptanalysis of a new substitution-diffusion based image cipher. *Commun Nonlinear Sci Numer Simul*. 15 (7), 1887-1892.

[18] C. Li, D. Arroyo and K. Lo (2010). Breaking a chaotic cryptographic scheme based on composition maps. *International Journal of Bifurcation and Chaos*. 20, 2561-2568.

[19] R. Rhouma and S. Belghith (2008). Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Physics Letters A*. 372 (38), 5973-5978.

[20] E. Solak, et al. (2010). Cryptanalysis of Fridrich's chaotic image encryption. *International Journal of Bifurcation and Chaos*. 20 (5), 1405-14013.

[21] C. Zhu, C. Liao and X. Deng (2013). Breaking and improving an image encryption scheme based on total shuffling scheme. *Nonlinear Dynanics*. 71(1), 25-34.

[22] R. Guesmi, et al. (2014). A novel design of Chaos based S-Boxes using genetic algorithm techniques. *In: IEEE 11th International Conference on Computer Systems and Applications*. 678-684.

[23] X. Wang and Q. Wang (2014). A Novel image encryption algorithm based on dynamic S-boxes constructed by chaos. *Nonlinear Dynanics*. 75 (3), 567-576.

[24] Y. Wang, et al. (2015). A colour image encryption algorithm using 4-pixel Feistel structure and multiple chaotic systems. *Nonlinear Dynamics*. 81 (1), 151-168.

[25] Y. Wang, et al. (2011). A new chaos-based fast image encryption algorithm. *Applied Soft Computing*. 1 (1), 514-522.

[26] A. Gonzalo, L. Shujun (2006). Some basic cryptographic requirements for chaos-based cryptosystems. *Int. J. Bifurcation Chaos*. 16, 21-29.

[27] Q. Zhang, L. Guo and X. Wei (2010). Image encryption using DNA addition combining with chaotic maps. *Mathematical and Computer Modelling*. 52 (11-12), 2028-2035.
[28] A. Rukhin, et al. (2010). A statistical test suite for the validation of random number generators and pseudo-random number generators for cryptographic applications," *NIST Special Revised Publication 800-22*, April 2010.
[29] J.S.A. Eyebe Fouda, et al. (2012). Efficient Cryptosystem Based on Chaotic Sequences Sorting. *American Journal of Signal Processing*. 2, 15-22.
[30] Z. Tang, J. Song, X. Zhang and R. Sun. Multiple-image encryption with bit-plane decomposition and chaotic maps. *Optics and Lasers in Engineering*.80, 1-11.
[31] C. Fu, et al. (2011). A novel chaos-based bit-level permutation scheme for digital image encryption. *Optics Communication*. 284, 5415-5423.
[32] L. Billings and E. Bollt (2001). Probability density functions of some skew tent maps. *Chaos, Solitons and Fractals*. 12 (2), 365-376.
[33] Y. Wang, et al. (2010). A new chaos-based fast image encryption algorithm. *Applied Soft Computing*. 11 (1), 514-522.
[34] C. Fu, et al. (2012). A chaos-based digital image encryption with an improved permutation strategy. *Optic Express*. 20 (3), 2363-2378.

e-Turist: An Intelligent Personalised Trip Guide

Božidara Cvetković, Hristijan Gjoreski, Vito Janko, Boštjan Kaluža, Anton Gradišek and Mitja Luštrek
Department for Intelligent Systems, Jožef Stefan Institute, Jamova 39, SI-1000 Ljubljana, Slovenia

Igor Jurinčič, Anton Gosar, Simon Kerma and Gregor Balažič
Faculty of Tourism Studies – Turistica, University of Primorska, Obala 11a, SI-6320 Portorož, Slovenia
E-mail: boza.cvetkovic@ijs.si

Keywords: tour planning, recommender system, route optimization

Received: June 16, 2016

We present e-Turist, an intelligent system that helps tourists plan a personalised itinerary to a tourist area, taking into account individual's preferences and limitations. After creating the route, e-Turist also offers real-time GPS guidance and audio description of points of interest visited. Here we focus on two main components, the recommender system and the route planning algorithm. We also present some use cases to highlight e-Turist functionalities in different configurations.

Povzetek: Predstavljamo inteligentni sistem e-Turist, ki turistom pomaga izdelati personaliziran načrt poti po določeni turistični regiji. Pri tem upošteva posameznikove želje ter omejitve. Po izdelavi poti e-Turist omogoča tudi vodenje s pomočjo sistema GPS in opis zanimivosti s pomočjo zvočnih datotek. V članku podrobneje predstavimo dve glavni komponenti sistema, in sicer priporočilni sistem ter algoritem za načrtovanje poti. Poleg tega predstavimo nekaj primerov uporabe, ki prikažejo delovanje e-Turista v različnih konfiguracijah.

1 Introduction

Tourism is one of the fastest growing global industries. Though suffering a setback during the late-2000s recession, the tourism sector has seen a robust growth for six consecutive years,[1] with the number of international tourist arrivals growing by 4.4 % from 2014 to 2015 and totalling 1.2 billion worldwide. From 25 million international tourists in 1950, the number is forecast to reach 1.8 billion by 2030.[2] Currently, tourism generates 9 % of global GDP through direct and indirect impact, creates 1 in 11 jobs, and represents 6 % of world's exports. At least 53 % of international tourists (600 million) travelled for holidays, recreation, and other forms of leisure. In addition, the number of domestic tourists is estimated to be between 5 and 6 billion.[2] Tourism represents an important part of economy in individual countries, contributing to 9-13 % of national GDPs in countries such as Italy, Germany, France, and Slovenia, whereas the contribution in countries such as Croatia, Malta, and Iceland is over 23 %.[3]

Tourists can plan their trips either individually or using services provided by tourist agencies. Each approach has advantages and disadvantages. By joining an organised group, little planning is required. Such groups typically employ a licensed tour guide who is familiar with the pre-planned itinerary and individual stops on the route. However, such groups typically focus on a smaller number of prime-level tourist destinations, suitable for large group visits. A fixed itinerary may also conflict with the interests of group members who would prefer spending more time at certain locations or visiting nearby attractions that are not included in the route. Individual tourists are more flexible in designing their itinerary and choosing points of interest (POI) to visit. Individual tourists may also find less-known destinations attractive to visit, often even more so, as they are better suited for smaller groups of people and are more diverse in the activities they offer. However, the initial planning is more complex. The tourists have to compile travel information from various sources, such as tourist guidebooks, websites, tourist information centres, etc. Apart from scattered information sources, they have to consider opening times, geographical distribution of POIs (distances between locations), and the availability of services such as accommodations and restaurants. This makes designing a good itinerary quite difficult. However, since each tourist has unique preferences related to type of activities, choice of food, special interests, and potential limitations due to physical or other impairments, fixed itineraries from a guidebook or similar source are not satisfactory. A platform containing all relevant information about a certain region that would allow simple creation of personalised itineraries could represent a significant advantage both for tourists (by facilitating the planning) and the local tourism sector (by highlighting local POIs and services that would otherwise stay overlooked).

