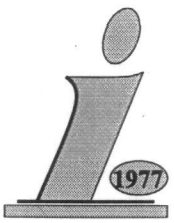
Volume 26 Number 1 May 2002 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics

Informatica **26** (2002) Number 1, pp. 1–101



The Slovene Society Informatika, Ljubljana, Slovenia

Informatica

An International Journal of Computing and Informatics

Archive of abstracts may be accessed at USA: http://, Europe: http://ai.ijs.si/informatica, Asia: http://www.comp.nus.edu.sg/ liuh/Informatica/index.html.

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, Vožarski pot 12, 1000 Ljubljana, Slovenia.
The subscription rate for 2002 (Volume 26) is
– USD 80 for institutions,
– USD 40 for individuals, and
– USD 20 for students
Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

LaTeX Tech. Support: Borut Žnidar, Kranj, Slovenia.
Lectorship: Fergus F. Smith, AMIDAS d.o.o., Cankarjevo nabrežje 11, Ljubljana, Slovenia.
Printed by Biro M, d.o.o., Šmartinska 130, 1000 Ljubljana, Slovenia.

Orders for subscription may be placed by telephone or fax using any major credit card. Please call Mr. R. Murn, Jožef Stefan Institute: Tel (+386) 1 4773 900, Fax (+386) 1 219 385, or send checks or VISA card number or use the bank account number 900–27620–5159/4 Nova Ljubljanska Banka d.d. Slovenia (LB 50101-678-51841 for domestic subscribers only).

Informatica is published in cooperation with the following societies (and contact persons):
Robotics Society of Slovenia (Jadran Lenarčič)
Slovene Society for Pattern Recognition (Franjo Pernuš)
Slovenian Artificial Intelligence Society; Cognitive Science Society (Matjaž Gams)
Slovenian Society of Mathematicians, Physicists and Astronomers (Bojan Mohar)
Automatic Control Society of Slovenia (Borut Zupančič)
Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Igor Grabec)

Informatica is surveyed by: AI and Robotic Abstracts, AI References, ACM Computing Surveys, ACM Digital Library, Applied Science & Techn. Index, COMPENDEX*PLUS, Computer ASAP, Computer Literature Index, Cur. Cont. & Comp. & Math. Sear., Current Mathematical Publications, Cybernetica Newsletter, DBLP Computer Science Bibliography, Engineering Index, INSPEC, Linguistics and Language Behaviour Abstracts, Mathematical Reviews, MathSci, Sociological Abstracts, Uncover, Zentralblatt für Mathematik

The issuing of the Informatica journal is financially supported by the Ministry for Science and Technology, Slovenska 50, 1000 Ljubljana, Slovenia.

Post tax payed at post 1102 Ljubljana. Slovenia taxe Percue.

Some approaches to information security of communication networks

Sergey Avdoshin and Victor Serdiouk
"MATI" – K.E. Tsiolkovsky Russian State Technology University,
Orshanskaya 3, 121552, Moscow, Russia
Phone: +8 095 9150196, Fax: +8 095 9150196
E-mail: avdoshin@mati.msk.su and vicsmati@online.ru

Keywords: network information security, informational attacks

Received: October 10, 2001

A classification of the attacks on the communication network information sphere is conducted. A new concept of communication network information security systems (ISS) construction is developed. In contrast to the existing security methods and tools, the new concept provides the integration of such functions as detection and elimination of network vulnerabilities, detection and prevention of attacks and functions of detection and elimination of consequences of undetected or unprevented attacks. The architectural principles of the network ISS construction, according to the proposed concept, are based on the use of integrated security blocks and security management centre. A description of the ISS prototype constructed according to the new concept and called "Shield" is given.

1 Introduction

At present there is a wide variety of tools for the communication networks protection against information attacks of the intruders. Among these tools are firewalls (Hunt 1998; Hare & K.Silyab 1996), intrusion detection systems (Ranum 1999; Serdiouk 2000), security scanners (Freiss 1999), tools for virtual private networks construction, etc. Nevertheless, it is necessary to state that to date they don't cope with the tasks of the minimization of damage caused by the realization of various information security threats. This fact is confirmed by the research results of USA's Computer Security Institute (Richard Power 2001) that show the steady growth of annual attack losses. The underlying reason for the current situation is, from the authors' point of view, the absence of integrated network information security tools that could simultaneously perform functions of detection and elimination of network vulnerabilities, functions of detection and prevention of informational attacks on the network and functions of detection and elimination of attack consequences.

Therefore the authors of this article have set a task to develop a new concept of information security systems (ISS) construction that could protect public and private communication networks against informational attacks. At first, in order to give a general idea about the research purposes it is necessary to cite the characteristics of types, structures and objects of communications networks for which a new concept of ISS construction was developed.

2 Communication network types and security objects

The *communication network* is a set of technologically connected telecommunication networks that is opened for individual and juridical persons (Tanenbaum 1996; Fred Halsall 1996). Communication networks can be private and public. There are several types of communication networks:

- telephone communication network;
- telegraph communication network;
- integrated services digital communication network;
- data communication network with circuit and packet switching;
- mobile communication network;
- satellite communication network.

It is worth to note that all mentioned communication networks use different physical mediums (cable, optical, satellite, etc.). It allows transmitting telephone and telegraph information, data packets and other information with different methods of switching.

Each communication network has its own management subsystem, which is a technical-organizational structure that provides management process functions and functions of information transmission servicing on the network. The communication network management center (NMC) acts as a control element that forms and distributes control commands among communication network nodes (routers, switches, multiplexers, digital exchanges, etc.). Control commands are distributed by means of network packets that contain control information. The NMC is managed by the network operator.

The capabilities of the ISS construction concept, developed by the authors, can be better shown on data communication networks based on TCP/IP protocol

suite, because it is the most widespread type of networks in most countries. At first, we describe communication network structure, its main elements and objects that must be secured by the ISS. The communication network base elements are cited below:

- communication network nodes;
- access communication links;
- network backbone links;
- NMC;
- users workstations connected to the communication network.

The communication network structure is depicted in Figure 1.

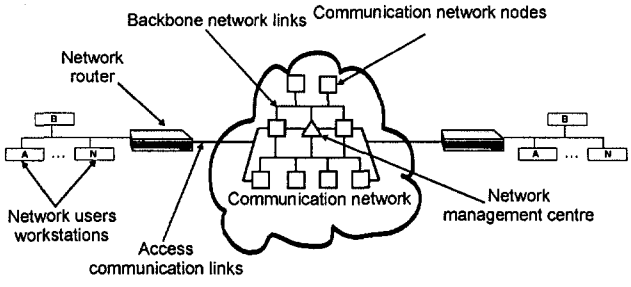


Figure 1: The structure of a communication network.

Usually the following network components are more frequently the subject for the attacks and therefore must be secured:

- data packets that are transmitted through the communication network and contain user information;
- data packets that are transmitted on the communication network and contain control information;
- the network nodes software (e.g. the node operating subsystem, the auditing subsystem, etc.);
- the information stored in network nodes (e.g. routing tables, the contents of the node audit log, management information base, etc.);
- the NMC software (e.g. the NMC operating system, the subsystem of network nodes management, etc.);
- the information stored in NMC (the contents of NMC audit log, the database that contains the network nodes management parameters, etc.);
- the users workstation software;
- the information stored in users workstations.

The collection of network components listed above will be denoted as *communication network information sphere*. The *communication network object* term involves such network elements as NMC, network node and user workstation. The collection of software and information, stored in network object will be considered as network object information sphere.

After describing communication network components and elements that must be secured it is necessary to consider network information security threats.

3 Communication network information security threats

An *attack* (or intrusion) on network information sphere is a set of intruder actions aimed at the realization of communication network information security threat. Here the *information security threat* is a possible consequence of the attack that may lead to the violation of network functioning process and therefore cause damage to network users or network operator.

The intruder can be foreign political, economic or military structures which purposes include the fulfillment of unauthorized actions aimed at the achievement of economic and political objectives. Personal goals usually have independent hackers and disgruntled employees which attack communication networks in order to cause damage or to acquire a new qualification status.

The realization of attack on network information sphere is based on the exploitation of network *vulnerabilities* – the shortcomings or weaknesses of network information security measures that may lead to the successful realization of network information security threat. There are two types of network vulnerabilities: technological and operational vulnerabilities. The technological vulnerabilities result from the intentional or unpremeditated error made on the stage of network design or implementation. The operational vulnerabilities result from the error made on the stage of network configuration or using.

The communication network information security threats can be categorized according to three main network information sphere security parameters – the confidentiality, integrity and availability. A description of these parameters in more details is given below.

The confidentiality of the communication network information sphere is a capability of the network to protect its information sphere against the unauthorized access to it.

The integrity of the communication network information sphere is a capability of the network to protect its information sphere against unauthorized modification.

The availability of the communication network information sphere is a capability of the network to provide timely and unhindered access of legal users to its services.

According to these security characteristics, all attacks can be divided into several groups and subgroups, depending on the types of threats for network information sphere integrity, confidentiality, and availability.

A) The violation of communication network information sphere integrity

A.1) The violation of integrity of data packets that contain network users information transmitted through the network:
- unauthorized modification of network user information;
- unauthorized iteration of data packets that have already been transmitted through the network;
- unauthorized modification of data packets headers.

A.2) The violation of integrity of data packets that contain control information, transmitted on the network:
- unauthorized modification of control information;
- unauthorized iteration of data packets that have already been transmitted through the network;
- unauthorized modification of data packets headers.

A.3) The violation of network nodes information sphere integrity:
- unauthorized modification of network node software;
- unauthorized removal of network node software;
- unauthorized modification of information stored in network nodes;
- unauthorized removal of information stored in network nodes.

A.4) The violation of NMC information sphere integrity:
- unauthorized modification of NMC software;
- unauthorized removal of NMC software;
- unauthorized modification of information stored in NMC;
- unauthorized removal of information stored in NMC.

A.5) The violation of users workstations information sphere integrity:
- unauthorized modification of users workstations software;
- unauthorized removal of users workstations software;
- unauthorized modification of information stored in users workstations;
- unauthorized removal of information stored in users workstations.

B) The violation of communication network information sphere confidentiality

B.1) The violation of confidentiality of data packets that contain network users information transmitted through the network:
- unauthorized access to network user information;
- unauthorized access to user information transmission parameters (source and destination addresses, type of the protocol which was used for information transmission, etc.).

B.2) The violation of confidentiality of data packets that contain control information transmitted on the network:
- unauthorized access to control information.

B.3) The violation of network node information sphere confidentiality:
- unauthorized access to information stored in network nodes;
- unauthorized access to network nodes software.

B.4) The violation of NMC information sphere confidentiality:
- unauthorized access to information stored in NMC;
- unauthorized access to NMC software.

B.5) The violation of users workstation information sphere confidentiality:
- unauthorized access to information stored in users workstations;
- unauthorized access to users workstations software.

C) The violation of communication network information sphere availability

C.1) The violation of network users information availability:
- unauthorized deletion of data packets containing user information;
- unauthorized delay of data packets containing user information.

C.2) The violation of control information availability:
- unauthorized deletion of data packets containing control information;
- unauthorized delay of data packets containing control information.

C.3) The violation of network nodes information sphere availability:
- the violation of network node operational capability;
- the blocking of access to the network node.

C.4) The violation of NMC information sphere availability:
- the violation of NMC operational capability;
- the blocking of access to the NMC.

C.5) The violation of users workstations information sphere availability:
- the violation of users workstations operational capability;
- the blocking of access to the users workstations.

C.6) The deterioration of data transmission process qualitative characteristics:
- the reduction of data rate;
- the increase of data packet transmission time;
- the reduction of data channels throughput;
- the increase of data loss ratio.

C.7) The unauthorized access to the communication network services.

It is necessary to note that all network information security threats that were mentioned above can be realized by an intruder in two ways: from the inside and from the outside of the communication network. In the first case the attacker is located beyond the bounds of communication network, e.g. in LAN connected to the communication network. In the case of threat realization from the inside of communication network, the attacker is located within the network itself.

After describing communication network information security threats, let's consider the ISS construction concept developed by the authors.

4 Conceptual approaches to the construction of communication network ISS

According to the concept that was developed by the authors, the communication network ISS must perform the following functional tasks:

- the detection of technological and operational network vulnerabilities;
- the warning security administrator of the detected vulnerabilities;
- the elimination of vulnerabilities detected in network hardware and software;
- the detection of attacks on the network information sphere that can cause the violation of confidentiality, integrity, and availability of network information sphere (see section 3);
- the warning network security administrator of detected attacks;
- the prevention of attacks on the network information sphere;
- the detection of consequences occurred as a result of unprevented attacks;
- warning security administrator of detected attack consequences;
- the elimination of unprevented attack consequences.

The ISS functional tasks model is shown in the Figure 2.

The information security functions mentioned above underline the architectural principles of network ISS construction. According to this concept, the network ISS consists of two main components: integrated security blocks (ISB), that protect network information sphere against informational attacks and security management centre (SMC) that controls and manages the ISB functions. Let's consider the functionality of network ISS components and their interoperation mechanisms in more details.

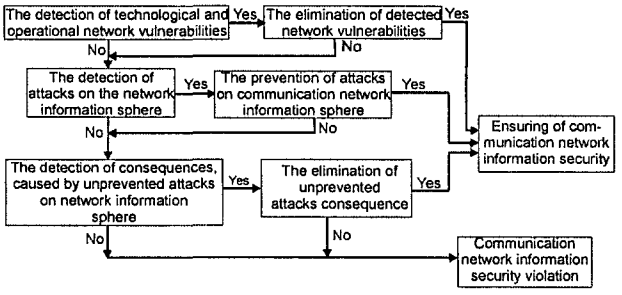


Figure 2: The ISS functional tasks model

4.1 Integrated security blocks

Network ISS ISB are the set of security tools integrated into unified software-hardware system. The ISB elements composition is selected in such a way that it could ensure the fulfillment of network information security functions, formulated in section 4. For these purposes ISB can be installed at two different locations (see Figure 3):

- outside the communication network, i.e. installed into data links, that are used for user and LAN connection to the communication network;
- inside the communication network, among the network nodes.

The SMC is installed inside the communication network and performs ISB monitoring and control functions.

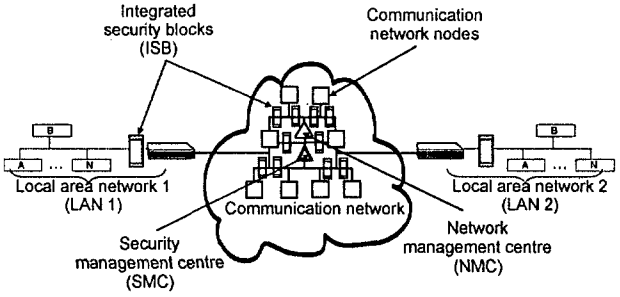


Figure 3: The installation of integrated security blocks and security control center

Depending on the installation type, ISB can perform different tasks. If ISB is installed outside the communication network it will perform the following functions:

- detection and elimination of vulnerabilities of user workstation software and hardware;
- LAN and network users protection against attacks whose source is located inside the communication network;

- LAN and network users protection against attacks whose source is located inside the LANs;
- network nodes and NMC protection against attacks whose source is located inside the LAN;
- confidentiality security of data packets that contain network user information;
- integrity security of data packets that contain network user information.

ISB which are installed inside the communication network perform other functional tasks:

- detection and elimination of vulnerabilities of network nodes and NMC software or hardware;
- network nodes and NMC protection against attacks whose source is located inside the communication network;
- confidentiality security of data packets that contains control information;
- integrity security of data packets that contain control information;
- integrity security of data packets that contain network user information.

It is necessary to note that the ISB components structure and composition is invariable and doesn't depend on the installation type of the ISB. Further, let's show the structure and the composition of ISB on the concrete information security functions examples.

4.1.1 The function of vulnerabilities detection and elimination

The detection of technological and operational vulnerabilities is implemented by means of inclusion of *security scanner* as a component of the ISB. There are two types of scanners: passive and active. Passive scanners collect information about hardware and software used on the network (e.g. the version of programs, types of operating systems (OS), etc.) and on the base of gathered information define the network vulnerabilities. Active scanners model the attacks on the network information sphere and, according to the attack results, scanners define the list of operational and technological vulnerabilities. After the detection of vulnerability the scanner offers recommendations for security administrator how he/she can eliminate the detected vulnerabilities. Usually detected vulnerabilities can be eliminated by means of new software modules installation or by means of software/hardware parameters reconfiguration.

Due to the capability of real informational attack modelling active scanners can obtain more precise information about network vulnerabilities.

4.1.2 The function of confidentiality and integrity security of data packets that contains control or network user information

This function is realized by means of inclusion of *data packets confidentiality and integrity security module* as a component of the ISB. This module is based on the virtual private networks construction tools and operates in the following way. When the data packet that must be secured arrives at the ISB input, the module creates so-called "tunnel", or virtual connection and transmits through it packets which contain user or control information. The tunneling technology operates in the following way. Before the forwarding procedure, the packets are processed by the cryptographic security tools, whereupon they are encapsulated in new packets that are used for the transmission through the tunnel. Information confidentiality is secured by means of its encryption and integrity – by means of digital signature mechanism. After the transmission of data packets they are extracted on the other side of the tunnel, decrypted, integrity checked, whereupon the information is transmitted to the recipients. The general tunnel scheme between two ISB is depicted in Figure 4. In practice the tunneling technology can be realized by means of such protocols as L2TP (Layer 2 Transport Protocol) or IPSec (IP Security).

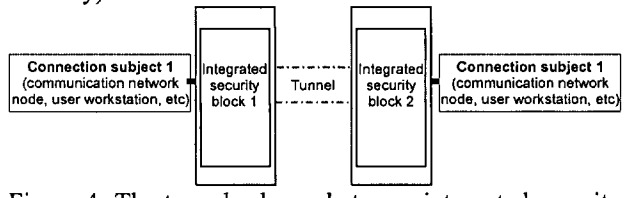


Figure 4: The tunnel scheme between integrated security blocks

The module of data packets integrity and confidentiality security must register the following information about the module work in ISB audit log:

- the parameters of tunnels created in communication networks;
- the time of tunnel creation;
- the time of tunnel closing;
- the type of used cryptographic operation;
- the result of cryptographic operation application.

With the help of ISB installed inside the communication network, it is possible to check the data packets integrity during their transmission through the network. When the encrypted data packet with message authentication code of the digital signature arrives at the ISB input, the module of data packets integrity and confidentiality security anew computes the authentication code and compares it with the value that is contained in received packet. The match of these values means that the packet integrity wasn't violated and it can be forwarded to the next network node or to the recipient. Otherwise the module registers in ISB audit log information about the

packet, deletes it from the network traffic and warns the security administrator of the data packet integrity violation.

4.1.3 The function of communication network objects confidentiality, integrity and availability security

The confidentiality, integrity and availability of communication network object (network nodes, NMC and user workstations) information sphere can be secured by inclusion of two subsystems as components of ISB: filtering subsystem and attack detection subsystem. Let's consider each of these systems in more details.

Filtering subsystem is based on the firewall functional capabilities, that allow to define conditions of packets transmission. The main firewall function is filtering of information flow that passes through it. During the traffic filtration process firewall scans and analyzes every packet that passes through it. The firewall forwards the packet only if it corresponds to the filtering criteria, set by security administrator. Firewalls can implement filtering mechanisms on different OSI levels.

The inclusion of filtering subsystem as an ISB component allows to use the mechanism of search and deletion of packets that threaten information security of network nodes, NMC and user workstations. Thereby the filtering subsystem prevents the realization of intruders' attack. For example, the filtering subsystem must filter out those packets that contain old and vulnerable protocols, packets that have deviation from existing standards, etc. However, it is necessary to note that the use of network traffic filtering function in ISB can protect communication network only against several types of attacks that can be prevented by means of rules of packet deletion from network traffic. Therefore, for protection against more complex attacks it is necessary to include an additional component to the ISB structure – attack detection subsystem. Let's consider operation algorithms of attack detection subsystem in more details.

Attack detection subsystem performs two main functions: the forming of a data bank that contains information about communication network operation and the detection of attacks on communication network information sphere on the base of the created data bank.

The data bank can include the following information:

- the information about types and parameters of communication network protocols;
- the information about the addresses of network objects which sent information through the network;
- the information about the volume of transmitted data;
- the information about the number of network connections created during the unit of time;
- the information about network objects load, etc.

After the identification of attack the detection subsystem reports about the incident to the SMC. The data bank forming function is implemented by means of interception of packets that passes through the ISB. After the packet interception the detection subsystem extracts from the data packets headers and writes to the data bank the following information:

- the date and time of data packet receipt;
- the length of data packet header;
- the length of packet data field;
- the IP-address of data packet recipient;
- the IP-address of data packet sender;
- the port number of packet sender and receiver;
- the type of the protocol;
- additional information about protocols (e.g. acknowledgement and sequence protocol numbers, etc.).

After the registration of this information in the data bank, the detection subsystem starts analyzing it and tries to establish the fact of attacks realization. The detection subsystem can use two types of methods of information analysis: the analysis method based on communication network profiles and the analysis method based on attack signatures (Bace 2000; Helmer & Wong 2001). Let's discuss each of these methods in more details.

The first method is based on the use of communication network profiles, which are a set of such basic characteristics as packets routes, the time of packet receipt and the number of transmitted packets. The communication network profile can be set by security administrator in the manual or automatic mode. If the network profile is formed in automated mode, the detection subsystem continuously collects and processes the network operation parameters during a certain period of time. At the end of forming process, the profile will reflect normal network operation mode. If any of the current network operation parameters doesn't correspond to profile parameters it will mean that the attack has taken place. To define the mismatch between current network operation characteristics and the parameters defined in the profile the mismatch index is used. The mismatch index is calculated by means of comparison of profile parameters and current network operation parameters. If the computed mismatch index exceeds the threshold defined by the security administrator, it will mean the attack realization. The security administrator must be immediately informed about this attack.

As an example, let's consider a communication network segment which contains a network node that must be secured. To detect attack on the network node information sphere, ISB must be installed in all data links of the node. It is necessary to remember, that detection subsystem of ISB must contain network node profile, which includes such parameters as types of protocols that uses network node, the maximum and the minimum number of packets that can be processed by the node, qualitative characteristics of data transmission on communication network segment where the node is installed, addresses of other nodes that can exchange information with the secured node and other parameters. If the current network node operation parameters are out

of profile tolerance range of values (e.g. the emergence of a new network protocol, the exceeding of the maximum packets number that can be processed by the node, the appearance of a new node with unknown address, and which also interacts with the security object, the absence of network node that must interact with the security object, the deterioration of data transmission process qualitative characteristics, etc.), then the detection subsystem informs security administrator and the SMC about detected attack. At the same time different network nodes installed at various network segments have different profiles. It is also necessary to note that detection method based on profiles can detect both the attack and its consequences. The deterioration of data transmission process qualitative characteristics is the example of the attack consequence. Another attack consequence example is the lengthy absence of a certain network node packets that can be caused by the node operation capability violation.

The second information analysis method is based on the detection of attacks by means of special templates or signatures. The attack signature is a set of characteristics of data packets transmitted through the communication network that helps to detect the violation of network information security. Attack signatures are formed on base of information about possible communication network vulnerabilities. The examples of the attack signatures are cited below:

- the templates of incorrectly formed data packets (or packets which format doesn't correspond to Internet's RFC), that can violate the operation capability of a network node or a workstation – the receiver of these packets;
- the templates of data packets or packets sequences that exploit vulnerability of the workstation or node software (such vulnerability may be an error made by the programmers, in the software);
- the templates of data packets or packets sequences which contain information that can violate the data transmission process or operation capability of network node or workstation if it will be included in node or workstation information sphere.