In her extensive research, Molz is introducing the notion of "flashpackers" (affluent interactive travellers with a budget, higher than backpackers, spending freely for activities at their chosen destinations), arguing that

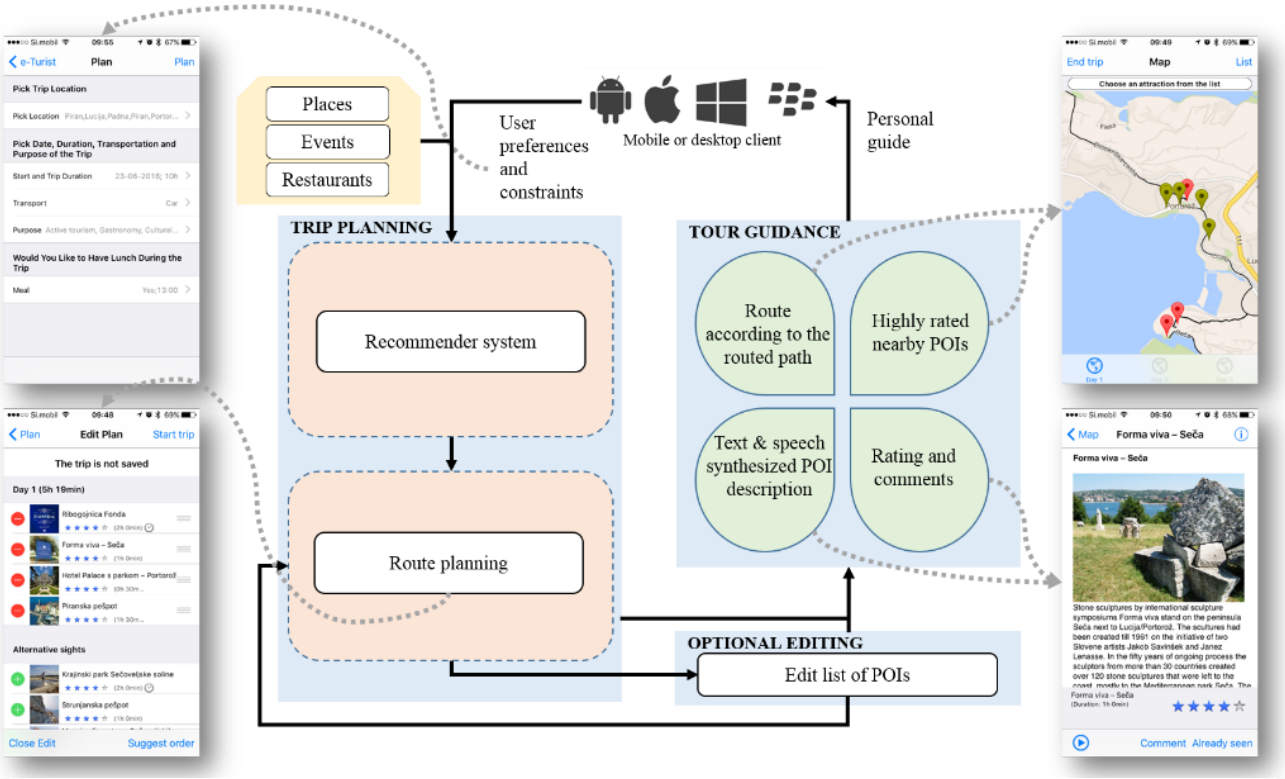


Figure 1. e-Turist system overview.

the rapidly evolving context of new mobile and media technologies has made interactive travel an ever more significant element of modern social life, especially in a mobile world.[4] In the last decade, extensive research has been carried out with the goal of improving tourist experiences,[5] [6] [7] partially stimulated by the fact that smartphones have become powerful computing tools with access to cloud-based services, which allow personalisation and real-time functionality. The potential for such applications is in creating personalised tours which, as discussed above, improve user experience over guidebooks that offer generic visitor tours through a city or a region.[5] Personalisation is achieved through user specifying their preferences and constrains, such as the available time and start and end point of the tour. While the current existing applications use various approaches to offer best experience, they have some common components. Typically, a recommender system is used [5] [6] [7] to create a list of most appropriate POIs. Some implementations are simple, for example recommending the nearest attractions based on location. Other are more advanced and use artificial intelligence approaches, such as various types of filtering (knowledge-based, demographic, hybrid, etc.), automatic clustering algorithms, approximate reasoning methodologies, such as fuzzy logic, or ontologies. Often, a route planning functionality is included. Here, the task of the application is to present the optimal route between POIs. Different approaches of solving the Travelling Salesman Problem (TSP) [8] are implemented in this module.[5] An example of such systems is a pilot study by Garcia et al.,[9] which was developed for the city of San Sebastian in Spain and uses a basic recommender system and route planning. However, the main issue with such systems is that they were mostly developed as pilot projects and/or have not left the academic environment.

On the other hand, there are several mayor players in the tourist industry that use various approaches. TripAdvisor [10] contains extensive lists of attractions with user reviews and also allows users to book flights or hotels. Other applications are more specific. For example, Triposo [11] includes a ranked list of attractions by category, Roadtrippers [12] is designed for car travellers and shows potential POIs close to the planned route, and Route Perfect [13] includes recommendations based on explicit user preferences and a route planning component, but lacks day-to-day sightseeing itineraries in individual cities.

As seen in this quick overview, several approaches have already been developed to help tourists, but we were unable to find a widely-used product that would contain the complete functionality, namely a recommender system, tour routing, and a real-time guiding component. In this paper, we present our solution, called e-Turist (e-Tourist),[14] [15] which is an intelligent platform designed to help individual tourists or small groups prepare a customised sightseeing/travel plan and offer them an experience comparable to one offered by a professional tourist guide. To some extent, the platform promotes smaller tourist-oriented businesses, which is important, since they contribute to local economy and ensure jobs in local environment. e-Turist was initially developed for tourist areas in Slovenia and is now being expanded to include destinations worldwide. The system can be accessed either through a mobile application or a website. Tourists

enter their preferences and the system prepares a customised itinerary which includes the most appropriate points of interest with the shortest route connecting them. In addition, the application guides the tourist using GPS and offers information about attractions either in text or audio format. We discuss the main components of e-Turist, namely the recommender system that creates a list of most interesting places for the tourist to visit, and the route planner which calculates the optimal route between these places. We also present some use cases, which highlight the main functionalities of the system.

2 System overview

The e-Turist system is designed as a web service (software as a service - SaaS) and is accessible either through a web-browser or a smartphone app. The system architecture is shown in Figure 1.

When opening the application, the user enters his preferences regarding the trip. Those may be basic, such as the trip duration, or more detailed, by creating a user profile that includes information such as age, education, or budget (these parameters were considered relevant by tourism experts). The application then creates the proposed itinerary, which is done in two steps. First, the recommender system creates a list of appropriate POIs chosen dynamically according to the user's preferences. In the second step, the route planning module proposes the optimal route between a subset of chosen POIs, based on the available time and some other parameters, such as stops for meal during the trip. The user can then customize the proposed route in one or more steps. The recommender module and the route planning module will be presented in more detail in the following sections.

When on the tour, the application uses real-time guidance based on the GPS data. When reaching each stop, the application offers information about the POI, either in text or audio form. During the tour, the application also highlights nearby POIs so that the user can decide whether to make a detour. At the end, the user can rate the visited POIs with 1–5 stars.

The browser application is essentially identical to the mobile one, with the exception of the GPS guidance being absent. Interchangeable use is possible, such as editing and saving the route via the browser application and opening it on the mobile device.

e-Turist employs an audio module, which offers audio POI descriptions. To create audio descriptions in the Slovenian language, the voice synthesizer Govorec [16] is used. Govorec was developed in collaboration between Jožef Stefan Institute and the company Amebis. For other languages – currently, English, German, and Italian are supported – Microsoft Speech API [17] is used. The audio files are generated automatically when a POI description is entered into the database and are stored there for user access.