All attack signatures are stored in the detection subsystem database. When using this method, the subsystem simply compares the contents of data bank containing the information about transmitted data packets with the attack signature database. If the subsystem finds the match between packets structure and the attack signature, then it will mean the attempt of computer attack realization. It is necessary to note that different types of network architectures require different signature databases. The attack signature database is supported by the security administrator.

The examples of detection subsystem operation method based on attack signatures are cited below. Let's assume that an intruder has chosen a communication network node as an attack object and wants to violate the operation capability of this node by means of "Land" attack (Serdiouk 2000). The point of "Land" attack is in sending a special TCP-segment to network node, where sender IP-address coincides with the recipient IP-address and the recipient service port number equals to the sender service port number. The sender can coincide his/her IP-address with the recipient's IP-address by means of low-level data packet formation using specialized software such as packet drivers that can directly work with network adapter functions. Another way of sender IP-address falsification is to modify OS parameters that store the real IP-address of the sender. Usually such parameters are stored at the OS registry. After the receipt of "Land" attack packet the node tries to redirect it to itself and it results in complete violation of node operation capability. The number of packets required for violation of node operation' vulnerability depends on the operation OS type and vary from one to one thousand. For the attack realization the intruder needs to forge the sender IP-address and port number. The example of such TCP-segment and IP-datagram is cited below (Comer 1995).

4 (IP version)	Header length	0 (Type of service)	Total length
1234 (Identificator)		0 (Flags)	0 (Offset)
29	06	The checksum	
195.164.125.16 (Recipient IP-address)			
195.164.125.16 (Sender IP-address)			
0 (Options)		0 (Padding)	

Figure 5: The format of "Land" IP-datagram

80 (Recipient port number)		80 (Sender port number)	
4136 (Sequence number)			
0 (Acknowledgement number)			
0 (Offset)	0 (Reserved)	2 (Flags)	20 (Window size)
The checksum		26	
0 (Parameters)		0 (Padding)	

Figure 6: The format of "Land" TCP-segment

The cited TCP-segment is sent to the communication network node to the IP-address - 195.164.125.16, to 80th port. To identify this type of attack the detection subsystem must contain the following "Land" signature: "If (<the IP-address of data packet sender> equals to <the IP-address of data packet receiver>) and (<the receiver service port number> equals to <the sender service port number>) then the data packet is a part of `Land` attack". The signatures of other attacks are defined in similar way. Attack signatures can be defined by means of syntax of any programming language such as Pascal or C. It is also necessary to note that the signature-based method can detect the both attacks and its consequences.

After the detection of the attack and the report about it to the SMC ISB must proceed to the attack prevention mechanism. This mechanism is implemented by means of interaction of detection and filtering subsystems. Let's

describe the interaction process in more details. If the detection subsystem identifies the attack that is implemented by intercepted packet (which may be a part of a packet sequence), it will send the command of deleting this packet from the information stream to the filtering subsystem. If the intercepted data packet contents doesn't threatens to the communication network information security, it will be forwarded further to the recipient. Thereby, by means of filtering packets that threaten to the network security the ISB prevents the attack realization by the intruder. The scheme of the attack prevention algorithm is shown in Figure 7.

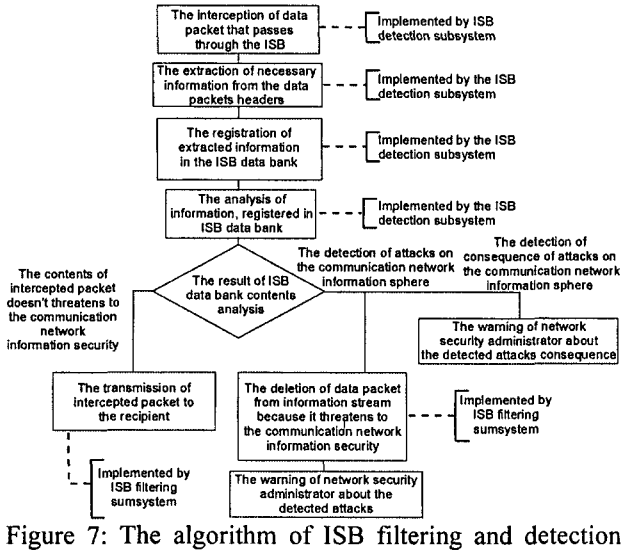


Figure 7: The algorithm of ISB filtering and detection subsystems operation

The example of the detection and prevention of the attack by means of interaction of two ISB subsystems is cited below. Let's assume that the detection subsystem has identified the distributed denial of service attack "Ping Flooding" (Harris & Hunt 1999; Ferguson & Senie 1998). The purpose of this attack is in sending to the recipient a large number of ICMP-echo-requests on behalf of non-existent computers. After the receipt of ICMP-echo-requests, the attacked node processes them and sends ICMP-echo-reply. Since the sources of these requests are non-existent nodes, the result of this attack will be the decrease of attacked node performance and the deterioration of communication network link throughput. After the identification of this attack the detection subsystem must report to the SMC about the detected attack. After that ISB automatically reconfigures the filtering subsystem. In other words it defines new rules of filtering of those data packet that contain IP-addresses of attack sources or rules of filtering of packets, destined for the network critical information resources. After the filtering subsystem reconfiguration the attacker packets stream will be decreased and the probability of successful realization of information security threat will also be minimized.

Therefore the inclusion of the filtering and detection subsystems as components of ISB allows not only to detect the violation of confidentiality, integrity and the availability of communication network objects, but also to prevent the attack.

The filtering and detection subsystems must register in ISB audit log the following information:

- the type and information about the detected attack;
- the time of attack detection;
- the method by which the attack was detected;
- the information about packets, passed through the ISB;
- the result of data packet processing (filtered / not filtered);
- the time when the filter was applied.

The summarized structure of ISB is depicted in Figure 8.

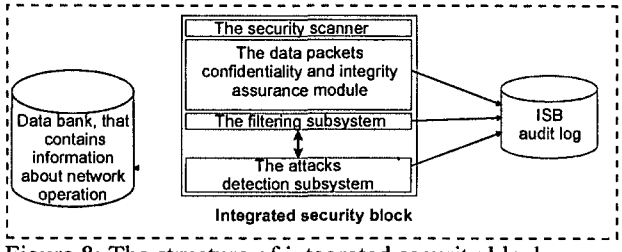


Figure 8: The structure of integrated security block

After the description of ISB functional capabilities we would like to consider tasks that must perform the SMC.

4.2 Security management centre

The SMC of communication network ISS performs the following tasks:

- the management of ISB cryptographic keys;
- ISB operation monitoring;
- the controlling of data transmission process qualitative characteristics;
- the attack source localization;
- the warning of security administrator about the detected attack and their consequences;
- the elimination of consequences of unprevented attacks on the communication network information sphere;

Let's consider SMC functions in more details.

4.2.1 The functions of ISB cryptographic keys management and ISB operation monitoring

All ISB operation control and management functions are implemented by means of "agent – manager" model, which is based on the simple network management protocol (SNMP). Within this model the agent is a program module, installed on the managed resource – ISB, and the manager is an operating control of this resource – SMC. The management process is implemented by means of analysis and modification of

parameters values stored at the management information base of ISB.

The ISB cryptographic keys management function includes the generation and distribution of new keys for the ISB module of data packets confidentiality and integrity security. The ISB operation monitoring function includes a periodic read and analysis of ISB audit log. Both these functions are implemented by means of SNMP commands.

4.2.2 The function of security of data transmission process qualitative characteristics control in the communication network

The security of data transmission process qualitative characteristics control includes the control of data transmission rate, control of data links throughput, control of data loss ration, etc. The control of these characteristics can be provided by SMC by means of analysis of data registered in ISB audit log in different communication network segments. For example, if the difference between the time of packet arrival to the network and the time of packet exit from the network exceeds the certain threshold value it will mean the deterioration of data transmission process qualitative characteristic. The security administrator must be immediately informed about this deterioration.

4.2.3 The attack source localization

If one of ISB detects the violation of communication information security it must send all information about it to the SMC. On the base of this information the SMC starts the attack localization procedure. Usually the attack source is localized by means of IP-addresses of packets received by attacked object. However, taking into the consideration the ability of the attacker to forge the IP-addresses, the attack can be localized by mean of tracing of packet route on the base of the ISB audit log contents.

4.2.4 The function of unprevented attack consequences elimination

The elimination of unprevented attack consequences is implemented by SMC by means of recommendations packet forming for the network security administrator. This packet contains the list of possible actions that security administrator has to take for the restoration of network functioning process. The recommendations can be fulfilled by the network security administrator in conjunction with the NMC operator. Usually the procedure of network functioning process restoration is realized be means of backup data transmission routes, backup data communication channels, backup data storages and network nodes.

4.2.5 The function of network security administrator warning of the detected attacks and their consequences

In the case of the detection of the attack or its consequence, the SMC must inform about it the network security administrator. The signaling about attack or its consequence can be fulfilled by means of e-mail, paging, and facsimile communication or by displaying the corresponding information on the network SMC console.

5 The practical realization of the communication network information security system construction concept

On the base of the developed network ISS construction concept, and within the research conducted at "Information Technologies" department of "MATI" – K.E. Tsiolkovsky Russian State Technology University a system prototype was developed and named "Shield". ISS "Shield" includes the following components: ISB and SMC (their structure was discussed in section 3). ISB of the network ISS "Shield" is based on the security scanner and the module of attacks detection and prevention, which in turn consists of filtering and detection subsystems. The developed ISS is designed for the use in network based on the TCP/IP suite. As a main instrument of ISS "Shield" creating a rapid application development environment, Borland C++ Builder was chosen. Such environment allows to use automated user interface design tools and interact with the Windows API functions directly. For working with the low-level network adapter functions, a packet driver was used. The packet driver operates on the data link layer of the OSI model. The testing of ISS prototype "Shield" was fulfilled at heterogeneous network of MATI "Information Technologies" department.

The installation and maintenance procedure of ISS "Shield" must be fulfilled only under the control of the communication network security administrator. The computer, where ISS "Shield" is installed must operate under the Windows OS (95/98/ME/NT2000). Before the installation to the network node, the procedure of its security level assessment must be conducted. For this purpose a security scanner of ISS "Shield" can be used. This scanner uses active method of vulnerabilities detection. Only if the network communication doesn't have vulnerabilities, the security administrator can proceed to the installation procedure. According to the developed network ISS construction concept, the ISS "Shield" ISB can be installed at the external border of LAN or inside the global communication network, inside the network nodes.

After the completion of the ISS "Shield" installation procedure, the security administrator must setup the ISB and SMC parameters. Let's discuss the setting procedure in more details.

The ISB setting includes the setup of the following attack detection and prevention module parameters:

- tests parameters for vulnerabilities detection;
- attack signatures;
- communication network operation profile;
- rules of filtering of traffic that passes through the ISB;
- rules of registration of information in audit log;
- rules of reporting about detected attacks to the SMC.

The SMC operation parameters include:

- the rules of gathering of ISB audit log contents;
- the cryptographic keys management parameters;
- the parameters of data transmission process qualitative characteristics control;
- the rules of security administrator warning of detected attack.

After the completion of ISS "Shield" configuration the security administrator has to check the correctness of its setting with the help of security scanner. After the configuration of parameters, considered above, the ISS "Shield" can be set in operation.

6 Conclusion

In this paper an original concept of integrated network ISS construction was described. The functional capabilities of ISS that operates on TCP/IP-networks was shown. The developed concept allows to create security systems that perform the following functions: the detection and elimination of vulnerabilities, the detection and prevention of attacks of the intruders and the detection and elimination of the consequences of unprevented attacks. The authors hope that the use of the developed concept during the network ISS design will allow to solve the problem of communication network information security.

References

[1] Andrew Tanenbaum (1996) *Computer Networks*, Prentice Hall.

[2] B. Harris, R. Hunt (1999) TCP/IP security threats and attack methods, *Computer Communications*, № 22, 885-897.

[3] C.Hare, K.Siyan (1996) *Internet Firewalls and Network Security*, New Riders Publishing, Indianopolis.

[4] Duglas E. Comer (1995) *Internetworking with TCP/IP: Principles, Protocols and Architecture*, Prentice Hall.

[5] Fred Halsall (1996) *Data Communications, Computer Networks and Open Systems*, Addison-Wesley.

[6] Guy Helmer, Johnny Wong (2001) Anomalous intrusion detection system for hostile Java applets, *The Journal of Systems and Software*, № 55, 273-286.

[7] Marcus Ranum (1999) Intrusion detection systems: expectations, ideals and realities, *Computer Security Journal*, vol. 14, № 4, 25-45.

[8] Martin Freiss (1999) *Protecting Networks with SATAN*, O'Reily&Associates.

[9] P. Ferguson, D. Senie (1998) Defeating Denial of Service Attacks which Employ IP Source Address Spoofing, *RFC 2267.*

[10] R. Hunt (1998) Internet/Intranet firewall security-policy, architecture and transaction services, *Computer Communication*, №21 (13), 1107-1123.

[11] Rebecca Gurley Bace (2000) *Intrusion Detection*, Macmillan Technical Publishing.

[12] Richard Power (2001) Issues and Trends: 2001 CSI/FBI Computer Crime and Security Survey, *Computer Security Journal*, vol. 1, № 1, 1-18.

[13] V. Serdiouk (2000) Intrusion detection systems and their role on network security, *BYTE/Russia*, St. Peterburg, Russia, №10, 28-31.

[14] V. Serdiouk (2000) The vulnerabilities of information networks, based on TCP/IP suite, *Telecommunication Security Systems*, № 33, Moscow, Russia, 77-81.

An improvement of a technique for color quantization using reduction of color space dimensionality

Kuo-Lung Hung and Chin-Chen Chang
Department of Computer Science and Information Engineering
National Chung Cheng University, Chaiyi, Taiwan 621, R.O.C.
E-mail:{ jackhong,ccc}@cs.ccu.edu.tw

Keywords: color image quantization, pixel mapping, principal component analysis

Received: March 16, 2002

Color quantization is essential due to the limitations of image displays, data storage, and data transmission. The process of color image quantization can basically be divided into two major steps: color palette design and pixel mapping. Many algorithms have been proposed for the design of the color palette and for pixel mapping. Among them, PMRC is a fast pixel mapping algorithm, which uses the concept of reduction of color space dimensionality [1]. Experimental results have shown that PMRC is much faster than the traditional exhaustive search method. However, there is room for improvement. In this paper, a new pixel mapping method for color quantization using principal component analysis is proposed. Our proposed method first uses the first principal component axis to replace the projection line in PMRC. Next, a new projection value search algorithm is used to solve the shortcomings of PMRC. Finally, a three-dimensional projection filter method is employed to further accelerate the search speed. The experimental results show that the execution speed of our proposed method is much faster than that of the exhaustive search method. Moreover, our proposed algorithm achieves the time complexity of O(NK), which is superior to the O(NKlogK) of PMRC. Our proposed method is therefore a very efficient method for pixel mapping in color quantization techniques.

1 Introduction

Color quantization of an image is a process that uses a small number of colors to represent the image. The objective is to approximate as closely as possible the original full-color images. Color quantization is essential due to the limitations of image displays, data storage, and data transmission. Although many modern systems can display full colors ($=2^{24}$), color quantization is still practical for systems running animation and those used for advanced graphics applications. Moreover, since the color quantization of an image reduces storage space, it also can save image transmission time over networks.

The process of color image quantization can basically be divided into two major steps: color palette design and mapping of pixels in the original image to colors in the designed palette. Many algorithms [1,3-11] have been proposed for the design of the color palette. Linde et al. proposed the LBG method [6]. Given an initial palette, the LBG method minimizes image quantization errors by repeatedly assigning image pixels to the colors closest to them in the palette, and then updates palette colors using the average colors of pixels assigned to them. The LBG method usually obtains good color palettes when initial palettes are carefully chosen. However, when a poor initial palette is used, the generated palette might also be poor. In order to further reduce quantization errors, Tasdizen et al. proposed a genetic method for use in YUV color space [7]. A common goal of the previously mentioned methods is minimizing predefined distortion measures. Unfortunately, this kind of approach usually results in the problem of heavy computation load, which makes these methods impractical for real-time quantization of colors.

Many methods use top-down approaches to accomplish real-time quantization. The median cut algorithm proposed by Heckbert uses a splitting technique [3,5] to repeatedly divide the color histogram into smaller and smaller groups which contain approximately equal numbers of color occurrences, and picks the median values of the groups as the representative palette colors. Wu and Witten proposed the mean-spilt method [8], which uses the mean rather than the median of the projected distribution as the partition distribution. Wan et al. proposed the variance-based method [10] which differs from the median-cut and mean-spilt methods in that the sum of the square errors in the projected distribution in one of the three color-component axes is repeatedly minimized. Therefore, distortions of quantized images obtained using this method are usually low.

When a palette has been designed, the remaining step is to map the original color of each pixel in the input image to its best match in the color palette. The simplest way is the exhaustive search method, but the computation is slow. In [1], a fast pixel mapping

algorithm using reduction of color space dimensionality (PMRC) is proposed. In order to preserve the color distributions, the color vectors are projected on a line which is selected by the mean color vector and the color that is the largest distance from the mean vector. Next, the triangular inequality principal is employed to shorten the computation time of finding the best match in the color palette. Since a large portion of color vectors are kicked-out when using the triangular inequality principal, the proposed method in [1] is much faster than the exhaustive search method.

Hwang and Chang [4] also proposed an algorithm using principal component analysis (PCA) as a faster approach to pixel mapping. The proposed scheme first computes two principal component directions (PCD) for the palette. Then, the projected values on PCDs are computed for each color in palette. Finally, as the PRMC, the projected values following the triangular inequality principle are used to reduce the computation time for finding the nearest color.

In this paper, a new, fast pixel mapping method for color quantization using principal component analysis is proposed. The method is an improved version of PMRC. Our proposed method first uses the first principal component axis in the principal component analysis technique to replace the projection line in PMRC. Next, a new projection value search algorithm is proposed to solve the shortcomings of PMRC. Finally, a three-dimensional projection filter method is employed to further accelerate the search time.

The remainder of this paper is organized as follows. The research related to the PMRC method is introduced in Section 2. Section 3 explains the details of our pixel mapping method for color quantization. In Section 4, the experimental results are presented and discussed. Finally, the conclusions are stated in Section 5

2 Pixel Mapping Algorithm Using Reduction of Color Space Dimensionality

Since our method is inspired by PMRC, in this section, we review the PMRC. PMRC maps color vectors of the color palette onto points on a line in the color space. In order to preserve the color distributions, the color vectors are projected on a line which is selected by the mean color vector and the color that is the largest distance from the mean vector.

Let N be the total number of color vectors in the color palette and $x_i = (r_i, g_i, b_i)$ be a tri-tuple including R, G, B values of the ith vector of the palette, where i = 1, 2, ..., N. Then the mean vector of the palette is computed by

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

The distances between color vectors and the mean vector in the palette are defined as

$$d_i = D(x_i, \bar{x}), \qquad (2)$$

where D(*,*) is the distance function between two color vectors.

Next, a vector k is chosen with the largest value of d_k in Equation (2). Assume the line l passes d_k and $\bar{x}$ in the 3D color space. The projection value p_i of the ith color vector on the line is obtained by

$$p_i = d_k - \frac{(x_i - \bar{x})\cdot(x_k - \bar{x})}{d_k}. \qquad (3)$$

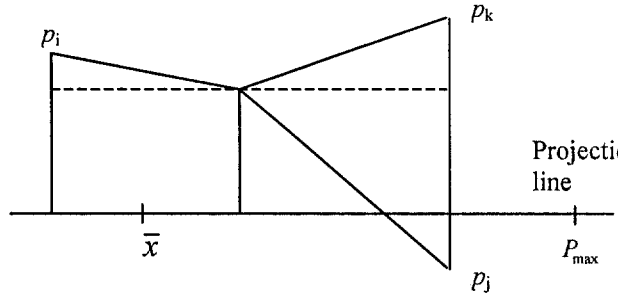
The palette colors can be considered as points in 3D space. As shown in Figure 1, the projection value of a pixel and a palette color is smaller than or equal to the actual distance between them. The symbols p_i, p_j, and p_k denote the ith, jth and kth palette colors, p_{max} is the palette color with the largest distance relative to the mean vector $\bar{x}$, and x is a pixel in the input image. Mathematically, let O_i and O_j be two points with distance fuction $D(*,*)$ and $P(O_i)$, $P(O_j)$ be their projection values. Then we have:

$$|P(O_i) - P(O_j)| \le D(O_i, O_j) \qquad (4)$$

for every pair of points O_i and O_j.

In [1], a fast pixel mapping algorithm based on the triangular inequality is proposed as follows.

Figure 1: Fast pixel mapping based on color dimensionality reduction and the triangular inequality



Algorithm Pixel mapping by the dimensionality reduction technique. [1]

Input: A digital color image and a palette with K colors.
Output: A quantized image.

1 Compute the mean color vector $\bar{x}$ of the palette colors.
2 For each palette color, compute the distance between the palette color and the mean color vector.
3 Find the color vector p_{max} with the largest distance relative to $\bar{x}$ among all palette colors and construct line l for projecting.
4 Project each palette color on line l, and collect the projection values as a set S. Sort S for the purpose of fast pixel mapping.
5 For each pixel x do

5.1 Project x on line l. Let PS_x be the projection values of x.
5.2 Set the value of a threshold ε to be a maximum integer value of.
5.3 Copy the set S to a new set SS.
5.4 Do the following steps to find the optimal color:
5.4.1 Among the set SS, find the nearest palette color p of x in terms of projection values using the binary search method.
5.4.2 If $\varepsilon > |PS_p - PS_x|$, then ε = $D(x,p)$; $SS \leftarrow SS-\{p\}$.
5.4.3 Else exit do-until loop.
5.5 Until the set SS is empty.
5.6 Replace the pixel's color with the palette color obtained in the last step.

Suppose the number of total pixels is N and the number of the palette colors is K. Then the complexity of the algorithm is $O(NK \log K)$.

3 The Proposed Method

In this section, we propose a new pixel mapping method for color quantization using principal component analysis. The idea of this method is inspired by PMRC. And the method is an improved version of PMRC. In this paper, the improvement is divided into three parts: the principal component analysis (PCA) technique, the projection value searching algorithm, and the three-dimensional projection filter method.

3.1 Principal Component Analysis Technique

The first improvement of our proposed method involves using the first principal component axis in the PCA technique to replace the projection line in PMRC. The projection line in PMRC is determined by two points, one of which is the mean $\bar{x}$ of the palette colors and the other is the palette color $p_{max} \in P$ (namely the largest distance relative to $\bar{x}$). Here, we cannot make sure whether the values on the projection line will preserve the maximum variances. As a rule, the larger variances of the projected values obtained, the less points needed to compute for measuring distortions. Fortunately, PCA has the ability to find a set of unit directions so that the projected values from the palette (three dimensional vectors) to these directions can maximally preserve the variances among vectors [2]. Now, let us have a brief look at PCA.

PCA is a powerful technique used in data analysis. The central idea of PCA is to reduce the dimensionality of a data set which consists of many interrelated variables. It is achieved by searching for the direction in data-space which has the highest variance, and subsequently projecting the data onto it.