The administration module is used to edit the database of POIs. For destinations (regions), geographical area and approximate radius are defined. For each POI, a description, one or more images, and metadata are defined. The metadata includes the type of the POI, opening hours, accessibility, expert rating, suitability for different types of tourists, etc. The administration module highlights missing information in the descriptions, and allows monitoring the visits and user ratings for individual POIs, which allows tourism workers to improve the service.

3 Recommender system

In order to prepare the most suitable itinerary for individual tourists, the recommender system module uses a combination of constraints filtering, knowledge-based recommendation and collaborative filtering, as shown in Figure 2.

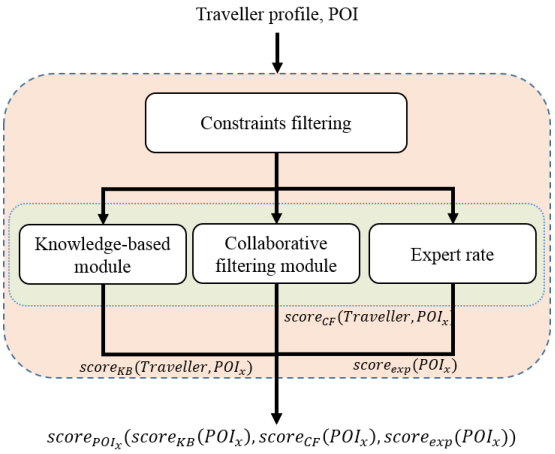


Figure 2. Recommender system schematics.

The **constraints filtering module** utilizes "hard" constraints that exclude the POIs that do not meet the user's limitations and key requirements. These constraints are (i) the location, (ii) the purpose of the trip – currently the options are active tourism, cultural heritage, entertainment and gastronomy, (iii) the opening hours of the POIs, and (iv) mobility limitations for physically impaired users. For example, if a user is interested in active tourism, he will not be suggested to visit museums but venues such as adrenaline parks instead.

The **knowledge-based module** computes the distance between each POI and the user based on four sets of expert-defined characteristics: age, education, country of residence, and budget. There are five age groups: age up to 26, 27 to 36, 37 to 45, 46 to 55, and 56 and higher. There are three education groups: primary, secondary, and tertiary, and also three budget groups: low, medium, and high. The country parameter corresponds to countries whose tourists often visit that POI. Each POI is assigned one or more groups for each characteristic, except for budget, where it can only have one value. Each group is assigned numerical value (e.g. 0 is low, 1 is medium, and 2 is high budget), respectively, which are used to compute the Euclidian distance between the user and POI characteristics, later transformed into score of the individual characteristic.

The score for age, education, and country characteristic is computed using Equation 1. The absolute distance between the user characteristic and POI

characteristic is divided by number of groups per characteristic and subtracted from the perfect-fit value of 1. The score for budget is computed with the same approach unless the $user_{budget} \geq POI_{budget}$, in which case the score is always set to the perfect-fit value of 1. In case the characteristic value is not defined for a POI or the user, the score is set to the medium-fit value of 0.5.

$$score_{char} = 1 - \left|\left(\frac{user_{char} - POI_{char}}{char_{groups}}\right)\right| \quad (1)$$

The final score $score_{KB}$ of the knowledge-based module is calculated using Equation 2. The score is afterwards normalised to interval from 1 to 5.

$$score_{KB} = \frac{score_{age} + score_{edu} + score_{country} + score_{budget}}{4} \quad (2)$$

The **collaborative filtering** uses a memory-based approach to assign the $score_{CF}$ for the user. Each instance represents one user. Its feature vector is composed of ratings per POI given by the individual user. In case the user did not rate a POI, that value is defined as a missing value. We used the *k*-nearest neighbour algorithm [18] to find *k* similar users, once a sufficiently high number of users have been recorded in the database. The final score for an individual POI is an average value of scores per POI for the *k*-nearest neighbours.

The final result of the hybrid recommender system is the final score calculated with Equation 3. It is a weighted sum of the knowledge-based rating, the collaborative-filtering rating, and the expert rating (POI rating provided by experts).

$$score = w_1 score_{KB} + w_2 score_{CF} + w_3 score_{exp} \quad (3)$$

We discuss the weight values in the following subsection.

3.1 Experimental evaluation and parameter settings for the recommender module

The recommender system was tested on data of 24 users with different age and background who were given a list of 90 POIs from the Heart of Slovenia region [19] . The users were asked to rate the POIs they are familiar with from 1 to 5 stars. These ratings were then compared to the recommender system modules, with the goal of accurately predicting the rating of POI as would be given by the current user. This helped us define the final weights of Equation 3.

Each recommender system module was evaluated for the performance using the mean absolute error (MAE) over all 90 POIs, as presented in Equation 4. The lower the MAE value, the better the prediction of the modules.

$$MAE = \frac{\sum_{i=0}^{n} |(score_{true} - score_{predicted})|}{n} \quad (4)$$

In the first step, we evaluated the baseline approach which rates all POIs with the score 3, being the medium score. The MAE value of the baseline approach is 1.05 rating.

In the second step, the expert rating was evaluated. It amounted to 0.99 rating.

In the third step, the knowledge-based module was evaluated. The users' budget preferences and demographic data were used to compute the $score_{KB}$. The evaluation of the score estimates per user yielded MAE of 0.98.

Next, we evaluated the collaborative filtering module. Before evaluation we had to define the number of neighbours that would be used by the *k*-nearest neighbours algorithm. We tested the algorithm for *k*=1 to *k*=5 and settled for *k=4* as it returned the best results. The MAE of the collaborative filtering module using 4-nearest neighbours algorithm was 0.87 score.

Before setting the weights of the three modules, we decided that the weight of the expert rating should be smaller than the weights for other ratings, otherwise the recommendations would not be personalised. We decided to assign it to 0.2. Since the difference between the knowledge-based and collaborative filtering module was not large, we gave them equal weights of 0.4 as shown in Equation 5. In case the expert score is missing, the recommender modules are assigned weights of 0.5 (Equation 6), and in case the knowledge-based or the collaborative filtering rating is missing, the 0.8 weight is assigned to the remaining one (Equation 7 and 8). If there is only one score, it is assigned the full weight.

$$score = 0.4(score_{KB} + score_{CF}) + 0.score_{exp} \quad (5)$$

$$score = 0.5 score_{KB} + 0.5 score_{CF} \quad (6)$$

$$score = 0.8 score_{KB} + 0.score_{exp} \quad (7)$$

$$score = 0.8 score_{CF} + 0.score_{exp} \quad (8)$$

The results of the above equations were compared to individual modules. The results are presented in Table 1. The MAE value of the baseline approach is 1.05 score and the MAE of the final rating is 0.86 score, which is better than the MAE of the knowledge-based, collaborative-filtering rating, and expert rating alone.

Approach	**MAE (rating)**
Baseline	1.05
Expert score	0.99
Knowledge-based module	0.98
Collaborative filtering module	0.87
e-Turist final rating	**0.86**

Table 1. Results of the baseline rating, individual modules ratings, and the e-Turist recommender system final rating.

4 Route planning algorithm

The task for the route planning module is to prepare the best route for the tourist from the list of POIs generated

by the recommender system, taking into account the attractiveness of individual POIs and the time limitations of the tourist. This task is reminiscent of the well-known knapsack problem,[20] where the best set of items (in our case POIs) with weights W_i (the time needed to visit the POI) and values V_i (POI attractiveness) have to be chosen so that the overall weight does not exceed the limit and that the total value is maximal. Such problem can be solved in a pseudo-polynomial time with dynamic programming.[21] However, in our practical implementation, the algorithm needs additionally to take into account the time needed to visit the chosen items (POIs) – which is an estimation problem by itself. The path duration estimation problem opens a new sub-problem inside the knapsack problem, i.e., how to find the route with minimal duration, given the POIs. This sub-problem is also a known problem in the theory of computation, called the TSP.[8] Therefore, in our case we have a "modified knapsack problem" where the total value of the chosen items changes dynamically (with each algorithm iteration). Additionally, the path estimation is computationally expensive process, because it requires solving an NP-hard problem, i.e., TSP. Moreover, the final algorithm execution time should be in the range of seconds, because it will be used in a real-time POI recommender application, where the user needs instant feedback from the system. Because of these reasons, two simplifications were proposed: a greedy approach for knapsack problem (POIs ordered by value) and an adapted TSP for path duration estimation (finds near optimal solution).

The first step in our algorithm is the estimation of the importance of an individual POI (how attractive a POI is – POI value). We defined the POI value considering three factors:

- POI's rating (provided by the recommender system)
- POI's visit duration
- POI's local reachability duration

The first factor is a value that is provided by the recommender system ($score$) which varies from 1 to 5 (attractiveness of the POI). The next factor, the POI's visit duration, represents the average time that a tourist needs in order to see the POI, which is defined by an expert in the POI database. The final factor, the POI's reachability duration, is a variable that represents how far a POI is from its nearest neighbours. In other words, if a POI is far from the rest of the POIs, the value for this variable would be greater compared to the reachability duration of the other POIs. Using this information, the POIs that are "outliers" (far from the rest of the POIs) are "punished" because the tourist needs more time to reach them. For the estimation of this variable we used a partial implementation of the Local Outlier Detection (LOF) algorithm. In particular, we used the local reachability distance (*lrd*) metric in order to estimate how far a POI is from its neighbours. The LOF algorithm and its mathematical definitions are described by Breunig et al [22] .

The mathematical definition of the POI importance, which includes all the three factors, is presented in Equation 7.

$$V = \alpha * score + (1 - \alpha) * (1 - P_{norm}) * rate \quad (7)$$

The variable $score$ is the POI's rating, which varies from 1 to 5. The variable P_{norm} represents the normalised value (from 0 to 1) of the P, which is a sum of the POI's visit duration (V_{d}) and the POI's local reachability distance (*lrd*) – note that V_{d} and *lrd* take values in hours:

$$P = V_d + lrd \quad (8)$$

Because the idea is to "punish" the POIs that require longer time to visit and the ones that are far away, the normalised P is subtracted from 1 (the bigger the value of P_{norm}, the less important the POI). α is a parameter regulating the importance of the evaluation value (V) on one side, and the POI's visit duration and POI's local reachability distance (P) on the other side. The empirical analysis of the data showed that 0.5 is a reasonable trade-off value for α. This way, both sides of the equation are equally weighted in the final importance value V. To summarize, the first term in Equation 8 is considered as it is, while the second term is reduced by the factor corresponding to the time needed for the visit ($1 - P_{\text{norm}}$).

In the next step of the algorithm, all the POIs are ordered by the importance value V. Using a greedy strategy, the algorithm then adds items (POIs) in the knapsack until the limit is reached. With each POI added, the weight of the knapsack is checked – if the weight (time duration) of the chosen POI combination is below the maximum weight (total available time of the tourist). In addition, with each added POI, a TSP algorithm estimates the path duration, which is also checked with the time limit of the tourist. This way, a near optimal combination of POIs is found.

Additionally, the algorithm checks for nested POIs, since there may be more than one POI at the same location. For example, the Ljubljana castle additionally has a museum and a tower. Therefore, if the algorithm has chosen some of the nested POIs in the optimal route, it additionally adds all the other nested POIs at that location, and by doing so it also updates the time for the visit and checks the constraints of the knapsack.

After the combination of POIs is found, the algorithm checks whether the user prefers to start from the nearest POI. If this is the case, the order of the POIs is recalculated with a modified version of the original TSP which creates a path using a fixed start POI.

In the final step of the algorithm, it is checked whether the tourist has chosen multiple days for sightseeing. In the case of a multiple-days visit, the POIs are segmented into groups, each group corresponding to one day of the trip. Additionally, it is checked if the user plans a meal at a particular hour of the day. If this is the case and there are restaurant-POIs in the list of POIs, the best (according to the evaluation value and the location) restaurant is chosen. That day's route is modified in such a way that the tourist is near the restaurant during the

Algorithm 1: Route planning algorithm

```
Input:
    allPOIs //POIs from the recomm. system
    transport //the means of transport
    numDays //number of days for the visit
Result:
    FinalPOIs //The near-optimal list of POIs
    AlternativePOIs //Alternative POIs

duration = 0
//order by value V
orderedPOIs = OrderPOIsByValue (allPOIs)

//start adding POIs ordered by the value, add the rest
to the alternative list
For POI in orderedPOIs
  If duration + POI.duration < totalDuration
    tempPOIs.add(POI)
    pathDuration = TSP(tempPOIs, transport)
    duration =
    updateDuration(POI.duration, pathDuration)
    If POI is child at location
      tempPOIs.add (POI.parent)
      duration =
      updateDuration(POI.parent.duration)
    If POI is grandchild at location
      tempPOIs.add (POI.grandParent)
      duration = updateDuration
      (POI.grandParent.duration)
  Else
  AlternativePOIs.add(POI)
End

//Find the near-optimal path and time to visit all
chosen POIs
Solution = TSP(tempPOIs, transport)

//Take care of the parent-child relation
FinalPOIs= OrderByParentChildRelation (Solution)

//If the user is less than 1 km away from some of the
chosen POIs, start from the nearest POI
If StartLocation.DistanceToNearestPOI < 1km
    FinalPOIs= Reorder(FinalPOIs)

//if there are multiple days, segment the solution to
multiple lists of POIs, for each day
If numDays > 1
    FinalPOIs= SegmentByDays (FinalPOIs)

Return FinalPOIs, AlternativePOIs
```

previously chosen meal time. This is done by dividing the problem in two segments (before and after the meal) and calling the TSP solver for each. First, for before the meal, the end location is fixed to the restaurant, and next, for after the meal, the start location is fixed to the restaurant.

The pseudo code of the route planning algorithm is given with Algorithm 1.

Since the TSP solves a sub-problem in the general knapsack problem, its execution time needs to be in the range of milliseconds. Therefore, for its implementation, we considered an open-source algorithm [23] that finds a near-optimal solution. It is a greedy approach with additional optimization mechanisms. The empirical tests showed that for our scenario (up to 200 POIs) it almost always finds the optimal solution, and also the execution time is acceptable i.e., several milliseconds. Additionally, we modified the original algorithm so that it can use start and end POI. This option is required when the users wants to start from the closest location to them (e.g. by using the GPS signal of the tourist's smartphone). Also, when the user selects meal-time, the path is divided into two parts: before and after lunch.

In order to decrease the execution time, we call the TSP algorithm only when the time needed to visit all the POIs reaches a predefined threshold of 80% of the total available time. In other words, when the time required to visit the POIs reaches 80% of the total available time, the TSP is called to estimate the exact path duration. Otherwise, every time a POI is added, the path is estimated simply by adding the path duration to the nearest POI. A detailed evaluation of the routing algorithm is a problem on its own and will not be addressed in this paper.

5 Use cases

In this section, we show different aspects of e-Turist functionality. Let us consider three different tourists, Ada, Bob, and Ted. The details about these users are presented in Table 2.