For a set of observed d dimensional data vector $\{x_i\}$, $i \in \{1,2, \ldots, N\}$, we define the q principal axes $\{\mathbf{w}_j\}$, $j \in \{1,2, \ldots, q\}$ as those orthonormal axes onto which the retained variance under projection is maximal. Then it can be shown that the vector $\{\mathbf{w}_j\}$ is given by the q dominant eigenvectors which correspond to the largest eigenvalues of the sample covariance matrix

$$\mathbf{C} = \frac{1}{N}\sum_{i}^{N}(x_i - \bar{x})(x_i - \bar{x})^T, \qquad (5)$$

where $\bar{x}$ is the data sample mean defined in Equation (1), such that $\mathbf{C}\mathbf{w}_j = \lambda_j \mathbf{w}_j$. The q principal components of the observed data x_i are given by the vector $\mathbf{z}_i = \mathbf{W}^T(x_i - \bar{x})$, where $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_q)$. The variables $\mathbf{z}_j$ are uncorrelated such that the covariance matrix $\sum_i \mathbf{z}_i \mathbf{z}_i^T/N$ is diagonal with elements λ_j's. A complementary property of PCA is the principal component projection of all orthogonal linear projections which minimizes the squared reconstruction error $\sum_i \|x_i - \hat{x}_i\|^2$, where the optimal linear reconstruction of x_i is given by $\hat{x} = \mathbf{W}\mathbf{z}_i + \bar{x}$.

3.2 Projection Value Searching Algorithm

As mentioned in Section 2, the binary search method is employed to search for the nearest palette color of a given color in the inner loop of the PMRC algorithm. Since the set of projection values is pre-sorted, after the first search is completed, the remaining search should be performed in the vicinity of the previously searched value. Obviously, the binary search of the PMRC algorithm is not necessary. To solve this shortcoming of PMRC, a new projection value search algorithm is proposed in this section.

Assume the projection line obtained by the principal component analysis is line l, and let P_x be the projection value of a given pixel x on line l. The goal of the projection search algorithm is to find the best match for x in the color palette. As in PMRC, the first candidate, palette color P_n, is found using the binary search method. Therefore, the remaining palette colors are checked one by one in two directions of the neighboring pixels of the first searched value. They are the palette colors in the descending order of projection values P_i, where $1 \le i \le n$, and the palette colors in the ascending order of projection values P_i, where $n < i \le K$ and K is the number of the palette colors. Note that, according to the triangular inequality , for each searching loop, if the difference of P_i and P_x is greater than the threshold, the search stops. Here the threshold is the minimum distance between the candidate palette color and the pixel x.

3.3 Three-Dimensional Projection Filter Technique

The triangular inequality can be applied to any projection axis. If more projection axes are employed to filter the palette color vectors, the computation speed may be accelerated. Note that the overhead of the computation of the projection values on the axes should

not be high. In this paper, the three major axes X, Y, and Z are employed as the filtering projection axes.

Let O_i and O_j be two points with a distance function $D(*,*)$ and $XP(Oi)$, $YP(O_i)$, $ZP(O_i)$, $XP(Oj)$, $YP(O_j)$ and $ZP(O_j)$ be their projection values on axes X, Y, and Z, respectively. Then we have:

$$|XP(O_i) - XP(O_j)| \le D(O_i, O_j),$$
$$|YP(O_i) - YP(O_j)| \le D(O_i, O_j),$$
and $$|ZP(O_i) - ZP(O_j)| \le D(O_i, O_j),$$

for every pair of points O_i and O_j.

3.4 Computational Complexity of the Proposed Algorithm

Combine the three improvements stated in the previous section, and a fast pixel mapping algorithm is proposed as follows:

Algorithm Pixel mapping by our proposed method.

Input: A digital color image and a palette with K colors.
Output: A quantized image.

1 Compute the covariance matrix **C** of the palette colors.
2 Find the eigenvectors $\mathbf{w}_1$, $\mathbf{w}_2$, ..., $\mathbf{w}_q$ and eigenvalues λ_1, λ_2, ... , λ_q of the covariance matrix **C**.
3 Construct the first principal component line l with the linear equation $\mathbf{Y} = \mathbf{w}_1^t\mathbf{X}$.
4 Project each palette color on line l, and collect the projection values as a set S. Sort S for the purpose of fast pixel mapping.
5 For each pixel x do the following steps:
  5.1 Project x on line l. Let P_x be the projection value of x on line l.
  5.2 Set the value of a threshold ε to be a maximum integer value.
  5.3 Among the set S, find the nearest palette color P_m of x in terms of projection values using the binary search method.
  5.4 For each palette color x_i in the descending order of projection values P_i do the following steps, where $1 \le i \le m$:
    5.4.1 If $|P_i - P_x| > \varepsilon$, then exit for loop.
    5.4.2 If $|XP_i - XP_x| > \varepsilon$ or $|YP_i - YP_x| > \varepsilon$ or $|ZP_i - ZP_x| > \varepsilon$, then skip to next x_i.
    5.4.3 If $D(x, x_i) < \varepsilon$, then $\varepsilon = D(x, x_i)$, and set x_i to be the expected palette color.
  5.5 For each palette color x_i in the ascending order of projection values P_i do the following steps, where $m < i \le K$:
    5.5.1 If $|P_i - P_x| > \varepsilon$, then exit for loop.
    5.5.2 If $|XP_i - XP_x| > \varepsilon$ or $|YP_i - YP_x| > \varepsilon$ or $|ZP_i - ZP_x| > \varepsilon$, then skip to next x_i.
    5.5.3 If $D(x, x_i) < \varepsilon$, then $\varepsilon = D(x, x_i)$, and set xi to be the expected palette color.
  5.6 Replace the pixel's color with the final expected palette color.

Suppose the number of total pixels is N and the number of the palette colors is K. Then the complexity of the proposed algorithm is in time $O(NK)$. Compared to the complexity of PMRC's algorithm, which is in time $O(NK \log K)$, our proposed algorithm is superior to PMRC in time complexity.

4 Experimental Results

Our experiments were performed on an Intel Pentium-III 450 MHz PC. Each of the color images we used contained 512 × 512 pixels. The quality of the encoded image was evaluated by the peak signal-to-noise ratio (PSNR), which was defined as

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{MSE} dB.$$

Note that the mean-square error (MSE) for an n×n image was defined as

$$\text{MSE} = (\frac{1}{n})2 \times \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_{[i,j]} - \beta_{[i,j]})^2,$$

where $\alpha_{[i,j]}$ and $\beta_{[i,j]}$ denoted the original and quantized pixel values, respectively.

Tables 1 and 2 analyze the average number of real computations for measuring the distortion between a query color x and a palette color p_i, i.e., the number of computations for $D(x, p_i)$, $1 \le i \le n$. Table 3 shows the computation times (in seconds) of the proposed method, the exhaustive search method and the PMRC method for various levels of color, respectively. For each method, the images "Lena", "Peppers", "Jet", "House", "Baboon", and "Girl" were tested under 256, 128, 64, 32, 16 palette colors. Here the palette colors were obtained from Photoshop software. We see that, in the table, the computation time of our proposed method is much less than that of the exhaustive search method, and is superior to those of the PMRC method and Hwang and Chang's method in all cases. For example, using 256 palette colors and the test image "Lena", the computation time of our proposed method is 0.555 seconds. The execution speed is ten times faster than the exhaustive search method, three times faster than the PMRC method and two times faster than Hwang and Chang's method.

5 Conclusions

In this paper, a novel and fast pixel mapping method for color quantization using principal component analysis has been proposed. This method is an improved version of PMRC. The improvements are divided into three parts: the principal component analysis technique, the projection value searching algorithm, and the three-dimensional projection filter method. The experimental results show that the execution speed of our proposed method is much faster than that of the exhaustive search method. Moreover, our proposed algorithm achieves the time complexity of $O(NK)$, which is superior to the $O(NK\log K)$ of PMRC. Our proposed method is therefore a very efficient method for pixel mapping in color quantization techniques.

References

[1] S. C. Cheng and C. K. Yang, "A fast and novel technique for color quantization using reduction of color space dimensionality," *Pattern Reconginition Letters*, vol. 22, pp. 845-856, 2001.

[2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1991.

[3] P. Heckbert, "Color image quantization for frame buffer display," *Computer Graphic*, vol. 16, pp. 297-307, 1982.

[4] K. F. Hwang and C. C. Chang, "A fast pixel mapping algorithm using principal component analysis," Technical Report, Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan, 2001.

[5] A. Kruger, "Median-cut color quantization," *Dr. Dobb's Journal*, pp. 46-92, Sep. 1994.

[6] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communications*, vol. COM-28, pp. 84-95, Jan. 1980.

[7] T. Tasdizen, L. Akarun, and C. Ersoy, "Color quantization with genetic algorithms," *Signal Processing: Image Communication*, vol. 12, pp. 49-57, 1998.

[8] X. Wu and I. H. Witten, "A fast k-means type clustering algorithm," Technical Report, Department of Computer Science, University of Calgary, Canada, 1985.

[9] W. J. Lin and J. C. Lin, "Color quantization by preserving color distribution features," *Signal Processing*, vol. 78, pp. 201-214, 1999.

[10] S. J. Wan, P. Prusinkiewicz, and S.K.M. Wong, "Variance-based color image quantization for frame buffer display," *Color Res. Appl.*, vol. 15, pp. 153-162, 1998.

[11] S. J. Wan, S.K.M. Wong, and P. Prusinkiewicz, "An algorithm for multidimensional data clustering," *ACM Trans. on Math. Software*, vol. 14, pp. 153-162, 1988.

Table 1. Average of the number computation for searching the nearest palette color before the three-dimensional projection filter technique is employed

Number of Colors	Lena	Peppers	Jet	House	Baboon	Girl
16	4.95	5.12	4.24	4.17	5.7	5.00
32	7.79	14.28	13.78	7.27	7.89	7.33
64	11.29	25.12	24.9	12.07	11.64	9.02
128	10.62	16.08	15.86	11.32	17.30	19.14
256	17.29	35.64	27.75	36.9	29.68	34.51

Table 2. Average of the number computation for searching the nearest palette color after the three-dimensional projection filter technique is employed

Number of Colors	Lena	Peppers	Jet	House	Baboon	Girl
16	1.93	2.21	1.95	1.75	2.3	2.01
32	2.94	3.52	3.7	2.48	2.86	2.76

64	3.8	3.6	5.4	3.6	3.7	3.18
128	3.39	5.2	4.92	3.72	4.7	5.2
256	4.7	9.85	8.9	7.18	7.09	8.5

Table 3. The computation times (in seconds) of the proposed method, the exhaustive search method and the PMRC method for various levels of color.

24 bit images	Number of colors	PSNR(dB)	Computation times (sec)			
			Full search	PMRC	Hwang and Chang's	Ours
Lena	16	29.596	0.688	0.550	0.33	0.221
	32	31.455	1.047	0.770	0.34	0.273
	64	33.914	1.898	0.980	0.70	0.367
	128	35.016	3.680	1.370	0.88	0.398
	256	36.698	7.708	1.820	1.05	0.555
Peppers	16	27.712	0.670	0.660	0.30	0.250
	32	28.912	1.212	0.870	0.45	0.320
	64	30.98	2.011	1.150	0.67	0.402
	128	33.02	3.710	1.540	0.91	0.452
	256	34.98	7.702	1.930	1.39	0.572
Jet	16	31.226	0.578	0.550	0.30	0.250
	32	31.926	1.000	0.610	0.38	0.398
	64	34.786	1.906	0.820	0.58	0.500
	128	36.255	3.609	1.040	0.79	0.562
	256	37.671	7.047	1.260	1.02	0.688
House	16	30.687	0.570	0.600	0.29	0.221
	32	32.401	1.047	0.710	0.57	0.281
	64	34.542	1.914	0.930	0.58	0.438
	128	35.96	3.680	1.260	0.75	0.469
	256	37.671	7.070	1.480	1.09	0.781
Baboon	16	24.47	0.594	0.770	0.35	0.262
	32	26.391	1.000	1.050	0.46	0.312
	64	28.671	1.883	1.430	0.61	0.383
	128	29.963	3.625	1.870	0.85	0.484
	256	31.742	7.008	2.470	1.29	0.711
Girl	16	26.502	0.531	0.720	0.31	0.248
	32	28.588	1.039	0.880	0.42	0.289
	64	30.027	1.859	1.160	0.59	0.344
	128	31.608	3.711	1.490	0.79	0.492
	256	33.021	7.016	1.810	1.24	0.734

On mirroring, connected components labelling and topological properties of images encoded as minimized boolean function

Debranjan Sarkar
Variable Energy Cyclotron Centre, 1/AF, Bidhan Nagar, Kolkata - 700 064, India
Phone: +91 33 337 1230, Fax: +91 33 334 6871, E-mail: dsarkar@veccal.ernet.in
AND
Pradip K. Das
Department of Computer Science & Engineering, Jadavpur University, Kolkata- 700 032, India
Tel: +91 33 472 0353; E-mail: pkdas@ieee.org

Keywords: Connected components labelling, Mirroring, Euler number, Minimized Boolean function

Received: May 20, 2001

In the scheme for representing binary images (Sarkar 1996), an image is considered as a map of a Boolean function which is minimized to obtain the prime implicants to represent the image. Such a scheme was shown to achieve a drastic saving of storage compared to Linear Quadtrees and Interpolation-based Binary Trees. In this paper we present the procedures for mirroring an image with respect to any coordinate axis, labelling its connected components and finding its topological properties. The computational complexities of the algorithms are also discussed.

1 Introduction

The internal (or region) representation of an image or of a graphical object in two- and three-dimension is of considerable importance in light of its efficient manipulation and visualization. Such representations are currently being investigated and new ones are still being proposed. Each such representation is characterized by a distinct data structure and thus the associated operations which are applied to the objects must be tailored to that structure. The data structure may also influence other factors, for example, the appearence of the object on output or the speed of display. The conventional methods of such representations in the *set of codes* type are Linear Quadtrees (LQ) (Gargantini 1982a) and Interpolation-based Binary trees (IBB) (Ouksel & Yaagoub 1992). Another method of representation of binary images based on minimization of Boolean functions has been proposed (Sarkar 1996). The basic operations like translation and rotation (Sarkar et al. 1997) and union, intersection, complement etc. (Sarkar 1997) were carried out, based on this new technique. Improved representations for LQ, IBB etc. were proposed and a few image manipulations were discussed (Chung & Wu 1999). A generalised technique was developed (Sarkar & Das 2000) for finding boundary codes from region representation of set-of-codes type.

Efficient region representation techniques find use in many applications such as computer graphics, image processing, pattern recognition, robotics, computational geometry, VLSI layout, geographic information systems and cartography. The encoding based on the minimized Boolean functions (MBF) is considerably important since it saves much space and image operations are relatively inexpensive. The comparison of the MBF method with other important methods (LQ and IBB) has shown (Sarkar 1996) an improvement in space ranging from 30 to 45% over IBB and about 60% over LQ. This improvement is important because space minimization has a direct effect on the time complexity of image processing algorithms. It is useful to develop algorithms to facilitate various operations on images represented by the MBF technique. Topological properties (number of connected components, number of holes, Euler number etc.) are useful for global description of regions in the image plane. Labelling various disjoint, connected components of an image is one such operation which is of fundamental importance in image analysis (Gonzalez & Woods 1992).

In this communication, we report the algorithms for (i) mirroring of an image with respect to an axis, (ii) labelling the connected components of an image, and (iii) finding the topological properties associated with an image encoded in the minimized Boolean function method. The organization of the paper is as follows. Section 2 is a brief review of the MBF approach. In section 3, the algorithms for various operations, e.g. mirroring, connected components labelling and finding topological properties (like Euler number etc.) are given. Section 4 presents the computational complexities of the algorithms. Section 5 describes the experimental results and in section 6, concluding remarks are given.

2 Minimized Boolean function based encoding

In this approach, the image is considered to be a rectangular array of size ($2^r \times 2^c$) of unit square pixels each of which can be either black or white. Each pixel is represented by an r-digit gray code (for the corresponding row position) in conjunction with a c-digit gray code (for the corresponding column position), thereby making it a unique combination of row and column variables of $(r + c)$ digits. A binary image is considered as a Boolean function which is minimized by algorithmic technique due to Quine-McCluskey (Kohavi 1970) or by a heuristic approach (Brayton et al. 1984) to get a set of prime implicants. Prime implicants represent the largest possible black blocks which can be stored as a single code of $(r + c)$ digits. The details of the MBF approach may be found in (Sarkar 1996).

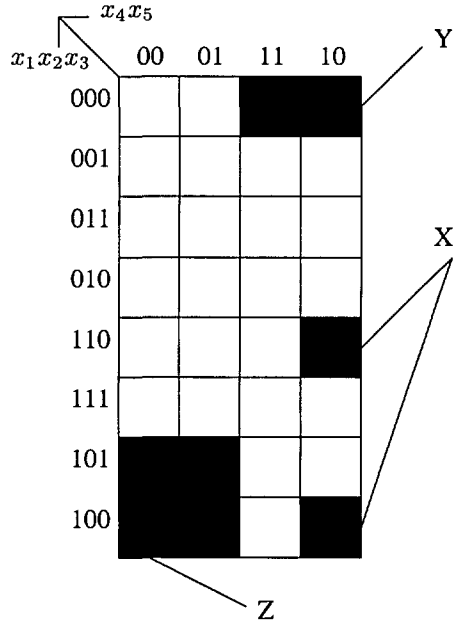


Figure 1: A $2^3 \times 2^2$ binary image

For example, in the image of figure 1 having $2^3 \times 2^2$ pixels, the block X may be represented as $x_1\bar{x}_3x_4\bar{x}_5$ or 120:10, where "1" indicates uncomplemented variable, "0" indicates complemented variable and "2" indicates a *don't care variable* (for which any binary value is acceptable). The symbol ':' is used as the separator between the row part and the column part of the code. Block Y is given by $\bar{x}_1\bar{x}_2\bar{x}_3x_4$ or 000:12 and block Z is represented as $x_1\bar{x}_2\bar{x}_4$ or 102:02. It is obvious that each variable in the code can assume three values 0, 1, and 2. Hence the code of a block in MBF approach is generally represented as ternary numbers (of base 3) and shown with the subscript 3. The decimal equivalent of a ternary number may easily be obtained (e.g. $10202_3 = 1\times3^4+0\times3^3+2\times3^2+0\times3^1+2\times3^0 = 101_{10}$)

3 Description of algorithms

The algorithms for mirroring, connected components labelling and finding the topological properties like number of connected components, number of holes and Euler number of binary images represented as MBF, are described in the following subsections.

3.1 Mirroring

Mirroring is one of the important geometric transformations frequently applied to images. Let us consider an image in a pixel array of size $2^r \times 2^c$. Four types of mirroring operations are considered as in (Schrack & Gargantini 1993). The mirroring operations are defined as follows:

(i) Mirroring w.r.t. the centre axis parallel to the abscissa: The pixel at location $(x, (2^r - 1) - y)$ in the transformed image corresponds to the pixel at location (x, y) in the original image.

(ii) Mirroring w.r.t. the centre axis parallel to the ordinate: The pixel at location $((2^c - 1) - x, y)$ in the transformed image corresponds to the pixel at location (x, y) in the original image.

(iii) Mirroring w.r.t. the main diagonal: The pixel at location (y, x) in the transformed image corresponds to the pixel at location (x, y) in the original image.

(iv) Mirroring w.r.t. the cross diagonal: The pixel at location $((2^k-1)-y, (2^k-1)-x)$ in the transformed image corresponds to the pixel at location (x, y) in the original image, where $k = r = c$.

The image space for the first two types of mirroring operations may not be a square ($r \neq c$) but the mirroring operations w.r.t. the main or cross diagonals require the image space to be a square one ($r = c = k$ (say)). The algorithms for mirroring transformation of images in linear quadtree form have been devised (Schrack & Gargantini 1993). Similar algorithms for mirroring of blocks of an image encoded by IBB technique are also proposed (Chung & Wu 1998, Sarkar & Gupta 1999). In the following we present the algorithms of mirroring of a block of an MBF scheme.

3.1.1 Mirroring w.r.t. the centre axis parallel to the abscissa (Mirror_x)

The MBF code of a block after mirroring w.r.t. the centre x-axis is obtained by

(i) keeping the column part unchanged

(ii) changing the most significant digit of *the row part* as follows:

$1 \rightarrow 0$; $0 \rightarrow 1$; $2 \rightarrow 2$

(iii) keeping the rest of the digits of the row part unaltered.

For example, in figure 2, block A is represented by an MBF code 102:002, where 102 is the row part and 002 is the column part. The most significant digit (i.e. the leftmost digit) of the row part of the MBF code of block A is 1. If it is changed to 0 and other digits remain unchanged,

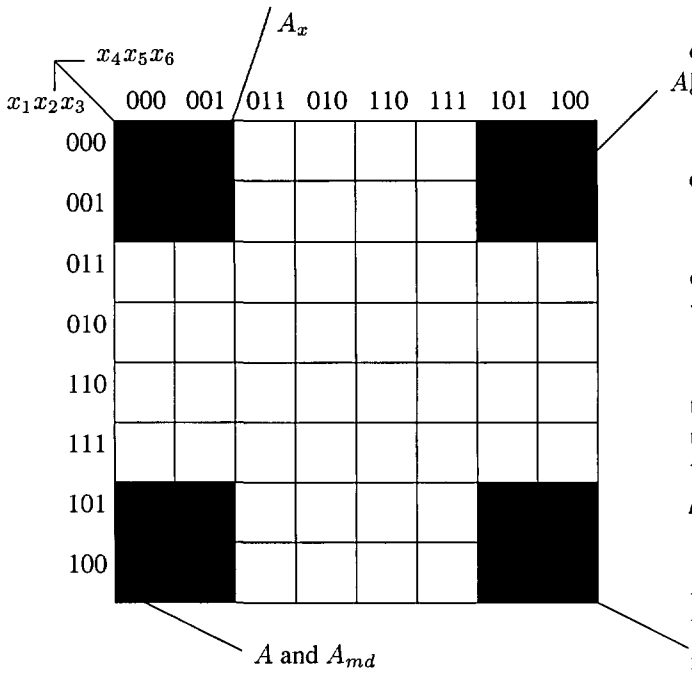


Figure 2: Various mirroring operations applied to a simple block

the code becomes 002:002 which is the code of block A_x (obtained after mirroring block A w.r.t. the centre x-axis).

3.1.2 Mirroring w.r.t. the centre axis parallel to the ordinate (Mirror_y)

The reflection property of the Gray codes suggests that in the MBF code of a block after mirroring w.r.t. the centre y-axis,

(i) the row part remains unchanged.

(ii) the most significant digit of *the column part* is changed as follows:

$1 \rightarrow 0; \quad 0 \rightarrow 1; \quad 2 \rightarrow 2$

(iii) the rest of the digits of the column part remain unchanged.

For example, in figure 2, the most significant digit (i.e. the leftmost digit) of the column part of the MBF code of block A is 0. If it is changed to 1 and other digits remain unchanged, the code becomes 102:102 which is the code of block A_y (obtained after mirroring block A w.r.t. the centre y-axis).

3.1.3 Mirroring w.r.t. the main diagonal (Mirror_md)

The main diagonal is the straight line from the lower left corner point to the upper right corner point of the domain. It is obvious that the mirroring w.r.t. the main diagonal is equivalent to a rotation operation w.r.t. the centre of the domain by 90^o (anticlockwise) and then a mirroring operation w.r.t. the centre axis parallel to the ordinate. The rotation of binary images encoded as MBF has already been studied (Sarkar et al. 1997).

For a 90^o anticlockwise rotation w.r.t. the centre of the domain, the following two steps are required to be followed:

(i) Interchange the original row part and column part

(ii) Apply the following mapping to the most significant digit of *the new row part* of the code

$1 \rightarrow 0; \quad 0 \rightarrow 1; \quad 2 \rightarrow 2$

Then apply the following mapping to the most significant digit of *the new column part* of the code for mirroring w.r.t. the centre axis parallel to the ordinate

$1 \rightarrow 0; \quad 0 \rightarrow 1; \quad 2 \rightarrow 2$

This indicates that for mirroring w.r.t. the main diagonal, the row and column parts of the original code have to be interchanged. Then the following mapping has to be applied to the most significant digit of *both the row and column parts* of the code.