		User		
		Ada	**Bob**	**Ted**
Profile	Age	/	60	/
	Country	/	Germany	/
	Budget	/	high	/
	Education	/	tertiary	/
	Mobility lim.	/	yes	/
Interest	Active tourist	✓	×	×
	Gastronomy	✓	×	×
	Entertainment	✓	×	×
	Cultural heritage	×	✓	✓
	Include lunch	✓	×	✓
Transport	Car	✓	✓	×
	Walking	×	×	✓
Location	Slovenian Istria	✓	✓	×
	The Hague	×	×	✓

Table 2. Users and their interests. Note that Bob creates his profile only in the second step of the use case.

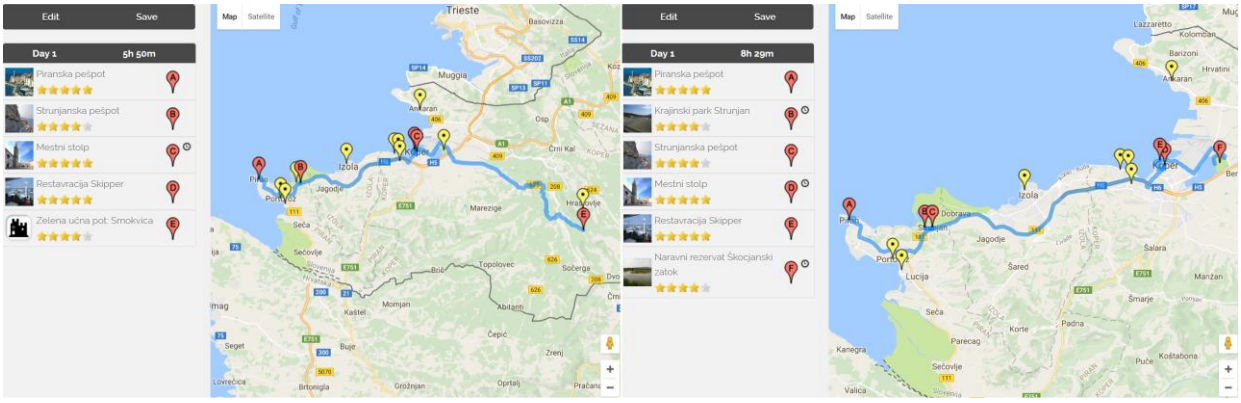


Figure 3. Left: Initial route, proposed for Ada. The sidebar on the left shows the list of the recommended POIs (red balloons), while the right side shows the route on the map (blue line) and other nearby POIs (yellow balloons). Expert ratings are marked with stars. Right: The new route, proposed for Ada, after she manually edits the POI list.

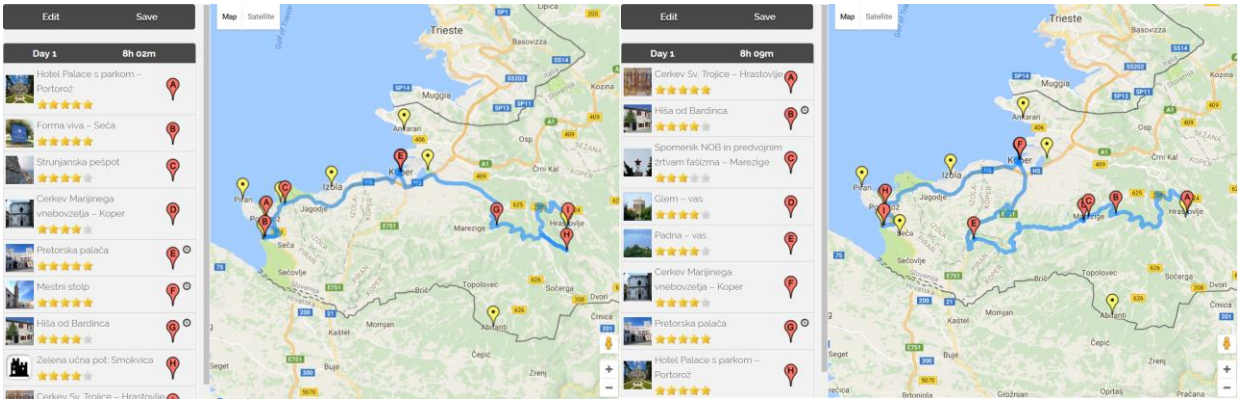


Figure 4. Left: Initial route, proposed for Bob, who is interested in cultural and natural heritage sites. Right: The new route, following Bob's creation of user profile.

Ada and Bob would like to spend a day in the Slovenian Istria region on Thursday, 1 September 2016. Both start the trip at 10 am and plan to spend 8 hours sightseeing. Both of them have a car available as a means of transport and Ada chooses to have lunch around 1 pm. Neither of them has yet created a user profile, which is likely for first-time users. The recommender system will therefore only consider the information from the constraints filtering module and the expert rating. Ada is interested in active tourism, gastronomy, and entertainment while Bob is interested in cultural and natural heritage.

The initial route proposed for Ada can be observed on the left side of Figure 3. After it is created, Ada decides to modify the proposed itinerary by clicking the "Edit" button which allows to remove the suggested and add other nearby/alternative POIs using drag and drop. When clicking "Update", Ada can choose whether to use the POIs in sequence as manually selected or to automatically reorder the sequence into an optimal route. Figure 3, right, presents the new proposed trip after choosing the automatic reordering. This new route requires longer than 8 hours to complete. If Ada decides that this is too long, she can edit the itinerary again, perhaps removing some of the POIs.

On the other hand, Bob has different preferences than Ada, which will reflect in different proposed itinerary (Figure 4, left). It is, however, possible for some POIs to overlap since they may be listed in more than one category, e.g. active tourism and cultural and natural heritage.

In the next step, Bob, who is a 60-year old German with tertiary education, high budget, and mobility limitations, creates his user profile. This allows the knowledge-based module and the collaborative filtering module to modify the list of POIs to better match Bob's profile. The new proposed route for Bob looks rather different (Figure 4, right). The constraints filtering module has first removed POIs that are less appropriate for people with mobility limitations, such as Strunjan trail (Strunjanska pešpot) and Smokvica educational trail (Zelena učna pot: Smokvica). In addition, the knowledge-based module has added the picturesque old villages Glem and Padna to the itinerary, since these are destinations that are likely to appeal to Bob's demographics, and also added the Second World War memorial (Spomenik NOB) - although it only has a 3-star expert rating. Additional alternative POIs appear on the map as well. Since Bob has not rated any POIs yet, the collaborative filtering has no specific ratings (for

Bob) to use for finding similar users and it therefore does not contribute to the POI selection in this use case. Once Bob starts rating visited POIs, the collaborative filtering will contribute to the final rating of the POIs in future trip planning.

One may observe that most POIs in routes for Ada and Bob have 4 or 5-star rating. They both travel by car, therefore they can cover larger distances during the trip, so the algorithm can include several POIs with high ratings.

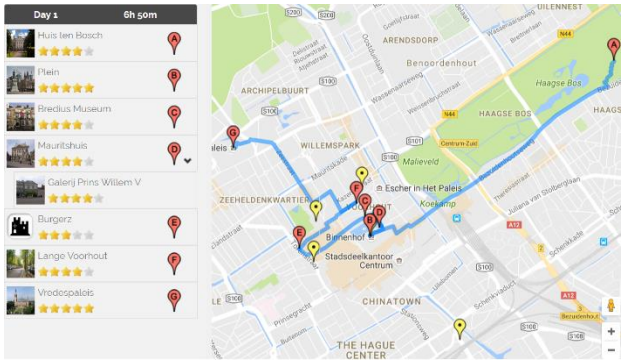


Figure 5. Proposed route for Ted, who explores The Hague on foot.

e-Turist also works on a city level. Let us consider Ted, also a first-time user, who is interested in cultural and natural heritage sites in the city of Hague in the Netherlands. Ted will explore the city on foot and use the same time window as Ada and Bob. The proposed route for Ted is shown in Figure 5. Here, we can also see an example of a nested POI, since Prince William V Gallery is located within the Mauritshuis complex which is a POI on its own.

6 Conclusion

We present a personalised trip planner that aims to improve tourist experience by facilitating the preparation of the trip and offering real-time guiding. e-Turist can be accessed either via browser or mobile application.

The platform is built on a POI database, which is easy to manage and expand. Based on the user profile and interests, the recommender system first creates a list of POIs which the user would find appealing. In order to create the list, the system uses a combination of constraints filtering, expert knowledge and collaborative filtering. In the next step, the route planning algorithm creates the optimal route between a subset of POIs. The algorithm uses the concepts from the TSP problem and the modified knapsack problem. The guiding component contains real-time GPS guidance and audio descriptions of the POIs, created using a speech synthesizer. It also allows the users to rate POIs, which serves for the collaborative filtering and as feedback available to tourism workers in the administration module.

Initial tests of the recommender system turned reasonably accurate (Section 3.1), however, since the current set of registered users is rather small, we expect the collaborative module to perform better in future. The advantage of e-Turist is that is easy to adapt to different regions. Initially, it was developed as a pilot project on two regions in Slovenia but is now being expanded to include regions worldwide. In the future, we also plan to integrate existing POI databases.

Acknowledgement

The e-Turist project was funded by Slovenian ministry of education, science and sport: call for proposals for co-funding of projects developing e-services and mobile applications for public and private non-profit organizations. We would like to thank the Faculty of Tourism Studies – Turistica and Municipality of Litija who provided the data and expert knowledge.

References

[1] UNWTO 2016, http://media.unwto.org/press-release/2016-01-18/international-tourist-arrivals-4-reach-record-12-billion-2015
[2] UNWTO Tourism Highlights, 2015 Edition
[3] Knoema, https://knoema.com/atlas/topics/Tourism
[4] Jennie Germann Molz, (2012). Travel connections: tourism, technology and togetherness in a mobile world. Routledge, New York.
[5] Wouter Souffriau, Pieter Vansteenwegen, Tourist Trip Planning Functionalities: State–of–the–Art and Future, Current Trends in Web Engineering, volume 6385 of Lecture Notes in Computer Science (2010) 474-485
[6] Damianos Gavalas, Charalampo sKonstantopoulos, Konstantinos Mastakas, Grammati Pantziou, Mobile recommender systems in tourism, Journal of Network and Computer Applications 39 (2014) 319–333
[7] Joan Borràs, Antonio Moreno, Aida Valls, Intelligent tourism recommender systems: A survey, Expert Systems with Applications 41 (2014) 7370–7389
[8] Gerhard Reinelt, The Traveling Salesman: Computational Solutions for TSP Applications, Springer Berlin Heidelberg (1994)
[9] Ander Garcia, Olatz Arbelaitz, Maria Teresa Linaza, Pieter Vansteenwegen, and Wouter Souffriau, Personalized Tourist Route Generation, ICWE 2010 Workshops, LNCS 6385, pp. 486–497, 2010. Springer-Verlag Berlin Heidelberg 2010s
[10] TripAdvisor https://www.tripadvisor.com/
[11] Triposo, https://www.triposo.com/
[12] Roadtrippers, https://roadtrippers.com/
[13] Route Perfect, https://www.routeperfect.com/
[14] e-Turist, https://www.e-turist.si/
[15] Igor Jurinčič, Anton Gosar, Mitja Luštrek, Boštjan Kaluža, Simon Kerma, Gregor Balažič, E-tourist: electronic mobile tourist guide. V: Peace, culture and tourism: collection of papers. Novi Sad: Faculty of Sciences, Department of Geography, Tourism and Hotel Management, 2013, str. 182-191.

[16] Amebis Govorec. http://govorec.amebis.si/
[17] Microsoft Speech API, http://www.microsoft.com/
[18] D. T. Larose. k-Nearest Neighbor Algorithm, Discovering Knowledge in Data, pp. 90-106, John Wiley & Sons, Inc, 2005.
[19] Srce Slovenije, http://www.srce-slovenije.si/
[20] Hans Keller, Ulrich Pferschy, David Pisinger, Knapsack Problems. Springer Berlin Heidelberg (2004)
[21] Dynamic Programming Knapsack 0-1 Problem. http://www.geeksforgeeks.org/dynamic-programmingset-10-0-1-knapsack-problem/
[22] Breunig, M. M.; Kriegel, H.-P.; Ng, R. T.; Sander, J. (2000). "LOF: Identifying Density-based Local Outliers". Proceedings of the 2000 ACM SIGMOD international conference on Management of data. SIGMOD '00: 93–104.
[23] Travelling salesman problem, Python implementation: https://github.com/dmishin/tsp-solver

Motivating Cultural Heritage Artifacts Presentation Using Persuasive Technology

Jernej Vičič
Faculty of Mathematics, Natural Sciences and Information Technologies
University of Primorska
E-mail: jernej.vicic@upr.si

Tine Šukljan
Faculty of Mathematics, Natural Sciences and Information Technologies
University of Primorska
E-mail: tine.sukljan@upr.si

Keywords: mobile application, persuasive technology, cultural artifacts, cultural heritage

Received: October 27, 2016

We have developed an information system in the scope of a project for the preservation and popularization of architectural heritage. The system aims at promoting archaeological tourism and contains a mobile application and an interactive on-line application. The mobile application guides the user on a pre-prepared learning paths and draw attention to the points of interest in the vicinity. At each point of interest it proposes to the user a choice of the level of information, he wants to learn. The user is motivated using persuasive technologies. The backbone of the system is a web server which serves as a main entry point for adding paths into the system and as a feeder of the data for both the mobile and online application.

Povzetek: V sklopu projekta za ohranitev in popularizacijo arhitekturne dediščine smo razvili informacijski sistem. Sistem, namenjen spodbujanju arheloškega turizma, vsebuje mobilno in spletno aplikacijo. Mobilna aplikacija vodi uporabnika po v naprej pripravljenih učnih poteh in ga opozarja na arheološki točki v bližini. Na vsaki točki, aplikacija ponudi uporabniku izbiro, koliko informacij želi o posamezni točki izvedeti. Uporabnik je motiviran k obiskovanju predstavljenih točk s prepričljivimi tehnologijami (persuasive technologies). Hrbtenica sistema je spletni strežnik, ki služi kot glavna vstopna točka za dodajanje poti v sistem in kot pošiljatelj podatkov tako mobilni in spletni aplikaciji.

1 Introduction

In the area of the Roman maritime villa and its background there is an extensive web of specimens of cultural and natural heritage of immense value nationwide. The archaeological site of "Simonov zaliv" (The Bay of St. Simon) was first mentioned in the 16th century. In the past years some activities were conducted for the valorisation and promotion of the site; since 2010 the University of Primorska, specifically the Institute for the Mediterranean Heritage of the Science and Research Centre is responsible for the management of this monument (owned by the Municipality of Izola). The Project AS – Projekt Arheologija za Vse, Project Archaeology for All[1] – tries to establish a modern archaeological park through these steps:

1. restoration, conservation in protection of the archaeological site Simonov zaliv;
2. education and training in the field of archaeological didactics and enhancing public awareness on the meaning of archaeological heritage;
3. enhancing and improving the accessibility of the monument;
4. planning tourist itineraries connecting archaeological sites of the Slovene coast, thus enhancing the appeal of this particular area in the segment of archaeological tourism.