$1 \rightarrow 0; \quad 0 \rightarrow 1; \quad 2 \rightarrow 2$

For example, in figure 2, the most significant digit (i.e. the leftmost digit) of the column part of the MBF code of block A is changed from 0 to 1 and that of the row part is changed from 1 to 0. The column part and the row part are then interchanged. The code becomes 102:002 which is the code of block A itself (obtained after mirroring block A w.r.t. the main diagonal).

3.1.4 Mirroring w.r.t. the cross diagonal (Mirror_cd)

The cross diagonal is the diagonal of the domain which is orthogonal to the main diagonal. It is observed that the mirroring w.r.t. the cross diagonal is equivalent to a rotation operation w.r.t. the centre of the domain by 270^o (anticlockwise) and then a mirroring operation w.r.t. the centre axis parallel to the ordinate.

For 270^o anticlockwise (or, 90^o clockwise) rotation operation w.r.t. the centre of the domain, the following two steps are required to be followed:

(i) Interchange the original row part and column part

(ii) Apply the following mapping to the most significant digit of the new column part of the code

$1 \rightarrow 0; \quad 0 \rightarrow 1; \quad 2 \rightarrow 2$

Then apply the same mapping to the most significant digit of the new column part of the code for mirroring w.r.t. the centre axis parallel to the ordinate.

This indicates that for mirroring w.r.t. the cross diagonal, because the last two mappings essentially cancel each other, only the row and column parts of the original code need to be interchanged.

For example, in figure 2, if the row part and column part of the code of block A is interchanged, the code becomes 002:102 which is the code of block A_{cd} (obtained after mirroring block A w.r.t. the cross diagonal).

3.2 Connected components labelling

Connectivity between pixels is an important concept used in establishing boundaries of objects and components of regions in an image (Gonzalez & Woods 1992). When only

the north, south, east and west neighbours of a pixel are considered part of its neighbourhood, the resulting regions are said to be 4-connected. When the north, south, east, west as well as northeast, northwest, southeast and southwest neighbours of a pixel are considered part of its neighbourhood, the resulting regions are said to be 8-connected. The neighbours of the pixel are known as 4-adjacent to the pixel in the former case and 8-adjacent in the latter case (Haralick & Shapiro 1992).

Labelling of connected components and finding the number of connected components are of fundamental importance in image analysis. Connected components labelling of images represented by quadtrees has already been accomplished (Samet 1981, Unnikrishnan et al. 1987). Connected components of images represented by linear quadtrees have also been labelled (Gargantini 1982b). In the following, we describe the algorithm for connected components labelling with 4- or 8-adjacency for the pixels and finding the number of connected components.

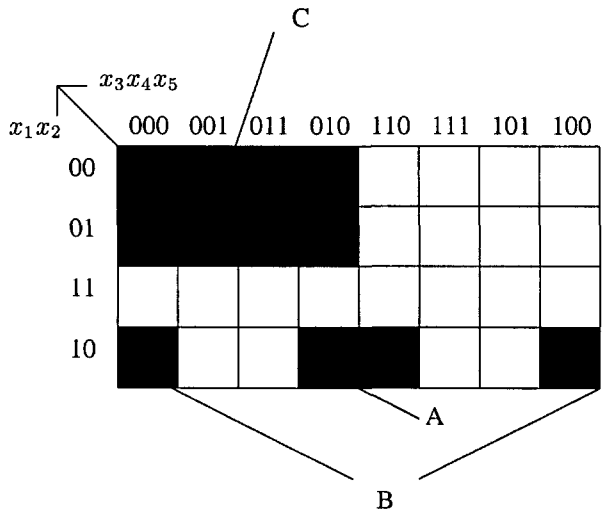


Figure 3: A $2^2 \times 2^3$ binary image

3.2.1 Physical continuity of pixels in a block

Let the image be represented by a set of codes $\{A_i, i = 1, 2, ..n\}$, where A_i is the code of the i^{th} prime implicant and n is the number of prime implicants or blocks.

In the MBF scheme, a prime implicant (A_i), designated by a ternary code, consists of pixels (or blocks) which may or may not be physically adjacent to each other. For example, in figure 1, block X consists of two pixels, which are not physically continuous but still they can be combined into a single prime implicant with code $120{:}10_3$.

A block is said to be physically continuous when all the individual pixels of this block occupy adjacent positions. The following rules may be applied to check the continuity of a block rowwise or columnwise.

(i) If there are *no* don't care terms in the row (or column) part of the code, the pixels belong to the same row (or column) and no question of row (or column) discontinuity arises.

For example, block Y in figure 1 is coded as 000:12, where the row part '000' has no don't care term. So block Y is certainly continuous rowwise.

(ii) If there is *only one* don't care term in the row (or column) part of the code which does not occupy the most significant position of the row (or column) part of the code, the block is continuous rowwise (or columnwise). If the single don't care term occupies the most significant (i.e. the leftmost) position and rest of the digits are not all 0's, then also, the block is continuous. If the single don't care term occupies the most significant position and rest of the digits are all 0's, then the block is not continuous.

For example, the MBF code of block Z in figure 1 is 102:02, where both the row part '102' and the column part '02' contain one don't care term each in the least significant (i.e. the rightmost) position. So, block Z is physically continuous both rowwise and columnwise, as is also obvious from the figure. Again, block A of figure 3 has a code 10:210. The row part is '10' and the column part is '210'. Here the single don't care term in the column part occupies the most significant (leftmost) position but since the rest of the digits are not all 0's, the block A is continuous columnwise. Whereas, block B in the same figure is coded as 10:200 with column part '200' has a single don't care term in the most significant position and all the other digits are 0's. So block B is discontinuous columnwise.

(iii) If there are *more than one* don't care terms *and* these terms in the row (or column) part of the code occupy consecutive least significant positions in the corresponding part of the code, the block is physically continuous rowwise (or columnwise).

For example in figure 3, the code of block C is 02:022, where the column part '022' contains more than one don't care terms which occupy the consecutive least significant positions (i.e. the rightmost position and the next one towards left). So block C is physically continuous columnwise. It may be noted that block C is continuous rowwise as well by rule (ii).

(iv) In all other cases, the block is not physically continuous and the don't care term in the most significant position in the row (or column) part is replaced with 0 and 1, thus dividing the block physically into two halves.

For example, block B (code 10:200) in figure 3 is discontinuous columnwise. The don't care term in the leftmost position of the column part is replaced with 0 and 1 and we get the codes 10:000 and 10:100 of the blocks which divide block B into two halves.

(v) The two halves are tested for continuity by the above rules till a continuous block is found (or when the original block is ultimately divided into individual discontinuous pixels).

It is important to know whether all the pixels of a block are continuous because all the pixels of only a continuous

block should have a unique label.

3.2.2 Finding the MBF code of a neighbouring block

If a block of an image is already labelled, its neighbouring pixel or block should have the same label. So it is important to find out the MBF code of a neighbouring block. Methods for finding the codes of the neighbouring block in MBF representation technique have been discussed in detail (Sarkar 1996). It has been observed there that the gray code next to the maximum of all possible combinations of the column variables of a block represents the column part of the neighbouring block in the east and the row part remains unchanged as the original block. Similarly, the codes of the neighbouring blocks in the west, north and south were found out.

In (Sarkar, 1996), the code of the north-east corner pixel of a block was also found out. The row part was the gray code previous to the minimum of all possible combinations of the row part of the code of the original block. The column part was the gray code next to the maximum of all possible combinations of the column part of the code of the original block. Similarly, the codes of the pixels in the north-west, south-east and south-west corners were found out.

3.2.3 Conditions for detection of adjacency

In case of 4-adjacency, a block B_2 is required to be labelled the same as that of block B_1 when any of the following conditions is satisfied:

(i) Block B_2 intersects with block B_1 [see figure 4(a)].

(ii) Block B_2 intersects with the neighbouring block (N_1) of B_1 in the north (N) direction [see figure 4(b)].

(iii) Block B_2 intersects with the neighbouring block (N_1) of B_1 in the south (S) direction [see figure 4(c)].

(iv) Block B_2 intersects with the neighbouring block (N_1) of B_1 in the east (E) direction [see figure 4(d)].

(v) Block B_2 intersects with the neighbouring block (N_1) of B_1 in the west (W) direction [see figure 4(e)].

To check the 8-adjacency, four more conditions, given below, are required to be checked. If any of these nine conditions is satisfied, block B_2 is labelled the same as that of block B_1.

(vi) Block B_2 intersects with the neighbouring pixel (N_1) of B_1 in the North-East (NE) direction [see figure 4(f)].

(vii) Block B_2 intersects with the neighbouring pixel (N_1) of B_1 in the North-West (NW) direction [see figure 4(g)].

(viii) Block B_2 intersects with the neighbouring pixel (N_1) of B_1 in the South-East (SE) direction [see figure 4(h)].

(ix) Block B_2 intersects with the neighbouring pixel (N_1) of B_1 in the South-West (SW) direction [see figure 4(i)].

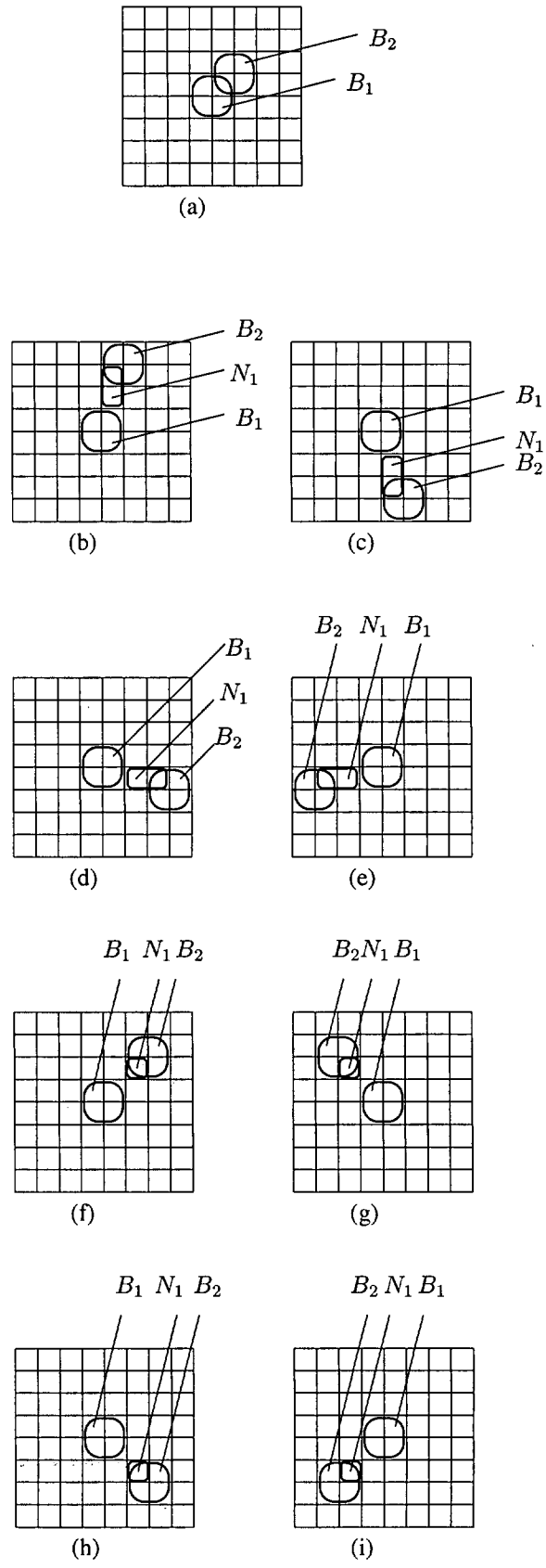


Figure 4: Various examples of connected blocks

The condition of finding whether two blocks intersect each other has been discussed (Sarkar 1997). It was observed there that if any particular digit in the codes of both the blocks under consideration are just complementary (0 and 1), the corresponding blocks do not intersect each other, otherwise they do intersect.

In the following section, we propose an algorithm for labelling the connected components of an image represented by the MBF technique.

The input to the algorithm is the set of MBF codes of the blocks.

3.2.4 Algorithm for labelling the connected components of an image represented as MBF

1. Get the set of codes in MBF
2. For all the codes
   Check whether the block corresponding to this code is physically continuous
   Endfor
3. If it is not continuous
   Divide it into parts till each part becomes physically continuous
   Put codes of each of these subblocks to the set of codes
   Endif
4. Let i = 1; current_label = 1
5. Get the i^{th} code
6. Check whether this block is already labelled
7. If not labelled
   Assign current_label to this block
   For all the labelled blocks with
   (label = current_label)
   Check whether this block is adjacent to any of the unlabelled blocks
   If adjacent
   Assign current_label to this unlabelled block
   Endif
   Endfor
   current_label = current_label + 1
   Endif
8. Repeat from step 5 after incrementing i till all the codes in the set are exhausted
9. Decrement current_label to get the number of connected components
10. End of algorithm

The labelled blocks and the number of connected components are the output of this algorithm.

3.3 Finding topological properties

A topological property is a property that is invariant to rubber-band distortions (Duda & Hart 1973). The topological properties of interest are (i) the number of connected components, (ii) the number of holes and (iii) Euler number.

The *number of connected components*, denoted by C, can be found out by the algorithm described in the preceding subsection.

The *number of holes*, denoted by H, in an image is one less than the number of connected components in the complement of the image (Duda & Hart 1973).

Thus $H = \bar{C} - 1$

where, $\bar{C}$ is the number of connected components in the complement of the image. This is valid when (i) the image space is infinity and (ii) 8-adjacency is used for white pixels if 4-adjacency is used for black pixels and vice versa.

In digital topology (Kong & Rosenfeld 1989), it is normally assumed that all pixels on the border of the image frame are white, to make all major results in infinite image frame to be valid in finite digital images. In the present scheme, the original image space is supposed to be infinity and to represent the image as a minimized Boolean function, we take a subspace of size $2^r \times 2^c$ such that all the black pixels in the image are totally confined in this subspace. In this case, all the pixels on the border of the image frame just outside the $2^r \times 2^c$ subspace are definitely white even if one or more pixels on the border of the image frame may be black. Thus the above relation is valid in images represented by the MBF method.

The MBF codes of the complement of the image are found out (Sarkar 1997). The number of connected components (C) in the original image and ($\bar{C}$) in the complemented image are found out following the algorithm described in section 3.2.4.

Thus the number of holes (H) is found out as $H = \bar{C} - 1$

The Euler number (E) is defined as $E = C - H$

Knowing the number of connected components and the number of holes in the image, Euler number can be easily found out.

4 Computational requirements

Let the binary image be placed in a domain of $2^r \times 2^c$ pixels.

Let n number of black blocks be required for its representation in the MBF technique. It is obvious that the computational complexity of the mirroring operation of the image w.r.t. any axis is of the order of $O(n)$.

It is obvious that in the worst case, step 7 of the connected components labelling algorithm, has to be iterated N times and steps 5 to 8 also have to be repeated N times, where N is the number of physically continuous blocks. So, the worst case computational complexity of the algorithm for connected components labelling of the image is of the order $O(N^2)$.

5 Experimental Results

The algorithms for various operations like mirroring, connected components labelling and finding topological properties (e.g. Euler number) were implemented in C language

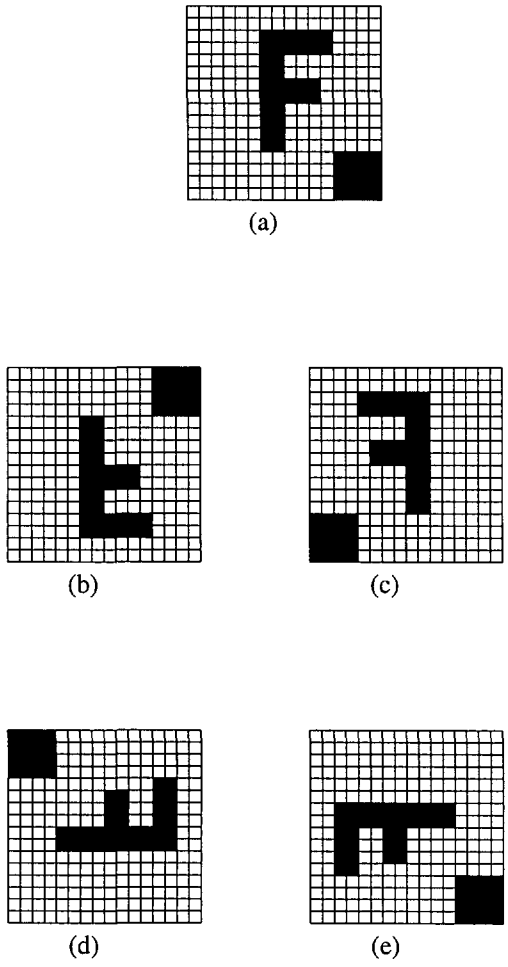


Figure 5: (a) A $2^4 \times 2^4$ sample image, (b)Mirroring of figure (a) w.r.t. the centre axis parallel to the abscissa, (c) Mirroring of figure (a) w.r.t. the centre axis parallel to the ordinate, (d) Mirroring of figure (a) w.r.t. the main diagonal, (e) Mirroring of figure (a) w.r.t. the cross diagonal

on a Digital Alpha Server 8200/5 (440MHz). These algorithms were tested on various images and encouraging results were obtained.

Figure 5(a) shows an image, taken from (Schrack & Gragantini 1993) on which different mirroring transformations were applied. The set of MBF codes for the image of figure 5(a) in ternary number is given by {0212:0102_3, 0102:1121_3, 1022:1022_3, 0102:2102_3, 2122:0102_3, 0012:1122_3}, where the subscript 3 indicates that the codes are in ternary numbers.

Figures 5(b), (c), (d) and (e) respectively show the mirroring of the image of figure 5(a) w.r.t. the centre x-axis, centre y-axis, the main diagonal and the cross diagonal. The codes of each of the basic blocks after the mirroring operations are found out by applying the algorithms proposed. These match with the set of the codes of the transformed figure.

Figure 6(a) shows another image on a $2^3 \times 2^3$ domain, the set of codes of which is {$112:121_3$, $010:210_3$, $100:200_3$, $102:022_3$, $212:101_3$, $022:000_3$, $022:011_3$, $020:201_3$}. The connected components of the image were labelled with 4-adjacency used for black pixels by applying the corresponding algorithm. Figure 6(b) shows the complement of the same image with connected components labelled with 8-adjacency used for black pixels i.e. the 8-adjacency used for white pixels in the original picture.

The number of connected components, the number of holes and the Euler number are thus found.

Fig. No.	No. of black pixels	No. of blocks in MBB	Total Execution time			
			Mirroring (μs)			
			x	y	md	cd
5(a)	50	6	118.8	121.9	124.6	123.9
6(a)	28	8	123	123	128	125
7	11501	2524	31930	31900	33360	33030

Table 1: Total execution time for different mirroring operations

	Mirror_x	Mirror_y	Mirror_md	Mirror_cd
Hardware	8.13	9.06	9.88	11.37
Software	8.02	8.84	51.69	57.84

Table 2: Execution time (in μseconds) for the bincode operations (Chung & Wu 1998)

Fig. No.	No. of blocks in MBB	Total Execution time	
		Connected Components Labelling	
		4-adj	8-adj
5(a)	6	1.08ms	1.48ms
6(a)	8	0.29ms	0.31ms
7	2524	60.1 sec	39.3 sec

Table 3: Total execution time for connected components labelling

The mirroring and connected components labelling operations were carried out on many other images including 'Monalisa' shown in figure 7. This figure is taken from (Sarkar 1996).

Table 1 shows the total execution time of various mirroring operations. These are compared with the execution time of each of these operations available for the bincoded images (Chung & Wu 1998), which is reproduced in table 2. An interesting observation is that the mirroring operation w.r.t. the cross diagonal is faster compared to the mirroring operation w.r.t. the main diagonal in the proposed algorithm. This is because the 'Mirror_md' operation requires additional mapping of two digits per block as compared to the 'Mirror_cd' operation. Whereas, as shown in table 2, the 'Mirror_md' operation is faster compared to the 'Mirror_cd' operation for the bincoded images (Chung & Wu 1998).

The total execution time for labelling the connected components (4-adjacent and 8-adjacent) is given in table 3. These operations are much slower in comparison with the mirroring operations, as expected. Our results are found to be comparable to the existing methods for LQ due to (Gargantini 1982b, Unnikrishnan et al. 1987).

6 Conclusions

The MBF approach is considered as one of the competitive schemes for representation of binary images, mainly because of its storage saving feature compared to LQ and IBB technique. Moreover, both the LQ and IBB techniques are applicable for a square ($2^n \times 2^n$) image domain. But an image placed in a rectangular ($2^r \times 2^c, r \neq c$) domain may be represented by MBF approach. This paper deals with some important image operations viz. mirroring, connected components labelling, and finding topological properties like Euler number etc. These operations are possible on images placed in a rectangular domain of pixels. Only the mirroring operations with respect to the main and orthogonal diagonals require the image to be placed in a square domain. An important feature of the MBF technique is that the blocks contain pixels which may or may not be physically continuous. So special care is required to be taken while labelling the connected components of an image represented as MBF codes. The question of physical discontinuity of a block does not arise in case of LQ or IBB techniques because the blocks in these data structure are inherently continuous. The computational complexities of the mirroring operation and connected components labelling are of the order of O(n) and $O(N^2)$ respectively, where n is the number of blocks required for its representation and N is the number of physically continuous blocks. Since the number of blocks in MBF is generally much less compared to those for IBB or LQ, improvement by a constant factor in time for those operations is achieved in the MBF approach.

Acknowledgements

The constant encouragement throughout this work by Dr. Bikash Sinha and Mr. S.K. De is highly acknowledged. The contributions during the initial phase of this work by Mr. Aditya Anirban Saha and Mr. Jayanta Madhu, summer trainees from the Indian Institute of Science, Bangalore, are gratefully acknowledged. The authors appreciate the reviewers for their constructive comments that improved the presentation and quality of this paper.

References

[1] Brayton, R. K., Hachtel G. D., McMullen C. T. & Sangiovanni-Vincentelli A. L. (1984) *Logic Minimization Algorithms for VLSI Synthesis.* Kluwer Academic Publishers, Boston.

[2] Chung, K.-L. & Wu J.-G (1998) Fast implementations for mirroring and rotating bincode-based images. *Pattern Recognition* 31, 12, p.1961-1967.

[3] Chung, K.-L. & Wu J.-G (1999) Improved representations for spatial data structures and their manipulations. *Informatica* 23, p. 211-221.

[4] Duda, R. O. & Hart P. E. (1973) *Pattern classification and scene analysis.* Wiley-Interscience, New York.

[5] Gargantini, I. (1982a) An effective way to represent quadtrees. *Comm. ACM* 25, 12, p.905-910.

[6] Gargantini, I. (1982b) Detection of connectivity for regions represented by linear quadtrees. *Computer Mathematics with Applications* 8, 4, p.319-327.

[7] Gonzalez, R. C. & Woods R. E. (1992) *Digital Image Processing.* Addison Wesley, New York.

[8] Haralick, R. M. & Shapiro L. G. (1992) *Computer and Robot Vision.* Addison Wesley, New York.

[9] Kohavi, Z. (1970) *Switching and Finite Automata Theory.* Mc Graw Hill Inc. N.Y.

[10] Kong, T. Y. & Rosenfeld A. (1989) Digital topology: introduction and survey. *Computer Vision, Graphics and Image Processing* 48, p.357-393.