The last two steps propose, among other activities, creation of an interactive on-line and mobile platform that enhances archaeological awareness and supports archaeological tourism. This paper presents the latter. A mobile application that guides the user through prepared archaeological paths with the accompanying server framework that enables input of paths and materials for each site. The server framework serves the mobile applications with new data on demand.

The paper is organized as follows: motivation is presented in Section 3 followed by the description of the methodology used in design and implementation of the presented system described in Section 4. Section 5 summarizes the achieved goals and presents a typical use-case. Section 6 discusses the results and presents the future work.

[1]Project Archaeology for All: `https://www.project-as.eu/sl/`

2 State of the art

Persuasive technologies [1] and Gamification [2] in particular was used to

A critical review of the gamification trend in tourism, the concept of gaming and gamification, intrinsic and extrinsic motivation of gamification elements and benefits of gamification are presented in [3]. A set of best practices of the application of games and gamification concepts, to create innovative products and services for the travel and tourism industry is presented in [4] and in [5]. A similar approach using gamification with geographical maps is the Geocaching initiative with a big community [6].

3 Motivation

The application will guide the user through existing and new walking and cycling routes, which are established in the Municipality of Izola and surroundings. User's attention will be focused to the fascinating natural and cultural points of interest along the way, with an emphasis on culture. The app will offer a good addition to the existing cycling and walking infrastructure trails (a few such tracks are described on the web and printed leaflets), and form the basis for possible further development of the entire network of trails. The application will alert the user to the point of interest and offer a choice of three levels of information enabling information presentation according to user's interest. Textual information will be supplemented with visual and video materials. The application will be distributed via the Internet or in the park – the starting point of the archaeological site. Three routes are currently prepared:

- circular walkway San Simon – Strunjan (a combination of P13 and P14 routes from the Slovenian net of hiking rotes),
- circular cycling route San Simon – Korte – Baredi (K16 - cycling route),
- route visiting Izola's cultural monuments (walk around the city).

4 Methodology

4.1 Requirements

These were the requirements at the start of the project:

- Mobile application that supports the two most used mobile platforms:
  - Android,
  - iOS;
- Ability to get new paths with no installation process;
- Seamless updates;
- Simple installation process;
- Simple, yet powerful object administration, possibly using on-line applications;

4.2 Point of interest

The points of interest (POI) were selected by te experts (historians and tourist guides), multimedia and text materials were gathered for each POI. We opted for a three-layer textual description accompanied by a gallery of images and an optional gallery of videos.

4.3 System architecture

The architecture of the whole system is presented in Figure 1 with the web page (web server) as the integrating part. Mobile application is installed from the web page or from the application store (Google PlayStore[2] or the Apple store[3]). The mobile application "AS" retrieves new paths from the web server. The paths with the accompanying objects are prepared and monitored from the Heritage Information Catalogue – HIC [7] and all changes are automatically propagated to the web server and the mobile apps.

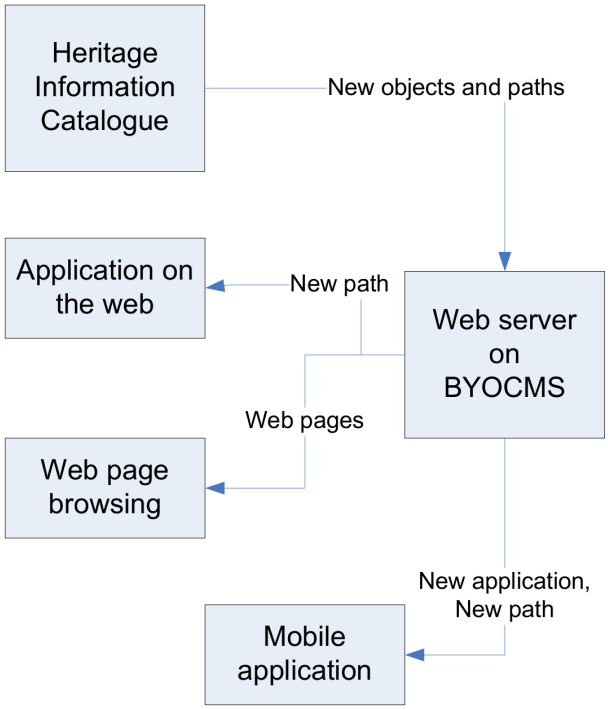


Figure 1: The architecture of the whole system with the web page as the integrating part.

4.4 Interfaces

The communication between system entities relies on an open REST API[8]. The answers to the API requests are in

[2]Google Play store: `https://play.google.com/store`
[3]Apple App store: `https://www.appstore.com/`

JSON[9] format. An example API call with the resulting reply is shown in Figure 2.

```
API request:

https://www.project-as.eu/api/path/get_paths

Reply:
[
{
"name":"Prva izolska pot",
"id":0
},
{
"name":"Druga izolska pot",
"id":1
}
]
```

Figure 2: An example API call with the resulting reply.

The paths with all data describing the archaeological sites (the images and videos are linked as urls linking to the Project AS web page) are encoded in standard GEOJson[10] and can be exchanged with other services. The first implementation of such service is our web page that displays the same data as the mobile applications. Part of such path is presented in Figure 3.

```
{
"type":"FeatureCollection",
"features":[
{
"type":"Feature",
"properties":
{
"name":"Izola 2",
"time":"2016-09-09T12:35:51Z"
},
"geometry":
{
"type":"LineString",
"coordinates":[[13.647916316986,45.533476766486], ...
```

Figure 3: Part of an actual GEOJson encoded path.

4.5 Heritage Information Catalogue

One of the goals of the HIC project was the construction of a publicly accessible multipurpose platform for storage, management and sharing of natural and cultural heritage artifacts. The system was populated with more than 4000 objects. A new interface for path definition and a new set of API calls for path management added to the existing infrastructure for the purposes of the Project AS. An example usage can be observed on Figure 4.

4.6 Web page

The web page is based on a proprietary Content Management System – CMS[11] Build Your Own CMS – BYOCMS[4], developed by the CORA Centre at the University of Primorska. It relies on open source technologies: AngularJS[12] for the single page app web development, CodeIgniter[13] for the backoffice development; MySQL[14] for the storage and Apache[15] as the web server. An example web page is presented in Figure 5.

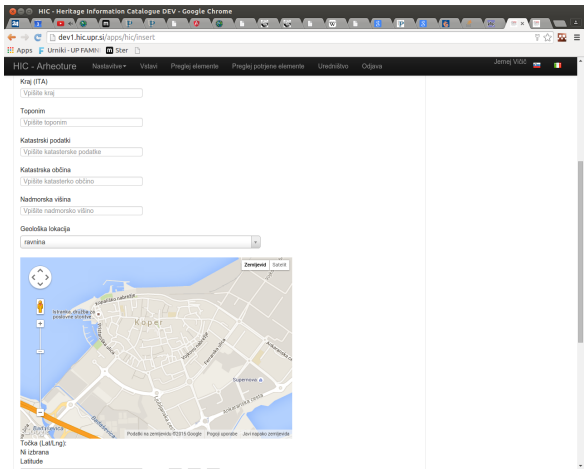


Figure 4: An example usage of the HIC platform for the input of heritage artifacts.