[11] Ouksel, M. A. & Yaagoub A. (1992) The interpolation-based bintree and encoding of binary images. *CVGIP:Graphical Models and Image Process.* 54, 1, p.75-81.

[12] Samet, H. (1981) Connected component labelling using quadtrees. *Journal of ACM* 28, p.487-501.

[13] Sarkar, D. (1996) Boolean function-based approach for encoding of binary images. *Pattern Recognition Letters* 17, p.839-848.

[14] Sarkar, D (1997) Operations on binary images encoded as minimized Boolean functions. *Pattern Recognition Letters* 18, p.455-463.

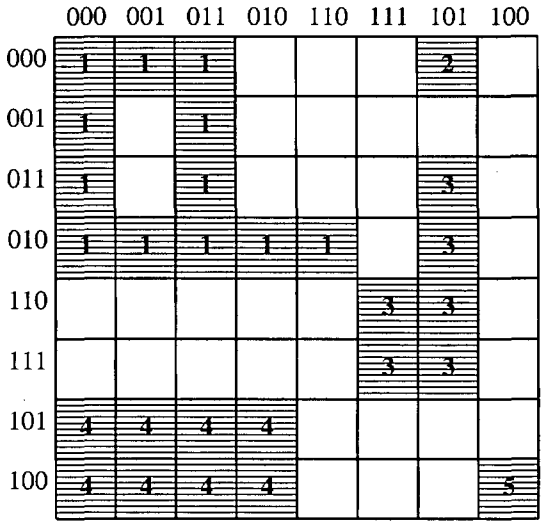
[15] Sarkar, D., Banerjee S. & Chattopadhyay S. (1997) Translation and rotation of binary images encoded as minimized Boolean functions. *Pattern Recognition Lett.* 18, p.157-163.

[16] Sarkar, D. & Gupta N. (1999) Operations on binary images represented by interpolation based bintrees. *Pattern Recognition Letters* 20, p.395-403.

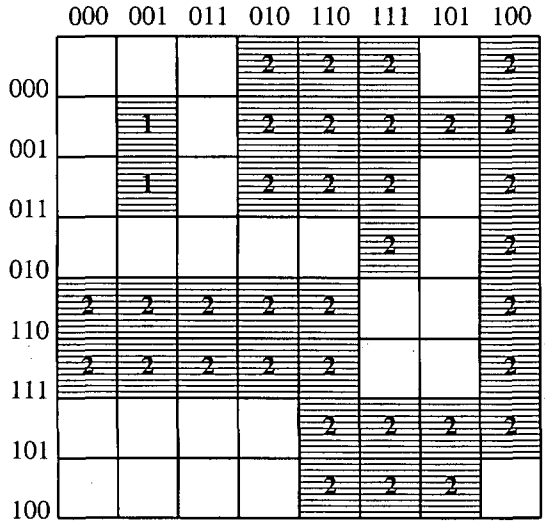
[17] Sarkar, D. & Das P. K. (2000) Generalized approach for finding boundary codes from region representation of set-of-codes type. *International Journal of Pattern Recognition and Artificial Intelleligence* 14, p.1039-1052.

[18] Schrack, G. & Gargantini I. (1993) Mirroring and rotating images in linear quadtree form with few machine instructions. *Image and Vision Computing* 11, 2, p.112-118.

[19] Unnikrishnan, A., Venkatesh Y. V. & Priti Shankar (1987) Connected component labelling using quadtrees - a bottom-up approach. *The Computer Journal* 30, 2, p.176-182.



(a)



(b)

Figure 6: (a) A $2^3 \times 2^3$ binary image with connected components labelled with 4-adjacency used for black pixels, (b) Complement of figure (a) with connected components labelled with 8-adjacency used for black pixels

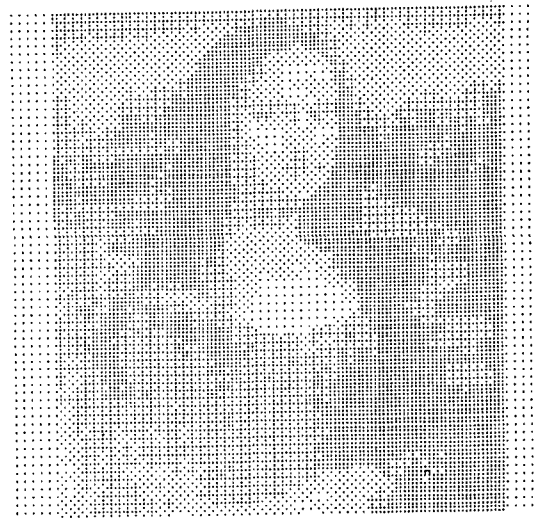


Figure 7: Monalisa

The next generation Internet protocol

Arathi Ramani
Department of Computer Engineering
Thadomal Sahani Engineering College, Bandra, Mumbai - 400050, India
E-mail: arathi@ieee.org
AND
Salim Vhora
Department of Electronics
K. J. Somaiya College of Engineering, Vidyavihar, Mumbai - 400077, India.
E-mail: salim_v@hotmail.com
AND
S. Sanyal
School of Technology & Computer Science
Tata Institute of Fundamental Research
Mumbai - 400005, India.
E-mail: sanyal@tifr.res.in

Keywords: Internet, protocol, next generation

Received: February 7, 2001

This paper describes and analyzes the key issues underlying the design of the Internet Protocol version 6 (IPv6). The paper highlights gains made by the new design and analyzes possible weaknesses as well. This gives an overview of the features which will be available in IPv6, why they are included, and how they will be implemented. It discusses certain significant algorithms which will form part of the new design. Differences between IPv6 and the existing protocol, IPv4, have been pointed out wherever a new feature has been introduced or an old one changed. The main focus of the paper is the provisions that the designers of IPv6 sought to make, which were not covered by IPv4. For this reason, issues like security, autoconfiguration and real-time communication have been dealt with in detail. Additionally, the address architecture of the new protocol has been explained, and changes in the formats and headers have also been discussed. Data fields have been described, where appropriate, to give a clearer picture of how messages will be structured under the new protocol.

1 Introduction

The Internet started out as a small research network intended for use by universities and the academic community. It was created to support a few simple distributed applications: FTP (the File Transfer Protocol), email and remote access using TELNET. From this modest beginning the Internet became a household name and now connects millions of individuals, corporations and academic institutions worldwide. The changing user profile caused a change in the whole Internet environment, which became increasingly multimedia-oriented, with the hugely popular WWW (World Wide Web) at the forefront. The present Internet needs strong support for real-time traffic, flexible congestion-control schemes, and security features. E-commerce and similarly demanding applications are forcing the move towards a more powerful Internet. The existing Internet, together with the Internet Protocol Version 4 (IPv4) currently in use, is not able to meet these requirements easily.

Some Limitations of IPv4 and Corresponding Improvements in IPv6: Internet protocols were first developed in 1969, when no one could have foreseen what the Internet would become. Although IPv4 has stood the test of time exceedingly well, there are some areas where it is unable to deliver the performance now required ([1]),([2]).

IPv4 Addressing: IPv4 uses a 32-bit address space, which provides for a total of 2^{32} (approximately 4 billion) addresses. These addresses are allotted in different classes of address space which support different numbers of host addresses.

- Class A address space: 16 million host addresses
- Class B address space: 65,000 host addresses
- Class C address space: 256 host addresses.

Major universities and corporations that played a founding role in the development of the Internet were assigned class A address space. Since no organization is likely to use 16 million addresses, utilization was sparse. This leads to considerable wastage, since once addresses have been assigned to a particular network, they are tied up with that network, whether actually used or not. Networks are proliferating rapidly, and with increased demand, there is a shortage of

address space. This problem, first predicted in the late 80s, became a major concern by 1992. The Internet Engineering Task Force (IETF) started working on a solution which would absorb and sustain future growth. Three areas of concern were identified:

- Scarcity of the Class B address space
- Inflation in the size of the routing tables for the Internet
- Eventual exhaustion of the 32-bit address space.

In order to provide a prompt solution to the first two problems, IETF developed Classless Inter-Domain Routing (CIDR). The official documentation of CIDR is available in RFC 1517([3]), 1518([4]), 1519([5]) and 1520([6]). This, however, was only a short term solution since the third problem could not be tackled in this fashion ([7]).

Another shortcoming lies in the fact that IPv4 is best employed for unicast addressing (explained later), where a single address bit pattern corresponds to a single host (point to point). It offers poor support to other forms of addressing which reduces the flexibility.

IPv6 Addressing([8]): IPv6 assigns a unique address for each connection between a computer and a physical network. IPv6 addressing differs from IPv4 addressing in a markedly different way. Addresses do not have defined classes. Here a prefix length must be associated with each address (e.g. in a routing table) to enable software to know where the prefix ends. Secondly, IPv6 defines a set of special addresses that completely differ from IPv4 special addresses. IPv6 does not include a special address for broadcasting on a given network. Essentially, each IPv6 address is one of three basic types: Unicast, Multicast and Anycast, explained later in Section 3.3.

Performance: LANs and WANs are constantly progressing to ever-higher data rates. Gigabit Ethernets have come into being. The increasing number of services (particularly graphics-related services), available over the Internet, has resulted in a rise in the ratio of external traffic (that leaves the local network) to internal traffic. With their immense speed and the increased load, routers need to be much faster than before, so as to be able to utilize the high-speed links to their full capacity and also to handle heavy traffic. Although the routers today can handle the traffic, improvement in design of the IP can significantly improve performance by reducing the size of the routing tables.

Quality of Service: This is the measure of performance that reflects transmission quality and service availability of a transmission system ([9]). There are standards for QoS which exist in IPv4. In IPv4, real-time traffic relies on the Type of Service(ToS) field. The packet is identified using a UDP or a TCP port. This identification cannot be done when the packet is encrypted. Also the ToS field has limited functionality and there have been various interpretations of it which are not consistent with each other.

Some parameters control the amount of traffic the source router sends over on the IPv6. If any switch (along the path) cannot accomodate the requested parameters, the request is rejected, and a rejection message is transmitted to the request originator.

Security: IPv4's only provision for security is in the form of an optional security label field. It is possible to provide for end-to-end security at the application level, but a standardized IP-level security service would be far more convenient as it would take the burden of providing security away from the application.

2 Some features of IPv6

Internet Protocol version 6 is the next generation Internet Protocol. It has been created with a view to providing the features that IPv4 lacks, providing faster, better service and meeting the growing demands of the Internet.

2.1 IPv4 Header

The IPv4 header **([12], page 10)** is variable length and includes several fields. This leads to a lot of processing overhead. Further, since IPv4 allows fragmentation at the router level, this adds to the processing delay. Table 1 describes the fields of the IPv4 Header. A summary of the contents of the Internet Header (IPv4) Format follows **([15])** :

Version: 4 bits: Version field indicates the format of the internet header. A value of 0100 indicates IPv4.

IHL (Internet Header Length Field): 4 bits: This field is the length of the internet header in 32 bit words and points to the begining of the data.

Type of Service (ToS): 8 bits: This field provides an indication of the abstract parameters of the quality of service desired, which in turn guides the selection of the actual service parameters, when transmitting a datagram through a particular network. Several networks offer service precedence, which treat high precedence traffic as more important than other traffic. The major choice is a three way tradeoff between low-delay, high reliability and high-throughput. Bits 0 – 2: Precedence.
Bit 3: 0 = Normal Delay; 1 = Low delay.
Bit 4: 0 = Normal Throughput; 1 = High Throughput.
Bit 5: 0 = Normal Reliability; 1 = High Reliability.
Bits 6 – 7: Reserved.

Total Length: 16 bits: Total Length is the length of the datagram, measured in octets, including internet header and data. This field allows the length of a datagram to be up to 65,536 octets. Such long datagrams are mostly impractical, 576 octets (or lower) being the standard length. This size allows a data block of 512 octets and 64 header octets to fit in a datagram.

Identification: 16 bits: An identifying value assigned by the sender to aid in assmbling the fragments of a datagram.

Table 1: **The IPv4 Header**

0	4	8	16	19	24	31
Vers	**IHL**	**Service Type**	**Total Length**			
Identification			**Flg.**	**Fragment Offset**		
Time to Live		**Protocol**	**Header Checksum**			
Source Address						
Destination Address						
IP Options					**Padding**	

Flags: 3 bits:
Bit 0: Reserved: must be zero.
Bit 1: DF: 0 = May Fragment
DF: 1 = Donot Fragment
Bit 2: MF: 0 = Last Fragment
MF: 1 = More Fragments.

Fragment Offset: 13 bits: This field indicates where in the datagram this fragment belongs. The fragment offset is measured in units of 8 octets.

Time to live: 8 bits : This field indicates the maximum time the datagram is allowed to remain in the internet system. This is measured in units of seconds, intention is to cause undeliverable datagrams to be discarded.

Protocol: 8 bits: This field indicates the next level protocol used in the data portion of the internet datagram.

Header Checksum: 16 bits: A checksum on the header only. Since some header fields change (e.g. Time To Live), this is recomputed and verified at each point that the internet header is processed.

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header. For purposes of computing the checksum, the value of this field is taken as zero. It is simple to compute checksum and experimental evidence indicates its adequacy. But this is provisional and may be replaced by a CRC procedure, depending on the situation.

Source and Destination Address: 32 bits each.

Options: To keep the headers of most datagrams small, IP defines a set of options that can be present, if needed. When an IP datagram does not carry options, the header length field (labeled IHL) contains 5, and the header ends after the DESTINATION IP ADDRESS Field. Because the header length is specified in 32-bit multiples, if options do not end on a 32-bit boundary, PADDING that contains zero bits is added to make the header a multiple of 32 bits ([7]).

2.2 IPv6: The new protocol

Faced with the increasing inadequacy of IPv4, it became necessary to design a new protocol that was equipped to handle the present-day Internet. The successor to IPv4 is the Internet Protocol version 6 (IPv6). The new protocol not only overcomes the shortcomings of the old protocol but also adds facilities for mobility, anycast addressing, packet tracing and, in particular, larger address space ([11]).

2.3 The IPv6 Header

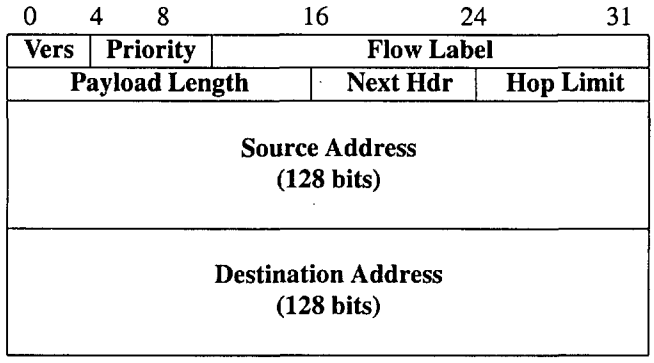
The header for the new protocol **([12], page 10)** has been designed to be much simpler than the old IPv4 header, which has several fields including a variable length options field (as shown in Table 1). This field was used to specify options for special case packets. The new header, in contrast, has only six fields and two addresses. Table 2 describes fields of the IPv6 Header.

One of the most significant improvements **([13])** in the IPv6 header is the fact that it is a fixed-length header, which simplifies processing. In fact it becomes simple enough to be handled by Application Specific Integrated Circuits. The fields in the IPv6 header are:

1. Version field (4 bits)
2. Priority value (4 bits)
3. Flow Label (24 bits)
4. Length of the payload (16 bits)
5. Type of the next header (8 bits)
6. Hop limit (8 bits) The header includes 128-bit source and destination address fields.

One change that has been subject of some controversy was the decision to omit the header checksum field, regarded by some as a risky move. However, since checksum computation takes place in both the data link and transport layers, its absence in network layer is unlikely to cause worries.

Table 2: **The IPv6 Header**



2.3.1 Version field

This field carries the same meaning as it did in IPv4 and is used to identify the version of IP being used.

2.3.2 Flow Label and Priority fields

These are both used in the handling of real-time traffic. The priority field is used to assign levels of precedence to different packets (this supports real- time traffic because real-time packets can be assigned higher priority). The flow label field is used to identify a packet as belonging to a particular flow. A flow is a sequence of packets from a particular source to a particular destination, all of which receive the same treatment at a router. The idea of using flows is that for the packets in a flow, routing algorithms do not have to be computed every time. Once a packet is identified as belonging to a flow, a router simply forwards it along the path designated for that flow.

2.3.3 Payload length

The payload length field specifies the length of the data carried after the header. IPv6 supports a maximum payload length of $2^{16} = 64KBytes$ for normal packets. However, packets of larger size can be supported through jumbograms ([34]).

2.3.4 Hop Limit

This field is intended to ensure that a packet does not live infinitely in the network. It specifies the maximum number of hops for which a packet is allowed to live. This number is decremented by every router forwarding the packet at each relay. The Hop Limit is similar to the IPv4 "Time to Live" field except that there is no historical relation to the amount of time (in seconds) that the packet is queued at the router ([8]). The notion of hop limit is easier to implement because it is very difficult to estimate the waiting time of a specific packet, which also has to be factored into the calculation. The decision to restrict the hop limit field to 8 bits has not found favour with critics, who feel that this restriction allows a maximum number of only 256 hops per packet in the network, which may be too few. It is to be hoped that the superior performance offered by IPv6 will make it unnecessary for any packet to spend more than 256 hops on the network.

2.3.5 Next Header

One of the major changes made by IPv6 ([1]; [2]; [12] **Page 17**) is in the organization of IP packets. In IPv4, the header is immediately followed by the transport protocol data. Special-case treatment of those packets that required it was provided for by the options field in the main header. IPv6 uses a fixed header format. If a packet requires special case treatment, it can be specified by an extension header. In such cases, the "next header" field is set to the type of extension header to follow. For a simple data packet, without any extension headers, the next header field is set to the transport protocol type, UDP or TCP.

The interleaving of extension headers results in a daisy chain ([12], **Page 15**) of headers. Each extension header can in turn specify another extension header or a transport protocol type.

Tables 3, 4 and 5 show different organizations of headers. Table 3 shows Packet without Extension Headers, Table 4 shows Packet with Routing Extension Header and Table 5 shows Packet with Routing and Fragment Extension Headers.

The IPv6 specification defines six extension headers ([13]) :

- **Hop-by-hop options header:** Defines special options that require hop-by-hop processing.
- **Routing Header:** Provides extended routing.
- **Fragment Header:** Contains fragmentation and reassembly information.
- **Authentication Header:** Provides Packet integrity and authentication information.
- **Encrypted Security Payload:** Provides privacy.
- **Destination Options Header:** Contains optional information to be examined by the destination node.

Address Space: IPv6 allows 128-bit source and destina-

Table 3: **Packet without Extension Header**

IPv6 Header	**TCP Header+Data**
Next Header = TCP	

Table 4: **Packet with Extension Header**

IPv6 Header	**Routing Header**	**TCP Header+Data**
Next Header = Routing	**Next Header =** TCP	

tion addresses, which means a total of $2^{128} = 3 * 10^{38}$ addresses. With such a large address space, each device can have a globally unique address and there is no danger of running out of addresses.

2.4 A Comparison of Two Headers

The new header of IPv6 is in fact much simpler than that of the classic IPv4. Only 6 fields and two addresses are counted, while the IPv4 has 10 fixed header fields, two addresses and some options. The Version Number kept the same meaning and the same position, in both cases it is encoded in the very first 4 bits of the header. It has been thought that whenever possible, IPv4 and IPv6 will be demultiplexed at the media layer. For example, IPv6 packets will be carried over Ethernet with the content type 86DD (Hexadecimal) instead of IPv4's 8000.

Six fields were suppressed, the header length, the type of service, the identification, the flags, the fragmentation offset and the header checksum. Three fields were renamed and in some cases slightly redefined: the length, the protocol type and the time to live. The option mechanism was entirely revised, and the two new fields were added : a priority and a flow label.

2.4.1 Simplifications

The IPv4 header was based on the state of the art of 1975. After 20 years, we could proceed with three major simplifications:
- Assign a fixed format to all headers
- Remove the header checksum
- Remove the hop-by-hop segmentation procedure

IPv6 headers contain no optional element but unlike IPv4 where a variable length option field is active, here extension headers are appended after the main header. Obviously, there is no need in IPv6 for a header length field (IHL).
Removal of Header Checksum has its advantage in terms of Header Processing meaning there is no need to check and update the checksum at each router point. The obvious risk is that undetected errors may result in misrouted packet. This risk is practically very low as most encapsulation procedures include a packet checksum.
IPv4 included a fragmentation procedure so that senders could send large packets without worrying about the capacities of routers. The large packets could be fragmented into smaller size, if necessary. The recipients would wait for the arrival of all these segments to reconstitute the packet. Incidentally, successful transmission of a packet depends on the successful transmission of each fragment. In IPv6, hosts should learn the maximum acceptable segment size through a procedure called path Maximum Transmission Unit (MTU) discovery. Larger than this size packets will simply be discarded. Hence, in IPv6, the segmentation control fields are not necessary. IPv6 networks have a general limit of payload size of 536 octets, this or smaller size packets will move freely.
In IPv6, Type of Service field has been removed as in IPv4 (in reality) this field was not frequently set by applications. IPv6 provides differnt mechanism for handling these preferences.

2.4.2 Some IPv6 Headers

- **Payload Length:** The Total Length of IPv4 is replaced by the Payload Length of IPv6 which is, by definition, the length of the data carried after the header. In both IPv4 and IPv6, the length field is of 16 bits, which limits the packet size to 64 kilobytes. For larger packets, IPv6 has the "Jumbogram" option.

- **Next Header:** The protocol Type field of IPv4 was renamed to Next Header type to reflect the new organization of the IP packets. In IPv4, the IP header is followed by the Transport Protocol Data, e.g. UDP or TCP packet. Simplest type of the IPv6 packets will have exactly the same structure, where the next header type will be set to the protocol type of UDP or TCP. IPv6 also allows to interleave Extension Headers between the IP and TCP or UDP payload. The Next Header Type will then be set to the type of the first extension header.

- **Hop Limit Field:** The Time to Live (TTL) field has

Table 5: **Packet with Routing and Fragment Extension Header**

IPv6 Header Next Header = Routing	Routing Header Next Header = Fragment	Fragment Header Next Header = TCP	TCP Header +Data

been renamed to Hop Limit field. In IPv4, the time to live was expressed as a number of seconds, indicating how long the packet could remain in the network before getting destroyed. Time to Live was based on the fact that if packets were allowed to remain infinitely in the network, old copies of the packet would create problem. In IPv4, the TTL will be decremnted by each router by 1 second or by the time spent waiting in the router queues is larger than 1 second. But it is very difficult to correctly estimate the waiting time of a specific packet as these are counted in milliseconds, hence most routers simply decrease the count by 1 at each router. In IPv6, this mechanism of decrementing the count by 1 has been implemented and the field has been renamed. It counts the number of hops, not number of seconds.

2.4.3 New Fields

Two new fields in the IPv6 header have been added, the Flow Label and the Priority. These two are added to help handling Real-Time traffic. The Priority Field has similarity with the Precedence Field of IPv4. The Flow Label is used to distinguish packets that require the same treatment, i.e. they are sent by a given source to a given destination with a fixed set of options.

3 IPv6 Addressing Details([14])

IPv6 has been designed with a goal in mind, to enable high-performance, scalable internet which will operate for a long time to come. IPv6 will allow Internet backbone designers to create a flexible and global routing hierarchy. The hierarchical address system is similar to that of the National and International Telephone systems. Large central-office switches need only a three-digit national area code prefix to route a long-distance telephone call to the correct local exchange.