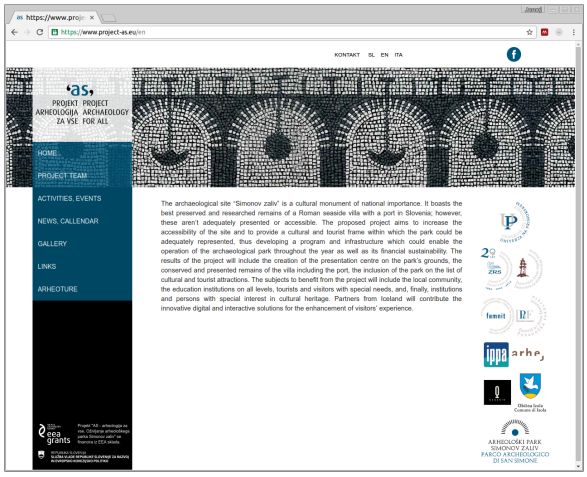


Figure 5: An example web page for the AS project.

.

4.7 Mobile application

Seamlessly download new paths and inform the user of new possibilities. We opted for a non aggressive approach meaning that the paths are downloaded only at the application start and at a user intervention – by clicking of the appropriate button. Scores are synchronized in the event of a path correction, the path statistics is also updated at that event. All technology for mobile application is based on JavaScript language and Node.js library, more specifically on the module npm from Node.js library that allows the

[4]BYOCMS: http://cora.iam.upr.si/en/byocms

developer an easy setup of the initial development environment on any (UNIX) operating system. We prepared the final version of the development environment, based on the Ionic environment [**?**] that is based on libraries Cordova[**?**] and AngularJS. Such an environment enables applications implementation for the two most widespread mobile platforms. The same code was used to set up the application on the website. The code is the same, but the functionality is limited partly because the web application is not intended as an on-field tools. It is designed to offer rapid overview of the offered paths that can be later visited by the mobile application.

4.8 Motivation using persuasive technology

The technology has been used in many tasks to persuade people and motivating them toward various individually and collectively beneficial behaviors. There are two dominant conceptual paradigms: persuasive technology [1] and the more recent gamification[2]. For the purposes of this paper we will use persuasive technologies and rely on studies such as [2] despite these differing titles, the conceptual core of both paradigms incorporates the use of technology that is aimed at affecting people's/users'. The concept of gamification is defined as the use of game design elements and game thinking in a non-gaming context [16]. Motivation is a central topic in gamification as gamified systems are implemented to change behavior for wanted and desirable activities, in our case the user involvement into historic artifacts discovery. Extrinsic motivation focuses on applying game elements into a non-gaming context to stimulate external motivation. Second, game thinking and motivational design has a positive influence on intrinsic motivation as it is done because of an internal desire to play [**?**]. The main goal of the mobile application is to engage the user into active discovery of the architectural heritage of the presented area. Three paths were prepared in the scope of the project till now and a few are still in the progress of development. The user is involved into active discovery through persuasive technology [1]. The user is collecting virtual rewards (treasures) by visiting the predefined sites along the path. The application is keeping score of the found (visited) sites and produces reports about the progress of the journey. The final objective of the application is to collect as many as possible objects from the path resulting in visiting all the sites presented by the path.

5 Results

The platform presented in 4 with the accompanying applications and the multimedia materials were made as one of the actions of the Project AS. The platform was created with the upgradability and updating as one the most important criteria. Three paths with the accompanying objects were prepared at the moment of writing this paper and new objects are already prepared. The system as presented in 4.3 is deployed and being used on a regular basis. The paths with the accompanying objects are prepared and monitored from the "Heritage Information Catalogue". The mobile app will be available for download from the project web page or installed from the Google Play Store and the Apple App Store.

5.1 Typical use case

A typical use case would involve a visitor intrigued by the invitation from the web page, installing the mobile application. The application presents a map that guides the user along the path. The user follows the path on the application from an artifact to the next and earns involvement points. The application also follows basic statistics such as the percentage of the visited path, the percentage of found artifacts along the path. A non-completed path viewed from the mobile application is presented in Figure 6, new entries are being added to the path in the time of writing of this paper.

Figure 6: A non-finished path from mobile application, additional elements will be added later.

6 Discussion

Persuasive technologies and gamification have been used quite frequently in tourism applications the last few years. Massive communities have formed around gamification systems such as Geocaching. A mobile application implementing gamification techniques has been developed following these findings. The application is still in testing

phase, the local testers (project members) were very interested in the usage and happily finished the presented tasks (receiving virtual presents - gaming points). The presented system is already deployed and first test installations were already done. The starting programming and interface errors have been addressed and the application with the underlaying platform will be made available to public in the next months. Three paths have been constructed at the moment and test users have already used the application through the paths (they managed to collect all the virtual treasures). All three tracks are fully accessible, no obstacles have been found on the test runs. The mobile app is being registered to the application stores. The whole framework can be used for new applications, the input system was designed to describe all natural and cultural heritage artifacts, the open APIs and standard interfaces allow easy data sharing, the mobile application can be upgraded with new paths with no need for additional installation.

References

[1] H. Oinas-Kukkonen and M. Harjumaa, "Persuasive systems design: Key issues, process model, and system features," *Communications of the Association for Information Systems*, vol. 24, no. 1, p. 28, 2009.

[2] K. Huotari and J. Hamari, "Defining gamification: a service marketing perspective," in *Proceeding of the 16th International Academic MindTrek Conference*. ACM, 2012, pp. 17–22.

[3] F. Xu, J. Weber, and D. Buhalis, "Gamification in tourism," pp. 525–537, 2013.

[4] J. Weber, "Gaming and Gamification in Tourism," The Digital Tourism Think Tank, Tech. Rep., 2014.

[5] C. Corrêaa and C. Kitanoa, "Gamification in tourism: Analysis of brazil quest game," in *Proceedings of ENTER 2015*, 2015.

[6] L. B. Gram-Hansen, "Geocaching in a persuasive perspective," in *Proceedings of the 4th International Conference on Persuasive Technology*. ACM, 2009, p. 34.

[7] A. BEGUŠ, Ines, HROBAT VIRLOGET, Katja, PANJEK, *Med kamenjem : snovna in nesnovna krajina Krasa.* Pokrajina Trst, 2015.

[8] M. Masse, *REST API design rulebook*. " O'Reilly Media, Inc.", 2011.

[9] D. Crockford, "The application/json media type for javascript object notation (json)," p. 10, 2006.

[10] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and C. Schmidt, "Geojson specification," *Geojson. org*, 2008.

[11] S. Baxter and L. C. Vogt, "Content management system," 2002.

[12] A. Lerner, *Ng-book: The Complete Book on AngularJS*. Fullstack. io, 2013.

[13] T. Myer, *Professional CodeIgniter*. John Wiley & Sons, 2008.

[14] S. Suehring, *MySQL bible*. John Wiley & Sons, Inc., 2002.

[15] R. B. Bloom and B. Foreword By-Behlendorf, *Apache Server 2.0: The Complete Reference*. Osborne/McGraw-Hill, 2002.

[16] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, and D. Dixon, "Gamification. using game-design elements in non-gaming contexts," in *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2011, pp. 2425–2428.

CONTENTS OF *Informatica* Volume 40 (2016) pp. 1–465

Papers

Editorials

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 twentytwo 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 2016 (Volume 40) 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 (Simon Dobrišek)
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 (Stane Pejovnik)
ACM Slovenia (Matjaž Gams)

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 40 Number 4 December 2016 **ISSN 0350-5596**

Informatica

An International Journal of Computing and Informatics

Informatica **40** (2016) Number 4, pp. 375–465