Apart from the relative size of the address fields of the two protocols, IPv4 (32-bit address) and IPv6 (128-bit address), another important point is the relative abilities of these protocols to provide a hierarchical address space that facilitates efficient routing. IPv4 was initially designed with Class A, B and C addresses which provided progressively lower addressing capability but did not provide a hierarchy that would allow a single high-level address to represent many lower level addresses. Hierarchical address systems work in much the same way as telephony country codes or area codes. This allow long-haul phone switches to route calls efficiently to the correct country or region using only a portion of the full phone number.

Without an address hierarchy, backbone routers would be forced to store route table information of the path reaching to every network in the world. With the present day huge number of routes, it is not practical to manage route tables and updates for so many routes. With the phenomenal growth of Internet, the non-hierarchical nature of the original IPv4 address space proved inadequate. This has been improved by use of Classless InterDomain Routing (CIDR) but legacy address assignments still hamper routing within the Internet. These address assignments limit both local and global levels of internetworking ([12]). CIDR uses bit masks to allocate a variable portion of the 32-bit IPv4 address to a network, subnet, or host. CIDR permits "route aggregation" at various levels of the Internet hierarchy, whereas Backbone Routers can store a single route Table entry that provides reachability to many lower-level networks. CIDR does not gurantee an efficient and scalable hierarchy. In order to avoid maintaining a separate entry for each route individually, it is important for routes at lower levels of the routing hierarchy, that have longer prefixes, to be summarized into fewer and less specific routes at higher levels of the routing hierarchy. Legacy IPv4 address assignments that originated before CIDR and the current access provider hierarchy often create problem in summarization. The lack of uniformity of the current hierarchical system, coupled with the rationing of IPv4 addresses, makes the whole business quite complicated. These issues affect all types of transactions. Moreover, in case of changing from one ISP to another, renumbering of IPv4 sites is extremely complicated. That is where IPv6's ease of use comes into picture.

3.1 Removal of special cases

When an enterprise cannot summarize its routes effectively, as it cannot present globally unique addresses to the internet, it may deploy private, isolated address space, which is non-unique but is not visible to the Internet. Users with such situations, typically use gateways and Network Address Translators (NATs), to manage their connectivity to the outside world. This blocks the integration of internal addresses with the global Internet addresses. This ad-hoc solution provides convenient connectivity between the enterprise and the Internet addresses of the present IPv4

world, but this is not suitable as a general solution and particularly so, if full robust connectivity with the outside is required. NAT translators also run into trouble when applications embed IP addresses in the packet payload, as is the case for FTP programs, Mobile IP and the Windows Internet Name Service (WINS) registration processes. Today's hierarchy of limited and poorly allocated IPv4 addresses has already caused problems and will continue to do so as more devices are getting connected to the Internet.

3.2 Address Allocation

The first field of an IPv6 address is the variable-length format prefix. (For details on different prefixes refer to ([1]); ([2])). This identifies an address as belonging to a particular category.

3.3 Address Categories

IPv6 addresses belong to one of three categories ([8]) :
• Unicast: Point to point addresses. An address identifies exactly one interface. A packet sent to a unicast address is delivered to that interface.
• Multicast: An address identifies a group of stations (or interfaces). A packet sent to a multicast address will be sent to all members of that group. However, multiple copies of the packet will be made only at the last possible router. Till then a single packet will be sufficient.
• Anycast: This address also identifies a group of stations but a packet sent to an anycast address is normally sent to only one point, the nearest member of that group.

3.3.1 Unicast Addresses

Unicast addresses can be structured and allocated in a number of ways. IPv6 defines two basic formats, provider-based and special address formats. Table 6 shows Provider-based address format.

Provider-based addresses are composed of the 3-bit prefix 010 followed by five variable-length components:

1. **Registry ID:** Identifies the registration authority, which assigns the provider portion of the address.
2. **Provider ID:** The ID of the Internet Service Provider, which is obtained from the registry.
3. **Subscriber ID:** Address of the subscriber, assigned by the provider.
4. **Subnetwork ID:** Uniquely identifies a subnetwork.
5. **Interface ID:** Uniquely identifies a station within a subnetwork.

Special address formats are of five types ([8]) :

Unspecified Addresses: This type of address consists of 16 null bytes, and can be used as a source address by a station that has not yet been configured with a regular address. Its address is 0.0.0.0.0.0.0.0 and it can also be used in control messages when the presence of an address is semantically required but no address is available. It shall never be used as a destination address.

Loopback Address: This address is written as 0.0.0.0.0.0.0.1 and may be used by a node to send an IPv6 datagram to itself (equivalent to 127.0.0.0/8 of IPv4). The loopback address and the unspecified address is never assigned to an interface.

Some parts of the global IPv4 address space have been reserved for special purposes. These blocks have been designated for such things as private internets and multicast traffic.

IPv4 based Addresses: By prepending a 96-bit null prefix we can construct an IPv6 address that is compatible with 32-bit IPv4 addresses.

Site Local Addresses: Identified by the site local prefix. These addresses cannot be routed on the global Internet. Their uniqueness is guaranteed only within a site. Organizations wishing to use the TCP/IP technology without being actually connected to the Internet can use this type of addressing.

Link Local Addresses: These addresses are defined only within a link. They can be used by stations connected to the same link or local area network.These packets are never sent through routers (equivalent to 169.254.0.0/16 of Link-local address in IPv4).

3.3.2 Multicast Addresses

IPv6 includes the capability to address a predefined group of interfaces with a single multicast address. A packet with a multicast address is sent to all members of the group. IPv4 did not originally support multicasting but the capability was added in the form of the Internet Group Multicast Protocol (IGMP) ([15]). However, in IPv6 the functionality for multicasting has been incorporated in the basic Internet Control Message Protocol (ICMP) ([16]), ([17]) itself.

Multicast Address Structure: Multicast addresses consist of an 8-bit prefix (all 1s), a 4-bit flag field, a 4-bit scope field, and a 112-bit group ID. Table 7 shows a Multicast Address Structure.

The first three flag bits are reserved (set to 0). The fourth bit is abbreviated T for transient, or one that is not permanently assigned.

The scope is encoded in a 4-bit integer, to limit the scope of the multicast group. This is to ensure that packets intended for local viewing do not leak out onto the global Internet.

Group Management: IPv6's version of ICMP, ICMPv6 (includes IGMPv6) ([17]), includes three group member-

Table 6: **Provider based address format**

010	Registry ID	Provider ID	Subscriber ID	Subnetwork ID	Interface ID

Table 7: **Multicast Address Structure**

11111111	flags	scope	Group ID
8	4	4	112

ship messages:
- Group Membership Query
- Group Membership Report
- Group Membership Termination

These are equivalent to the messages of IPv4's IGMP. To test membership of a station to a group, a router sends a membership query and stations that are members of a group respond by a group membership report. Group membership terminations are sent by stations that leave a group. (For more details on group membership message, refer to **([12]), pp 46 – 50**).

3.3.3 Anycast Addresses

Such an address enables a source to specify that it wants to contact any one node from a group of nodes via a single address. A packet with such an address will be routed to the nearest interface within a group, according to the router's measure of distance. A possible use of anycast address is to specify an anycast address in a routing header to indicate an intermediate address along a route. This address would refer to a group of routers within a particular subnet or for a particular provider. The packet would be forwarded to the nearest router. Usage of anycast addresses in this manner, naturally reduces the time spent by a packet on the network (if it is always sent to the nearest router) and improves routing efficiency.

Anycast addresses are allocated from the same address space as unicast addresses. Members of a group must therefore be configured to recognize the group address, and routers should be able to map an anycast address to a group of unicast addresses.

4 IPv6 Headers

As mentioned earlier, IPv6 specifies six extension headers:
- Hop-by-hop options header
- Routing Header
- Fragment Header
- Authentication Header
- Encrypted Security Payload
- Destination options header

IPv6 Headers must appear in the following order:
- IPv6 header: Mandatory, must always appear first
- Hop-by-hop options header
- Destination options header: For options to be processed by the first destination that appears in the IPv6 destination address field plus subsequent destinations listed in the routing header.
- Routing Header
- Fragment header
- Authentication header
- Encapsulating security payload header
- Destination options header: for options to be processed only by the final destination of the packet

4.1 Hop-by-Hop options header

This carries optional information that, if present, must be examined by every router along the path. Table 8 shows the fields of Hop by Hop Extension Header.

The next header field identifies the type of header immediately following this header. The header extension length field gives the length of the header in 64-bit units, not including the first 64 bits. The options field is variable-length and consists of one or more option definitions.

At present, only one option is present, the jumbo payload option. The payload length field in the IPv6 header is 16-bit, which means that the maximum size for normal packets is $2^{16} = 64KByte$. This option has a 32-bit option data field which gives the length of the payload in octets (header excluded). This allows a packet size of upto 4 billion octets. For a jumbogram, the payload length field in the IPv6 header must be set to 0 and there can be no fragment header.

4.2 Routing Header

This header carries a list of intermediate addresses through which the packet shall be relayed, a source route. Table 9 shows the fields of a Routing Header.

Fields:
- The next header, which identifies the type of header following immediately.
- The routing type used to identify the header format (current specification is only Type 0). For Routing Type 0, which is defined in RFC 2460, the routing type-specific data is a list of intermediate destination addresses **([18])**.

Table 8: **Hop by Hop Extension Header**

Next Header	**Hdr. Ext. Len.**	
One or more options		

Table 9: Routing Header

Next Header	**Routing type=0**	**Num Addrs**	**Next Addr**
Reserved	**Strict/Loose Bit Mask**		
Address[0]			
Address[1]			
.. ..			
.. ..			
.. ..			
Address[Num Address - 1]			

• The number of addresses in the list (Num Addrs).
• The number of the next address in the list (Next Addr).
• The Strict/Loose Bit mask to classify routing as strict or loose.

If the bit named Next Addr is set in the strict/loose bit mask, the station must check whether the next value in the list of addresses is the address of a neighbour. The packet may be forwarded to this address only if it is a neighbour, if not, it is rejected. If the bit is not set, the packet is forwarded regardless of the next address being a neighbour or not.

Another improvement IPv6 designers have sought to make, is to take away the burden of checking the source router field from all routers. In IPv4, all routers have to check the source router field even if they are only intermediate routers, not part of the explicit source route. In IPv6, a router checks the source router field only if it finds its own address among the destination addresses in the main header.

A station that recognizes one of its own addresses in the main header first checks whether the Next Addr value in the routing header exceeds the number of addresses in the list (Num Addrs). If it does, the packet has reached its destination and the station processes the next header. Otherwise it simply proceeds with source routing.

When using the type 0 routing header, the destination address in the IPv6 header is that of the first router in the path. The final destination is listed as the last address in the routing header's list of addresses. When a packet reaches the node identified in the IPv6 header, it either signifies the end of the routing process (as above) or the packet is forwarded and the destination address field in the IPv6 header and Next Addr field in the routing header are updated.

In order to return a packet to the sender, an IPv6 node must reverse the route in a packet it receives containing a routing header.

4.3 Fragment Header

In IPv6, fragmentation is allowed at source nodes only, not at routers. In IPv6, each subnetwork must support a Maximum Transmission Unit (MTU) of at least 576 bytes. In order to send packets larger than this, a node must perform a path discovery algorithm that enables it to learn the smallest MTU supported by any subnetwork on the path. It can then create fragments according to this value of the MTU. Otherwise all packets must have a fixed size of 576 bytes.

For each fragment, a fragment header is inserted between the basic IPv6 header and the payload. Table 10 shows the fields of a Fragment Header.

Fields:

• Next Header: Identifies the next header in the daisy chain.
• Fragment Offset: Fragmentation is supposed to occur on a 64-bit word boundary. IPv6 uses the most significant 13 bits of a 16-bit word to specify fragment offset, unlike IPv4 which used the least significant 13 bits. To obtain an octet offset in IPv6, the last 3 bits have to be set to 0, whereas in IPv4 the offset had to be multiplied by 8.
• Identification: Provides 32-bit packet identification.
• More fragment bit: Set to 1 for all fragments except the last fragment of a packet.

4.4 Destination Options Header

This contains optional information that, if present, is examined only by the packet's destination node. Table 11 shows the fields of a Destinations Options Header. An option is encoded as a variable number of octets.

The option type identifies the type of the option, and includes information about the action that must be taken if the processing node does not recognize the option. It also indicates whether the option may change en route, and the

Table 10: Fragment Header

Next Header	Reserved	Fragment Offset	Res	M
Identification				

Table 11: Destinations Options Header

Option Type	Opt Data Len	Option Data

option number itself.
Some options merely provide additional information on the packet or express preferences. If not recognized, they can be safely ignored by a processing node. Other options are critical and cannot be ignored, and the packet must be discarded.
The authentication header and encrypted security payload header are explained in the section on security.

5 Autoconfiguration

Autoconfiguration, or plug and play literally means that a machine will automatically discover and register the parameters that it needs to use, in order to connect to the Internet.

The autoconfiguration process includes creating a link-local address and verifying its uniqueness on a link, determining what information should be autoconfigured (addresses, or other information, or both) and in the case of addresses, whether they should be obtained through the stateless mechanism, the stateful mechanism, or both.

5.1 Link Local Addresses

As soon as an interface is initialized, the host can build up a link local address for this interface by concatenating the well-known link local prefix and a unique token, a number that is unique to the host on this link. A link local address can only be used on the local link.

5.2 Stateless Autoconfiguration ([19])

IPv6 nodes start initializing their behaviour by joining the "all nodes" multicast group. This is done by programming their interfaces to receive all the packets sent to the corresponding multicast address. The nodes then send a solicitation message (which is an ICMP message in IPv6) to all the routers on the link. The solicitation message must include the link layer address of the source, for example, its Ethernet or token ring address.

When the routers receive such a solicitation, they are supposed to reply with a router advertisement message. This is sent to the link layer address of the requestor. The router advertisement message (also an ICMP message) will contain several parameters that can be used both for autoconfiguration and for neighbour discovery.

The parameters used by the address configuration procedures are two control bits, M and O, and a prefixes option.
• Managed address configuration bit M: This is set to 0 when stations are authorized to perform stateless autoconfiguration, and set to 1 for stateful autoconfiguration (through address servers)
• Other configuration flag O: set to 1 when stations can carry out stateless address configuration, but require servers for obtaining other configuration parameters
• Prefix information option: There may be several prefix options in a message, each of which encodes one separate prefix. A prefix is encoded by a 128-bit address and a prefix length. When the M bit is null, (autoconfiguration is permitted) the station will examine the list of prefixes encoded as options.

Stateless configuration is based on unique tokens that are usually 48 bits long. Each address refers to a single Ethernet segment and this is not the best usage of address space. The whole IPv4 Internet uses 32-bit addresses. It also raises security issues. Stateless configuration is based on the principle that a user can simply plug in a new machine and it automatically configures itself within the network and runs. There is really no way of making sure that unauthorized personnel, who managed to carry a machine into the premises, could not do the same thing.

(For further information on Stateless Configuration refer to ([12]), pp 71–74).

5.3 Stateful Configuration

The routers' advertisements can specify that the host should use stateful configuration by setting up the managed configuration bit. If this bit is set, the host should contact an address server. Isolated hosts do not receive any router advertisement. They should also try to contact a local address server.

(For further information on Stateful Configuration refer to ([12]), pp 75- 76).

5.4 Stateless vs. Stateful Configuration

Stateless configuration is simple to use and does not require many servers. However, it has its disadvantages, namely inefficient address space usage and a lack of network access control.

Stateful configuration, while involving the expense and complexity of address servers, does allow easier enforcement of administrative controls.

A possible theory is to use stateless configuration largely for temporary address allocation but stateful configuration for permanent address allocation.

5.5 Address Resolution

The designers of IPv6 developed a neighbour discovery procedure that encompasses the functions of address resolution (the address resolution protocol of IPv4) as well as router discovery. The neighbour discovery protocol is defined as part of IPv6 ICMP.

The description of the neighbour discovery procedure assumes that the host maintains four separate caches:

- The destinations cache has an entry for each destination address toward which the host recently sent packets. It associates the IPv6 address of the destination with that of the neighbour toward which the packets were sent.
- The neighbour's cache has an entry for the immediately adjacent neighbour to which packets were recently relayed. It associated the IPv6 address of that neighbour with the corresponding media address.
- The prefix list includes prefixes that have been recently learned from router advertisements.
- The router list includes the IPv6 addresses of all routers from which advertisements have recently been received.

5.6 The Basic Neighbour Discovery Algorithm

In IPv6, Neighbor Discovery replaces the broadcast-based **A**ddress **R**esolution **P**rotocol (ARP), ICMPv4 Router Discovery, and ICMPv4 Redirect messages with efficient Neighbor Discovery messages.

- To transmit a packet, the host must first find out the next hop for the destination. The next hop shall be a neighbour, directly connected to the same link as the host. The host should then find a valid media address for that neighbour.
- If a packet has already been sent to that neighbour (as is usually the case), the neighbour address is found in the destination cache. If not, the host checks if a cached prefix matches the destination address. If a match is found, the destination is local, and the next hop is the destination itself. If no match is found, the destination is remote and the host should select a router from the table of routers and use it as the next hop.
- Once the next hop is decided, it is entered into the destination cache, and the neighbour's cache is looked up to find the media address of that neighbour. There are now four possibilities:

1. If there is no entry for that neighbour in the cache, the host should send a neighbour solicitation message. The neighbour is added to a new cache line whose status is set to incomplete.
2. If there is no entry for that neighbour, but its status is incomplete, the host should wait for the completion of the procedure to learn the media address and then send the packet.
3. If there is a complete line in the status of the neighbour, the media address is known and the packet can be sent immediately.
4. If the neighbour's entry in the cache has not been refreshed for a long time, its status is suspect. The media address can be used but a neighbour solicitation message should also be sent.

6 Security

By implementing security at the IP level, an organization can ensure secure networking, not only for applications that have security mechanisms but also for security-ignorant applications. IPv6 has been designed to provide end-to- end security, from source host to destination host.

IP-level security encompasses two functional areas: authentication and privacy. The authentication mechanism ensures that a received packet was in fact transmitted by the party identified as the source in the packet header. In addition, this mechanism ensures that the packet has not been modified in transit. The privacy facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties. The extension header provided in IPv6 for authentication is called the **A**uthentication Header (AH), and for privacy, the **E**ncrypted **S**ecurity **P**ayload (ESP) header ([20]).

6.1 Security Associations

An association is a one-way relationship between a sender and a receiver. If a relationship is needed for two-way secure exchange, then two security associations are required. A security association is uniquely identified by an Internet destination address in the IPv6 header and a security parameters index (SPI) in the enclosed extension header (AH or ESP header).

A security association is normally defined by the following parameters:

1. Authentication algorithm and algorithm mode being used with the IP AH (required for AH implementations).
2. Key(s) used with the authentication algorithm in use with the AH (for AH implementations).
3. Encryption algorithm, algorithm mode and transform being used with the IP ESP (for ESP implementations).

4. Key(s) used with the encryption algorithm in use with the ESP (for ESP implementations).
5. Presence/absence and size of a cryptographic synchronization or initialization vector field for the encryption algorithm (required for ESP implementations).
6. Authentication algorithm and mode used with the ESP transform, if any is in use (recommended for ESP implementations).
7. Authentication key(s) used with the authentication algorithm that is part of the ESP transform, if any (recommended for ESP implementations).
8. Lifetime of the key or time when key change should occur (recommended for all implementations).
9. Lifetime of this security association (recommended for all implementations).
10. Source address(es) of the security association (recommended for all implementations).
11. Sensitivity level eg. secret or unclassified referring to the protected data (recommended for all systems and compulsory for systems providing multilevel security).

6.2 Authentication Header

The AH provides support for data integrity and authentication of IPv6 packets. The presence of the AH will not change the behaviour of TCP or any other end-to-end protocol like UDP or ICMP. It will simply provide explicit insurance for the origin of the data. Table 12 shows the fields of an Authentication Header.

- **Next header:** Identifies the following header.
- **Length:** Length of authentication data field in 32-bit words.
- **Security Parameters Index:** SPI is the acronym for "Security Paramater Index". The combination of a destination address, a security protocol, and an SPI uniquely identifies a Security Association, described earlier. The SPI is carried in AH and ESP protocols to enable the receiving system to select the SA under which a received packet will be processed. Association is uniquely identified by the SPI and the destination address.
- **Authentication data (variable):** An integral number of 32-bit words.

6.2.1 Authentication Data

The AH is designed to protect the integrity of the entire datagram and is therefore calculated over the whole IP packet, to ensure that it has not been modified in transit. However, some fields have to be modified in transit. In the IPv6 header, the hop count is decremented at every hop. If the routing header is used, the IPv6 destination and next address are changed at every hop on the source route, and some hop-by-hop options are also allowed to change. In IPv4, the header checksum and time to live fields are subject to change. To avoid problems, the sender must prepare a special version of the message, independent of transformations in transit and then compute authentication data:

- In the IPv6 header, the hop count is set to 0.
- If the routing header is used, the IPv6 destination is set to the final destination, the routing header content is set to the value that it should have upon arrival, and the address index set accordingly.
- Options allowed to change in the hop-by-hop header are not taken into account in the checksum computation. The checksum is then computed using a cryptographic algorithm. RFC 1828 **([21])** specifies the use of MD5 algorithm for authentication.

6.2.2 Authentication using keyed MD5

The Authentication Header (AH) [RFC-1826] **([22])** provides integrity and authentication for IP datagrams. All implementations that claim conformance with the Authentication Header specification must implement this keyed MD5 [RFC 1321] **([23])**.

"Keyed MD5" is derived from the Message Digest 5 algorithm, MD5. MD5 computes a 128-bit checksum of the message (128-bit hash code) using non- linear transformations that make reverse engineering extremely difficult. Keyed MD5 operates by combining the message with a secret key and then computing the hash code on the result. The key is both prepended and appended to the message.

The secret authentication key, shared between the communicating parties should be a cryptographically strong number, not a guessable string of any sort. The system will support shared key of 128 bits or less. MD5's 128-bit output is 64-bit aligned. MD5 software speeds are adequate for common application but are slow for newer link technologies.

Calculation: First, the variable length secret authentication key is filled to the next 512-bit boundary, using the same pad length technique, defined for MD5. Then, the filled key is concatenated with the invariant fields of the entire IP datagram (variant fileds are zeroed), concatenated with (immediately followed by) the original variable length key again.

A trailing pad with length padded to the next 512-bit boundary for the entire message is added by MD5 itself. The 128-bit MD5 digest is calculated and the result is inserted into the Authentication data field. When the implementation adds, the keys and padding in place, before and after the IP datagram, care must be taken that the keys and/or padding are not sent over the link by the link driver.

6.3 Encrypted Security Payload

The authentication header does not transform the data, which remain subject to sniffers' attacks. When confidentiality is desired, the encrypted security payload is used.

Table 12: Authentication Header

Next Header	Length	RESERVED
Security Parameter Index		
Authentication Data (variable number of 32-bit words)		
(more authentication data)		

This is the last header in the daisy chain that remains visible after encryption is applied. Table 13 shows the Generic Format of the encrypted security payload.

6.3.1 Security Considerations

Users need to understand that the quality of the security provided is completely dependent on the strength of the MD5 hash function, the correctness of that algorithm's implementation, the security of the key management mechanism and its implementation, the strength of the key ([18]), ([21]), and upon the correctness of the implementation in all of the partcipating nodes.

Till date it is known that it is possible to produce collisions in the compression function of MD5 ([16]). There is not yet a known method to exploit these collisions to attack MD5 in practice, but this fact is nevertheless disturbing. It has also been determined ([21]) that it is possible to find two chosen text variants with a common MD5 hash value. However, it is unclear whether this attack is applicable to a keyed MD5 transform.

Although there is no substantial weakness for most of the IP security applications, it should be recognized that current technology is catching up to the 128-bit hash length, used by MD5. Applications requiring extremely high levels of security may wish to move to algorithms with longer hash lengths.

The ESP header has been designed so that only the security parameter index remains in clear text. Other parameters, such as payload type of encrypted data, are encrypted together with the data.

The precise format depends, in fact, on the algorithm used for encryption. The default algorithm suggested by the specification is the Cipher Block Chaining mode of the Data Encryption Standard (DES-CBC). Table 14 shows the Precise format of ESP header using DES-CBC.

- **Security Parameters Index:** Parameters for security association.
- **Initialization Vector (IV):** Composed of a variable number of 32-bit words. The precise number of words is randomly generated. The role of the IV is to ensure that the first words of the messages cannot be predicted. The randomness is propagated to the rest of the words by the Cipher Block Chaining algorithm.
- **Payload Data:** Actual encrypted data.
- **Padding:** This is added so that the message ends in a 64-bit word boundary. Padding octets may have any value.
- **Padding length:** Gives the number of padding octets used.
- **Payload type:** For example TCP, UDP.

6.3.2 Transport Mode ESP

This mode is used to encrypt the data carried by IP. This data is a transport layer segment, like a TCP or UDP packet. Table 15 shows the fields of Transport Mode ESP.

6.3.3 Tunnel Mode ESP

This is used to encrypt an entire IP packet including the destination address. For this mode, the ESP is prefixed to the packet and then the packet plus a trailing portion of the ESP header is encrypted. This method can be used to counter traffic analysis, which allows an intruder to measure volume of data flow between nodes even if the data is kept secret.

The IP header of the packet which is encrypted contains the destination address and routing information etc., so it is not possible to transmit the encrypted packet because the only unencrypted portion will be part of the ESP header. Intermediate routes would be unable to process such a packet. It is therefore necessary to encapsulate the entire block (ESP header plus encrypted IP packet) with a new IP header that will contain sufficient information for routing but not for traffic analysis. Table 16 shows the fields of Tunnel Mode ESP.

6.3.4 Key Distribution

The establishment of security association relies on the existence of secret keys known only to the members of the association. Efficient deployment of security will rely on an efficient key distribution method. Key management procedures are expected to not only provide the keys, but also the other parameters of the security association. One proposal for key distribution is called the design of Photuris and is based on zero-knowledge exchanges, completed by authentication of the exchanging parties.

6.3.5 The Design of Photuris

The Photuris algorithm is based on the zero-knowledge key exchange algorithm proposed by Diffie and Hellman.

Original Diffie-Hellman proposal: Two parties, Alice and Bob, agree on a prime number p and a generator g. Alice then picks a random number x. She computes the value:

$$n = g^x \bmod p$$

Table 13: Generic Format of the encrypted security payload

32-bit SPI
Encrypted Data and Parameters

Table 14: **Precise format of ESP header using DES-CBC**

Security Parameter Index (SPI)		
Initialization Vector (IV) (variable length)		
Payload Data		
Payload Data	Padding.....	
.....Padding	Padding Length	Payload Type

Table 15: **Transport Mode ESP**

IP Header	Other IP Headers	ESP Header	Transport-level Segment

Table 16: **Tunnel Mode ESP**

IP Header	Other IP Headers	ESP Header	IP Header plus transport-level segment

and transmits it to Bob. Bob in turn picks a random number y and computes the value:

$$\mathbf{m = g^y \bmod p}$$

and transmits it to Alice. At this stage, Alice knows m and x, Bob knows n and y while third parties may know m or n, but cannot obtain either x or y.

Alice and Bob can compute the session key:

$$\mathbf{z = n^y \bmod p = m^x \bmod p = g^{xy} \bmod p}$$

This key could then be used by the encryption or authentication algorithm. The advantages of a zero-knowledge algorithm are:

• Keys are computed when needed. There is no need to keep values secret for long periods. Experience shows that it is difficult to keep a secret very long in a computer.

• The exchange does not require any preexisting infrastructure such as key servers or certificate servers. It can thus be very easily deployed.

However the original Diffie-Hellman algorithm has some known weaknesses:

• It does not provide any information about the identities of the parties.

• It is subject to a person-in-the-middle attack in which a third party Trudy postures as Bob for Alice and as Alice for Bob. Both Alice and Bob negotiate a key with Trudy, who can then eavesdrop.

• It involves heavy computation since it requires very large values of p (about 1000-bit).

• The heavy computational load can be used in clogging attack, where an opponent requests a large number of keys. The victim spends too much time in useless computation.

The design of Photuris attempts to improve the situation in the following ways:

• Photuris supports authentication methods that can be coupled with key exchange. This can prevent the man-in-the-middle attack.

• Photuris uses a set of predefined primes. This avoids the computation required to discover long prime numbers.

(For more details on Photuris refer to **([10]; [12], pp 104 – 112)**).

7 Quality of Service: Support for real-time traffic, priority and flows

The IPv6 header has a priority field which enables the source to identify the transmit and delivery priority of each packet relative to other packets from the same source. There are two separate priority-related characteristics for each packet. A packet is first classified as being part of traffic for which the source provides congestion control or not, and is then assigned one of eight relative priority levels within the classification.

7.1 Congestion Controlled Traffic

This refers to traffic which the source reduces in response to congestion, for example TCP ([3]). If there is congestion in the network, TCP segments will take longer to arrive at their destination, and acknowledgments from the destination take longer, too. As a result, the source reduces the number of segments it generates. As congestion increases, packets may have to be discarded by routers. For a discarded packet, the acknowledgment does not arrive at all and it has to be retransmitted.

IPv6 defines the following categories of congestion-controlled traffic, in decreasing priority:

• **Internet control traffic:** Such as router updates etc. This is the most important traffic to deliver in times of congestion as it contains information about traffic conditions which is used to update routes and is used by network management to remove congestion.

• **Interactive traffic:** Online user to host connections. Here user efficiency depends on response time.

• **Attended bulk transfer:** Transfer of large volumes of data where the user is usually waiting for transfer to complete.

• **Unattended data transfer:** Transfer initiated by a user but not expected to be delivered instantly.

• **Filler traffic:** Expected to be handled in the background in the absence of higher-priority traffic.

• **Uncharacterized traffic:** If no guidance about priority is given, then lowest priority is assigned.

7.2 Non-congestion controlled traffic

This is traffic for which a constant data rate and a constant delivery delay are desirable. For example, real time audio and video, for which it makes no sense to retransmit a discarded packet and the delivery flow should be smooth. For this type of traffic, priority is assigned on the basis of how much the quality will deteriorate in the face of some dropped packets.

There is no priority relationship between congestion controlled and non-congestion-controlled traffic. Priorities are relative only within each category.

7.3 Flow Label

The IPv6 standard defines a flow as a sequence of packets sent from a particular source to a particular destination for which the source desires special handling by the intervening routers. A flow is uniquely identified by a combination of a source address and a nonzero flow label. All packets that are part of the same flow are assigned the same flow label by the source.

From the source's point of view, a flow typically will be a sequence of packets generated from a single application instance at the source and having the same transfer service requirements. From the router's point of view, a flow is a sequence of packets that share attributes which affect how they are handled by the router, like path, resource allocation, discard requirements and so on. Routers may treat packets from different flows in different ways, such as by requesting different qualities of service from subnetworks.

Flows basically provide for a better quality of service that the user pays for. In traditional packet switches, when a packet is received, it is passed to the routing module, which then decides on which outgoing line the packet should be forwarded. If the line is available, the packet is forwarded immediately, and if not, it must wait in the queue.

In order to ensure higher quality of service for real-time communication, we must make sure that the service rate for real-time data is higher than the arrival rate (queuing theory terminology). To achieve this, we can have one queue for each real-time communication and one default queue for all packets.

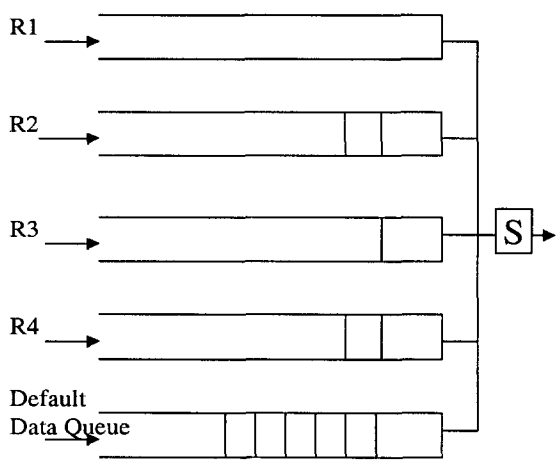


Figure 1: Four real-time flows and the default data queue.

In the Figure 1 we see four real-time flows. The flow label together with the source address is used to assert which packets belong to what flows. Packets not recognized as part of one real-time flow are part of the default data queue. It will never suffer from unpredictable queuing delays or experience congestion. The data queue will be served on a best effort basis and receive only whatever capacity is left after servicing real-time flows.

7.4 Quality of Service (QoS):

A flow is described using a token bucket and given the description of a flow, a service element (a router, a subnet etc.) computes various parameters, describing how the flow's data will be handled. By combining the parameters from the various service elements, the maximum delay a piece of data will experience, when transmitted via that path, can be established ([24]).

To achieve a bounded delay requires that every service element in the path supports guranteed service in its backbone and provide guranteed service between customers. Because a delay bound is produced, it has two parts: a fixed delay (transmission delays) and a queueing delay. The fixed delay is a property of the chosen path, which is determined not by guranteed service but by the set up mechanism. Only queueing delay is determined by guranteed service. And the queueing delay is primarily a function of two parameters: the token bucket and the data rate (R) the application requests. These two values are completely under the applications' control. An application can usually accurately estimate, a priori, what queuing delay guranteed service will likely promise. Additionally, if the delay is larger than expected, the application can modify its token bucket and data rate in predictable ways to achieve a lower delay.

7.5 Resource Reservation Protocol (RSVP) and (Flows)

The key assumption of RSVP **([12], pp 129 – 130)** is that resource reservation will mostly be needed for multicast applications like high-speed video transmission. Such applications usually have a large number of receivers that may be experiencing very different transmission conditions. RSVP is therefore a receiver driven protocol. Receivers decide from which source they want to receive and how much bandwidth they want to reserve and pay for.
The protocol:

- Sources enable reservation by regularly sending PATH messages alongside regular data packets.
- Routers learn about ongoing communications through these messages.
- Receivers specify the source from which they want to receive, bandwidth etc. by sending RSVP messages on the network.
- These messages are sent on the reverse path marked by PATH messages so that resources are reserved on the same link that is used to propagate data.

8 Support for Mobile Computing

Without specific support for mobility in IPv6, packets destined to a mobile node (host or router) would not be able to reach it while the mobile node is away from its home link (the link on which its home IPv6 subnet prefix is in use), since routing is based on the subnet prefix in a packet's destination IP address. In order to continue communication in spite of its movement, a mobile node could change its IP address, each time it moves to a new link, but then the mobile node would not be able to maintain transport and higher-layer connections when it changes location.

8.1 Basic Theory of Mobile IP ([25])

IPv6 allows a mobile node to move from one link to another without changing the mobile node's IP address ([26]). A

mobile node is always addressable by its "home address", an IP address assigned to the mobile node within its home subnet prefix on its home link. Packets may be routed to the mobile node using this address regardless of the mobile node's current point of attachment to the Internet, and the mobile node may continue to communicate with other nodes (stationary or mobile) after moving to a new link. The movement of a mobile node away from its home link is thus transparent to transport and higher-layer protocols and applications.

In Mobile IPv6, mobile nodes make use of the enhanced features of IPv6, such as Neighbor Discovery and Address Autoconfiguration to operate in any location away from home without any special support required from its local router. Most packets sent to a mobile node, while away from home in Mobile IPv6 are tunneled using an IPv6 Routing header rather than IP encapsulation, whereas Mobile IPv4 must use encapsulation for all packets (For more details about Mobile IPv4 refer to **([25]); ([27])** — **([31])**).

While a mobile node is away from home, its home agent intercepts any packets for the mobile node that arrive at the home network, using IPv6 Neighbor Discovery rather than ARP as is used in Mobile IPv4.

8.2 Procedure for Mobile IP

A mobile node is always addressable by its home address, whether it is currently attached to its home link or is away from home. While a mobile node is at home, packets addressed to its home address are routed to it using conventional Internet routing mechanisms, in the same way as if the node was never mobile. A router on a mobile node's home link with which the mobile node has registered its current care-of address is called Home Agent. While the mobile node is away from home, the home agent intercepts packets on the home link destined to the mobile node's home address, encapsulates them, and tunnels them to the mobile node's registered care-of address. Since the subnet prefix of a mobile node's home address is the subnet prefix (or one of the subnet prefixes) on the mobile node's home link (it is the mobile node's home subnet prefix), packets addressed to it will be routed to its home link.

While a mobile node is attached to some foreign link away from home, it is also addressable by one or more care-of addresses, in addition to its home address. A care-of address is an IP address associated with a mobile node while visiting a particular foreign link. The subnet prefix of a mobile node's care-of address is the subnet prefix (or one of the subnet prefixes) on the foreign link being visited by the mobile node; if the mobile node is connected to this foreign link while using that care- of address, packets addressed to this care-of address will be routed to the mobile node in its location away from home. Among the multiple care-of addresses that a mobile node may have at a time (e.g. with different subnet prefixes), the one registered with the node's home agent is called its "primary" care-of address.

9 Conclusion

Three decades of experience and the urgent needs of future applications have helped to re-engineer the most popular computer communication protocol in the world. This effort has successfully addressed very significant issues such as the convergence of data, voice and video communication (voice and video are real-time traffic). The telephone exchanges of the future will surely be based on IP switching. The provision for flow labeling in IPv6 is a major innovation to support this, as it added real-time capabilities. An equally significant success deals with the excellent provisions for multicasting, a feature that will revolutionize the broadcasting of voice and video. Another major change provided is the support for security features such as authentication and encryption. Only at the tail-end of this long list should we add the provision for longer addresses which will enable the Internet to be scaled up beyond any possible requirement that can be foreseen for the next few decades.

Internet2 : Though this article is covering the basics of IPv6, the case of Internet2 is being mentioned here for the sake of completeness. As most people know, IPv6 is one of the protocols (along with the Internet Group Multicast Protocol IGMP, the Protocol for Mobile Computing, and others) that is being developed for the famous "Internet2", which should achieve speeds of several Gb/s. The Internet2 will consist of existing campus LANs, which are aggregated at high-speed Giga Points-Of-Presence (GigaPOPs). The GigaPOPs are then connected to the very high performance Backbone Service Network (vBNS). The Internet2 is currently in the testing stage at several universities in the United States and has forged links with the Abilene project launched by Cisco Systems and Qwest Communications. The infrastructure for an undersea fiber-optic network is currently being laid at locations around the world by the telecom start-up CTR Group Ltd. The venture, called Project Oxygen, is scheduled in three phases and the first phase has become operational in early 2000. When we talk of transitions to IPv6, in a wider context, we are talking of the transition to Internet2.

As a final consideration, we should remember that IPv4 has remarkable adaptibility and scalability. The Internet Protocol that has been around for so long can have a lot of capabilities added on to it. The problem of address shortage could be solved through Network Address Translation (NAT). NAT boxes would have allowed addresses that were not globally unique, as long as they were unique in a defined region, they could then have been translated into unique addresses, used for communication outside the region. Provisions for multicasting have been added to IPv4 already. Almost everything that IPv6 has been created to provide, could have been supported by IPv4, but it has staggered under the weight of a truckload of new applications. The re-engineered protocol has been created

with these applications in mind, and would handle them much more efficiently. IPv6 is a major step forward in computer communications. But we must face the reality that, with time, it too will become inadequate, and as technology advances, there will almost certainly be a Next Generation Internet in every human generation!

Acknowledgement

Authors would like to thank Dr. S. Ramani for his having initiated them in the arena of IPv6 Networks, Prof. S. S. Jha, Director, TIFR for providing the ambience, Prof. R. K. Shyamasundar, DEAN, School of Technology & Computer Science, TIFR for his active support and guidance, Aniruddha Sen, TIFR and Ms. Reshma Mehta, Sardar Patel College of Engineering for having spent numerous hours, helping in shaping up the paper. Sanyal would also like to take this opportunity to thank his friend Marcin for his kind help and the reviewers for their critical comments. But for all these help, this paper would not have seen the light of the day.

References

[1] Stallings W. [1996], IPv6: The New Internet Protocol, http://www.comsoc.org/pubs/surveys/stallings/stallings-orig.html#gen 3.

[2] Stallings W. [1998], IPv6: The New Internet Protocol, IEEE Communications Magazine, July, pp 96-108.

[3] Hinden R. [1993], Applicability statement for the implementation of Classless Inter-Domain Routing [CIDR]. Internet Engineering Steering Group. RFC 1517, September, http://www-isi.edu/in-notes/rfc1517.txt.

[4] Y. Rekhter, T. Li.[1993], An Architecture for IP Address Allocation with CIDR. Internet Engineering Steering Group. RFC 1518, September, http://www-isi.edu/in-notes/rfc1518.txt.

[5] V. Fuller, T. Li, J. Yu, K, Varadhan. [1993], Classles Inter-Domain Routing [CIDR]: an Address Assignment and Aggregation Strategy. Internet Engineering Steering Group. RFC 1519, September, http://www-isi.edu/in-notes/rfc1519.txt.

[6] Y. Rekhter, C. Topolcic. [1993] Exchangig Routing Information Accross Provider Boundaries in the CIDR Environment. RFC 1520, September, http://www-isi.edu/in-notes/rfc1520.txt.

[7] Chuck S., Understanding IP Addressing: Everything You Ever Wanted To Know. http://www.3com.com/nsc/501302.html

[8] Hinden R., Deering S. [1998], IP Version 6 Addressing Architecture, RFC 2373, July. http://www-isi.edu/in-notes/rfc2373.txt.

[9] S. Shenker, Xerox; C. Partridge, BBN; R. Guerin, IBM [1997], Specifications of Guranteed Quality of Service, RFC 2212, September. http://www.isi.edu/in-notes/rfc2212.txt.

[10] Mark Mentovai, IPv4 Address Space Allocations, Internet Networking, http://www.mentovai.com/network/ipv4-allocation.html.

[11] Introduction to IP Version 6, White Paper. http://www.microsoft.com/technet/network/ipvers6.asp.

[12] Huitema C. [1996], IPv6: The New Internet Protocol, 1st Edition, Upper Saddle River, NJ: Prentice Hall.

[13] Thomas S. A. [1996], IPng and the TCP/IP Protocols Implementing the Next Generation Internet, 1st Edition, John Wiley & Sons, Inc.

[14] Jon Postel, Ed.; DARPA [1981], Internet Protocol, DARPA Internet Program Protocol Specification, RFC 791, September. http://www.isi.edu/in-notes/rfc791.txt.

[15] J. Postel [1981], Internet Control Message Protocol (for IPv4), Darpa Internet Program Protocol Specification, RFC 792, September, http://www.isi.edu/in-notes/rfc792.txt.

[16] Douglas E. Comer [1999], Computer Networks and Internets, Prentice-Hall, Inc., pp 295-296.

[17] S. Deering [1989], Host Extensions for IP Multicasting, Appendix I, Internet Group management Protocol (IGMP) (for IPv4), RFC 1112, August,

[18] Deering S. and Hinden R.[1998], Internet Protocol, Version (IPv6) Specification, RFC 2460, December, http://www.ietf.org/rfc/rfc2460.txt.

[19] Thomson S., Narten T. [1998], IPv6 Stateless Address Autoconfiguration, RFC 2462, December, http://www-isi.edu/in-notes/rfc2462.txt.

[20] S. Kent, Atkinson R. [1998], Security Architecture for the Internet Protocol, RFC 2401, November, http://www.isi.edu/in-notes/rfc2401.txt.

[21] P. Metzger, W. Simpson [1995], IP Authentication using Keyed MD5, RFC 1828, August, http://www-isi.edu/in-notes/rfc1828.txt.

[22] Atkinson, R., [1995], "IP Authentication Header", RFC 1826, NRL, August. http://www.isi.edu/in-notes/rfc1826.txt.

[23] Rivest, R., "The MD5 Message-Digest Algorithm", RFC 1321, MIT and RSA Data Security, Inc., April 1992. http://www.isi.edu/in-notes/rfc1321.txt.

[24] A. Contra and S. Deering [1998], Internet Control Message Protocol for IPv6 Specifications [ICMPv6], RFC 2463, December, http://www.isi.edu/in-notes/rfc2463.txt.

[25] D. Johnson and C. Perkins,[1996], "Mobility Support in IPv6", ACM Mobicom-'96, ACM, Nov 1996, pp 27-37.

[26] Deering S. and Hinden R.[1998], Mobile IP Networks, IEEE Potentials.

[27] Perkins C, ed., [1996], IP Mobility Support, RFC 2002, October, http://www.isi.edu/in-notes/rfc2002.txt

[28] Perkins C., [1998] "Mobile IP : Design Principles and Practice", Addison-Wesley Longman, Reading, Mass.

[29] C. Perkins, [1997],"Mobile IP" IEEE Communications Magazine, Vol 35, No 5, pp 84-99.

[30] P. Bhagwat, C. Perkins and S. K. Tripathi, [1996], "Network Layer Mobility : An Architecture and Survey", IEEE Personal Communications, Vol 3, No 3, June, pp 54-64.

[31] C. Perkins and P. Bhagwat, [1993], "A Mobile Networking System Based on Internet Protocol [IP]", Proceedings Usenix Symposium Mobile and Location Independent Computing, August, Usenix Association, pp 69-82 .

[32] Dowden D. C. , Gitlin R. D. , Martin R. L. [1998], Next-Generation Networks, Lucent Technologies Journal, Lucent Technologies, December.

[33] Steve King, Ruth Fax, Dimitry Haskin, Wenken Ling, Tom Meehan; Bay Networks; Robert Fink, LBNL; Charles E. Parkins, SUN Microsystems; [19-January-1999], The Case for IPv6, Draft-IETF-iab-case-for-ipv6-04.txt; Internet Draft; January. http://onoe2.sm.sony.co.jp/ipv6/id/draft-ietf-iab-case-for-ipv6-04.txt.

[34] D. Borman, S. Deering, R. Hinden [1999], "IPv6 Jumbograms", Network Working Group. RFC 2675, August. http://www.faqs.org/rfcs/rfc2675.html.

Construction and application of hierarchical socioeconomic decision models

Marjan Krisper and Blaž Zupan
Faculty of Computer and Information Science
University of Ljubljana, Slovenia
Phone: +386 1 476 8388m Fax: +386 1 426 4647
E-mail: marjan.krisper@fri.uni-lj.si, blaz.zupan@fri.uni-lj.si

Keywords: socioeconomic models, socioeconomic development, decision support, hierarchical decision models, what-if analysis, comparative analysis, data visualization

Received: January 17, 2000

The article presents a utility of multi-attributed hierarchical modelling approach to represent, analyze and study socioeconomic processes. The models are based on criteria tree for which the expert specifies the utility functions. The specific advantages of the approach are structuring the problem domain, a relative ease to build the models and the existence of underlying tools for comparative and what-if type of data analysis. We use these tools to construct two socioeconomic models, one for assessment of country's knowledge infrastructure and the other one for assessment of quality of political and economic system. We demonstrate the utility of these two models through experimental application in the analysis of real-world data from Word Competitiveness Yearbook.

1 Introduction

A determined orientation of the developed countries to foster the growth of information infrastructure that will allow their transition to information society [7] shows that we are undergoing a period that will exert a decisive influence on their future development. This is also or even more true for the Central European countries like Slovenia, Czech Republic, and Poland where the change of the political, economic, and legal system is the basis for their gradual transition to a modern society and their prospective integration within European Union.

In order to monitor and evaluate such transition, compare countries' successfulness, and investigate for the alternative development scenarios, one may benefit from models that assess the value of country's system given a selection of its *observable criteria*. A well-known example of such approach has been carried out by International Institute for Management Development (IMD), a non-profit foundation from Lausanne, Switzerland. IMD systematically collects different criteria from over 40 world-wide countries (roughly one half of them being OECD members and another half being newly industrialized and emerging market economies), resulting in a yearly report called The World Competitiveness Yearbook (see, for example, [6]; in this article we will refer to as Yearbook).

Each Yearbook normally includes more than 200 criteria, of which about two thirds present measurable quantities (*e.g.* GDP, unemployment, *etc.*), while the other third is obtained from the Executive Opinion Survey. Different aspects of world competitiveness are described by eight *factors* (like Domestic Economy, Government, Finance, *etc.*) which are derived from observable criteria. To organize the criteria further, each factor includes several criteria subgroups — in this tree-like three-level structure (Figure 1), each criterion belongs to a single subgroup, and consequently to a single factor.

Factor and factor subgroups thus represent an aggregation of the observable criteria. The observable criteria are first scaled, and then weighted and summed to obtain the value of their corresponding factor subgroup (see [6] for details). Finally, factors are computed as the sum of their corresponding factor subgroups. The country's data is then analyzed by presenting country's rank when considering each of the criteria, subgroups or factors. The advantage of IMD's approach lies in the high number of quality criteria being gathered, and providing a simple two-level structure in which these criteria are aggregated and studied. The disadvantage, however, is that the criteria aggregation by means of weighted sum may be over-simplistic as it does not take into consideration any potentially more complex criteria interaction. Furthermore, IMD's evaluation procedure that assigns all measurable criteria an equal weight of 1 and all the survey criteria an equal weight of about 0.9 may be too restrictive as it would be expected that different criteria are differently important (relevant). And finally, the Yearbook is in a sense static and calls for a computer-supported environment that would allow an interactive use of underlying evaluation model, supporting the decision making in terms of what-if and comparative analysis.

Crucial to the utility of such computer-supported models is their ability not only to reach a valid and (hopefully) accurate conclusions, but also to explain why such conclusion were reached [11, 10]. The modeling methodology should provide grounds for explorative analysis of alternatives being evaluated, making the model and decision support envi-

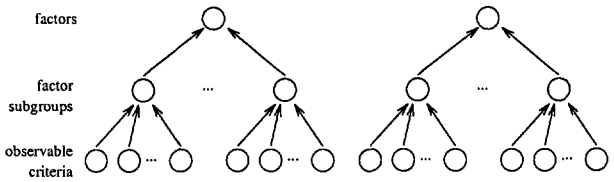


Figure 1: General systematization schema for criteria used by IMD.

ronment a valuable tool for decision expert. In these terms, classical numerical decision models that are based on criteria weighting [5] may be inadequate and pose problems where modeling a more complex interdependence of criteria is required [3]. This article builds on alternative approach for multi-attribute decision making that hierarchically organizes the criteria in the criteria tree and introduces new aggregate criteria. The aggregate criteria simplify utility function elicitation and play major role for explorative analysis. The approach was first proposed by Efstathiou and Rajkovič [8] and subsequently used in many applications, including the evaluation of R&D projects [3], evaluation of applications for nursery schools [14], priority ranking of applications for housing loans [1], portfolio analysis [9] and strategic planning [16]. In this article, we refer to its implementation in an expert system shell for decision support DEX [2].

Compared to IMD's three-level (criteria-subgroups-factors) criteria tree, we define models that have arbitrary number of layers, and refer to all internal nodes of the trees as *intermediate criteria* and the root of the tree as *target criteria*. Intermediate and target criteria are also referred to as *aggregated criteria*, as their value is computed from other underlying criteria rather than provided as an input to the model. The leaves of the trees represent criteria selected from those defined in the Yearbook — we refer to them as *basic criteria*. Using this terminology, the IMD's criteria subgroups are intermediate, and factors are target criteria. We propose two different models, one for *Knowledge infrastructure* and one for *System* target criteria.

The article is organized as follows. Section 2 introduces DEX paradigm for hierarchical multi-attribute decision models. DEX-based socioeconomic models for Knowledge infrastructure and the Quality of Political and Economic System are presented in Section 3. Section 4 illustrates the benefits of the DEX methodology through using the two socioeconomic models for tasks such as what-if and comparative analysis for the countries and data from the Yearbook, as well as for Slovenia — a country at the time of the writing of this paper not (yet) enlisted in the Yearbook but interesting since being a country in transition. Section 5 summarizes the results and concludes the article.

2 Hierarchical multi-attribute decision models

Hierarchical multi-attribute decision models as used by DEX consist of *criteria tree* and *utility functions*. Figure 2 shows a simple decision model — constructed only for illustrative purposes — to assess the quality of country's knowledge infrastructure from the quality of education, telecommunication network and computer deployment. Knowledge infrastructure (`ki`) is the *overall utility* or a *target criterion*, located at the root of the tree, that is modeled and derived from a set of *basic criteria* which are found at leaves of the criteria tree and which include the level of general education (`educ`), the quality of telecommunication network (`tel`) and the level of computer deployment (`comp`). The basic criteria are those that can be measured and/or obtained for specific country. The criteria tree also includes an internal node, which is an *intermediate criterion* that assesses the quality of technical infrastructure (`infr`). Both `ki` and `infr` are also referred to as *aggregated criteria*, as their value is determined from the values of other criteria in the criteria tree (*e.g.*, `infr` from `comp` and `tel`, and `ki` from `educ` and `infr`). The aggregated criteria are those that can not be directly observed or measured, but are besides the target criterion useful to be modeled. For the real-world problems, a criteria tree would include several aggregated criteria, depending on a complexity of the domain being modeled.

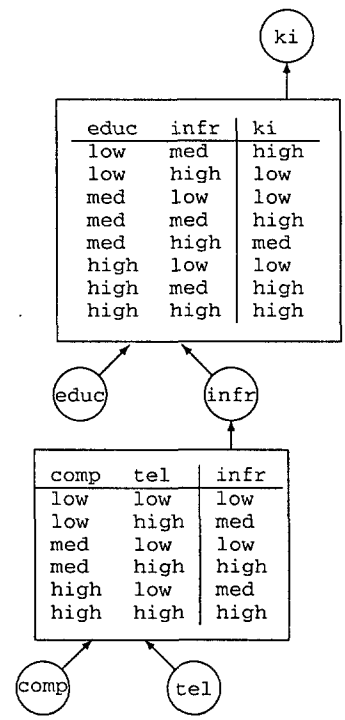


Figure 2: A simple decision model with three basic criteria (`educ`, `comp`, `tel`) and one intermediate criterion (`infr`).

DEX uses qualitative criteria, *i.e.*, every criterion in the criteria tree is assigned a finite value domain. In our case,

the value domain for ki, educ, and comp is {low, med, high} and the value domain for tel is {low, high}.

Utility functions are used to compute the value of aggregated criteria. DEX utility functions use so-called *if-then decision rules*, where each rule includes a specific combination of values for criteria entering the criteria function (the *if* part) and associated utility (the *then* part of the rule). These rules can then be represented with *utility table*. For example, in Figure 2, the utility function for infr specifies that when comp is low and tel is low, the value of the technical infrastructure is low (the first line in the utility table). Differently, when comp is med and tel is high, the value of infr is high (the fourth line in the utility table).

Within DEX, the rules in utility tables are defined manually, most often in a setup where a domain expert collaborates with a knowledge engineer. Once a sufficient number of the rules for some aggregated criteria have been entered, DEX assists in the elicitation of the new rules by proposing a viable set of values of the corresponding aggregated criteria. The complete process of defining the criteria structure and utility tables typically takes from one to five days, where a definition of criteria tree is often a more demanding task.

DEX models are evaluated from bottom up, starting at the aggregated criteria that depend solely on basic criteria to finally derive the overall utility. For our model from Figure 2, aggregated criterion infr is evaluated first based on criteria comp and tel, and then the overall utility ki is obtained from the values of educ and infr. For example, given the values of basic criteria for the two countries A and B from Table 1, the same Table shows the derived value of the intermediate criterion and the overall utility.

3 Socioeconomic models for knowledge infrastructure and the quality of political and economic system

Using DEX modeling paradigm, we have developed two different socioeconomic models, the first one modeling the level of the knowledge infrastructure (Knowledge infrastructure model) and a second one the quality of political and economic system with respect to their support of the economy and business (System model). Each model uses a separate set of basic criteria taken from the World Competitiveness Yearbook (Table 2).

The Knowledge infrastructure target criteria (KI) represents the level of development of knowledge infrastructure to support business and economic development. The KI model employs the criteria hierarchy as given in Figure 3. The model incorporates the utilization (IT_USAGE) and level of development of information technology (TEC_INFRA) and the quality of education (EDUCATION). The general education with regard to IT depends on computer literacy (C_LIT) and the overall quality of general education (GEN_EDUCATION). The development of technological infrastructure is estimated from diffusion of computers (C_INFRA) and the state of development of telecommunications (TELECOMM), which in turn depends on the current level and development potential of telecommunication infrastructure (TEL_INFR_INV) and accessibility and diffusion of telephones (TELEPHONES).

The quality of political and economic system in regard to their support of the economy and business is modeled as a target criteria SYSTEM. Its dependency on intermediate and basic criteria is outlined in a criteria tree shown in Figure 4. The value of the SYSTEM depends on the quality of government and economic system (QUAL_GOV_ECON), which aggregates the value of economic system and policies (ECONOMIC) and quality of government with respect to the support economy (GOV_QUALITY) and on the quality of politics and public trust (QUAL_POL). The later aggregates the values of quality of system and policies (POLITICAL) and the value of public trust to the current political system (TRUST). In its quality of government subtree, the model includes also the aggregated criteria that estimate the impact of lobbying (LOBBYING) and the government effectiveness and openness (EFFECT).

The knowledge infrastructure model uses 12, while the model for system uses 15 basic criteria. Each basic criterion has a domain of four values labeled "1" to "4", where "1" denotes the "worst" value of the criterion, *i.e.*, the one that has a negative influence to the value of the target criterion, and "4" denotes the "best" value of the criteria, again with respect to the influence to the target criteria. In this sense, the criteria values are nominal and ordered. The same domain definition was used for all aggregated criteria.

Together, the two models define 17 aggregated criteria. Presenting all utility functions defined is beyond the scope of this article, and for illustrative reasons we provide only an example. Consider thus one of the utility functions for knowledge infrastructure model that aggregates the value of educational system (EDUC_SYS) and in-company training (TRAINING) to the value of aggregated criteria for general education (GEN_EDUCATION, see Table 3). The utility function defines all 16 possible combinations of values for EDUC_SYS and TRAINING. For example, consider the rule number 7, which states that the value of general education level is 3 if the quality of educational system is 3 and in-company training is 2. We found that this pointwise definition of utility functions provides means to straightforward elicitation of knowledge from experts, since the experts find relatively easy to answer concrete questions (such as, "what is GEN_EDUCAT if the level for EDUCAT_SYS is 3 and TRAINING is 2"). Pointwise definition allows for defining non-linear functions. For example, in the function for GEN_EDUCAT the outcome never exceeds 2 if one of the input criteria (EDUC_SYS and TRAINING) has the value of 1. Nonlinearity in the aggregate function for GEN_EDUCAT is

	Basic criteria			*Aggregated criterion*	*Overall utility*
Option name	educat	comp	tel	infra	ki
Country A	med	low	low	low	low
Country B	high	high	low	med	high

Table 1: Evaluation results for countries A and B

Knowledge infrastructure	
MANAG_IT	Management of information technology: utilization of and familiarity with information technology by management
IT	Information technology: exploiting by companies
C_LIT	Computer literacy among employees
EDUC_SYS	The educational system: educational system meets the needs of competitive economy
TRAINING	In-company training: investing of companies in training of their employees
C_USE	Computers in use: share of worldwide computers in use
C_PC	Computers per capita: number of computers per person
INF_REQ	Infrastructure requirements, Telecommunications: Extend to which infrastructure meets business requirements
TEL_INFR	Telecommunications infrastructure
INVEST	State investments in telecommunications
TEL_LINES	Telephones: number of main lines in use per 1000 inhabitants
TEL_COST	International telephone costs

System	
INTERF	State interference: State interference does not hinder the development of business
SUBSID	Subsidies: Government subsidies are directed towards future winners
CONTROL	Control of enterprises: State control of enterprises does not distort fair competition in the country
IMP_PRACT	Improper practices (such as bribing and corruption)
EXTENT	Lobbying: Extent to which lobbying accelerates government decision making
INT_GROUPS	Lobbying by special interest groups
RESPONS	Government responsiveness: Ability to quickly adapt policies to new realities
DECENTRAL	Administrative decentralization: Decision-making independence of local/regional authorities from central government
PUB_SEC	Public sector contracts: openness to foreign bidders
POLICIES	Government economic policies: Extend to which government adapts its policies to new realities effectively
ADAPTATION	Political system: Extend to which political system is well adapted to today's economic challenges
TRANSPAR	Transparency of government towards citizens
POL_RISK	Political risk rating
GOV_POL	Government policies: Supporting by public consensus
SUPPORT	Public consensus and support for economic policies

Table 2: List of basic criteria used by Knowledge infrastructure and System model, respectively.

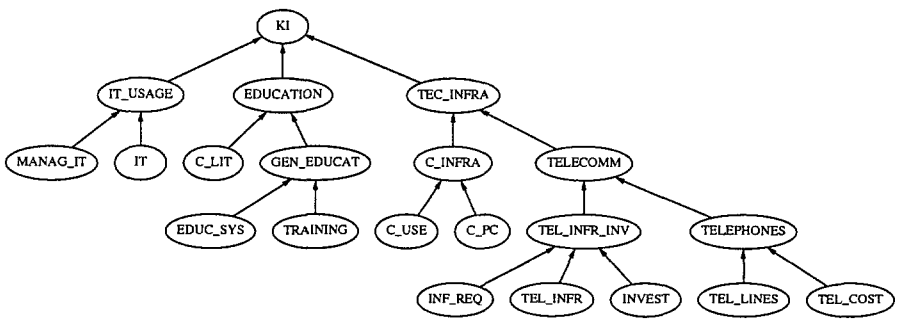


Figure 3: Criteria hierarchy for Knowledge infrastructure model.

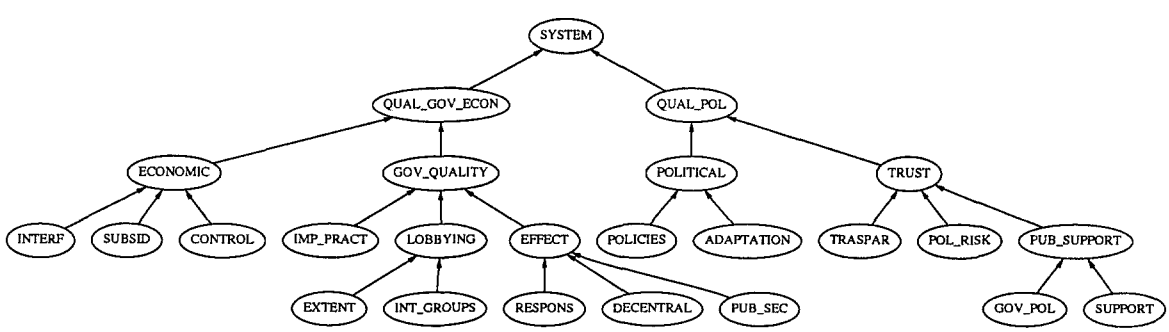


Figure 4: Criteria hierarchy for System model.

rule #	EDUC_SYS	TRAINING	GEN_EDUCAT
1.	1	1	1
2.	2	1	1
3.	3	1	2
4.	4	1	2
5.	1	2	1
6.	2	2	2
7.	3	2	3
8.	4	2	3
9.	1	3	2
10.	2	3	3
11.	3	3	3
12.	4	3	4
13.	1	4	2
14.	2	4	3
15.	3	4	3
16.	4	4	4

Table 3: An example of a utility function defined within the knowledge infrastructure model.

also evident from a graphical presentation of decision rules (Figure 5).

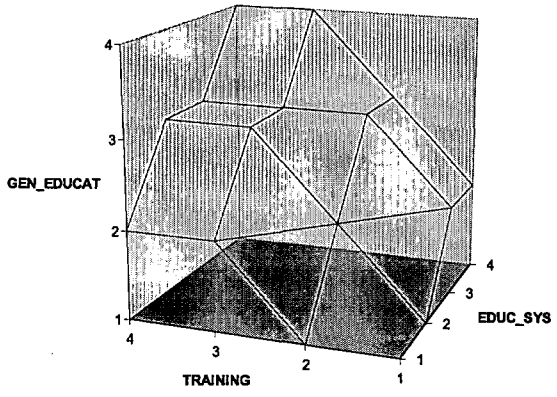


Figure 5: A graphical presentation of utility function from Table 3.

The pointwise definition of utility functions follows a case-based human way of thinking and as such implicitly states the relevance of each of the criteria. In practice, besides requiring linear relationships between input and aggregated criteria, eliciting explicit weights from the expert is usually a difficult task, as it forces the expert to think in more abstract way [2]. Note that not all of these were manually defined by expert, since DEX incorporates a mechanism that, based on the currently entered rules, provides suggestions for the rules not defined. In practice, we needed to define only about one half of the rules in utility functions for the two socioeconomic models — for the other half the expert most often accepted the suggestions provided by DEX.

4 Socioeconomic models in use

To demonstrate the applicability of the models defined in the previous section, we have first prepared the data set to be used. The models were built such that their set of basic criteria was taken from the list of criteria included in the World Competitiveness Yearbook (WCY, [6]). Obviously, the data of the countries included in WCY constitutes our basic data set.

For each of the criteria from WCY, the values were first ordered such that low values would potentially lower the model's outcome (final criterion) and that high values would increase it. Since DEX models require criteria to be qualitative (*i.e.*, "1", "2", "3", and "4"), the criteria values needed to be discretized. Discretization used quantiles, such that each resulting qualitative value would represent roughly the same number of countries for that criterion. Note that in this setup the qualitative values of criteria can also be interpreted as: "1" as "low", "2" as "below average", "3" as "above average", and "4" as "high".

The models developed can be used in a number of different ways. First, the models and their utility functions may provide additional insight to the domains. Next, the models can be used to evaluate the countries' data and derive corresponding values for aggregated and final criteria. The differences between two or more countries can then be studied by means of graphical comparison of criteria values. Finally, a specific country may be studied to see the effect of changing the values of basic criteria and studying its good and bad points.

4.1 Analysis of the model

The decision model as such can be analyzed locally by inspecting each of the defined utility function or globally by observing the overall impact of basic criteria on the target criterion. For the first task, DEX provides several tools. First, the utility functions can be visualized by selecting two input criteria and observing the output of aggregated function (see Figure 5 for an example). Another interesting DEX's tool is construction of aggregated rules from the set of elementary rules. For instance, an example of utility function that represents the function from Table 3 but is expressed by aggregated rules is given in Table 4. Note that instead of 16 there are just 9 rules required to define the aggregated criteria GEN_EDUCAT. Also, the utility function is much easier to comprehend. For example, from the last rule it is easy to see that GEN_EDUCAT can reach the highest value (4) only when EDUC_SYS is 4. Furthermore, the first two rules indicate that GEN_EDUCAT is 1 whenever one of the input criteria is 1 and the other less than or equal to 2.

We have further used both socioeconomic models to estimate the relevance of basic criteria to the value of the target criteria. For these, from each model a dataset was constructed that consisted of only basic criteria values and corresponding value of a target criterion. We have arbitrarily sampled each model with about 2000 such "data points", and then used the *information measure* (IM) score as defined in [13] to estimate the relevance. IM was originally used in recursive partition algorithms for decision tree induction to identify most appropriate (*i.e.*, important) criteria for decision tree nodes [15]. The criterion importance

agg. rule #	EDUC_SYS	TRAINING	GEN_EDUCAT
1.	1	≤2	1
2.	≤2	1	1
3.	≥3	1	2
4.	2	2	2
5.	1	≥3	2
6.	3	≥2	3
7.	≥3	2	3
8.	2:3	≥3	3
9.	4	≥3	4

Table 4: An example of aggregated rules for utility function from Table 3.

is assessed in independence of the other basic criteria: only the relationship with the target criterion is observed.

For the two socioeconomic models, the basic criteria are ranked according to their importance in Table 6. For knowledge infrastructure model, the three most important basic criteria are management of information technology, computer literacy and the value of education system. The three most important criteria from the System model are the control of enterprises, the level of state interference, and government subsidies. These results in general meet experts' intuitive expectations.

rank	criterion	IM
1	MANAG_IT	0.2200
2	C_LIT	0.1297
3	EDUC_SYS	0.0694
4	TRAINING	0.0618
5	IT	0.0576
6	TEL_LINES	0.0201
7	C_PC	0.0174
8	INF_REQ	0.0164
9	TEL_COST	0.0102
10	TEL_INFR	0.0097
11	C_USE	0.0043
12	INVEST	0.0033

rank	criterion	IM
1	CONTROL	0.1526
2	INTERF	0.1463
3	SUBSID	0.1404
4	IMP_PRACT	0.0438
5	POLICIES	0.0351
6	EXTENT	0.0263
7	PUB_SEC	0.0202
8	RESPONS	0.0186
9	ADAPTATION	0.0125
10	DECENTRAL	0.0092
11	INT_GROUPS	0.0059
12	GOV_POL	0.0054
13	SUPPORT	0.0052
14	POL_RISK	0.0019
15	TRASPAR	0.0012

Table 5: Ranking of basic criteria from Knowledge infrastructure (above) and System (below) model.

4.2 Comparative data analysis

The Knowledge infrastructure and System models were used to derive the value of the corresponding target criteria (KI and SYSTEM, respectively). Although DEX can be used for this task, another system called Vredana [17] was employed instead. Besides graphical presentation, the unique feature of Vredana is that it can evaluate each country not only to a single qualitative value of the target criterion, but can also estimate country's relative position within this range. For example, consider that the two countries having the values of EDUC_SYS and TRAINING 1 and 1 or 2 and 1, respectively, would both be classified to 1 for GEN_EDUCAT (see Table 3). In such case, Vredana would — within the qualitative value of 1 — rate the first country a bit lower than the second one by assigning a lower quantitative adjustment to the first country. In general, the gain of such rating is that Vredana allows further differentiation of the countries that were evaluated to the same qualitative rank. Since it is beyond the scope of this paper to further describe Vredana's evaluation algorithm, please see [4] for details.

Before presenting the results of comparative analysis, we needed to consider that the World Competitiveness Yearbook data we have used contains missing values. Both DEX and Vredana can properly handle these by deriving a range of values (probability distribution) for aggregated and final criteria. Although this is often a very desired feature, the requirement for the analysis in this section was that we required to unambiguously rank the countries and thus we needed crisp evaluation outcomes. For this purpose, the missing values were estimated as follows. If a country C was having a missing value for some criterion, we first found three other countries having most similar GDP/capita to the country C. Next, we replaced a missing criterion value of C with the average for this criterion over the three countries found.

Using the above introduced schema for handling missing values, the results of evaluation for both models are given in Figure 6. Note that in terms of the knowledge infrastructure, Finland, Sweden, Singapore, Hong Kong, and Germany rank the highest. It would be expected that USA would rank very high here, but additional analysis shows that it is ranked in class "3" because of the low value of general education. The specific comparison of Finland and USA that also highlights this deficiency is shown in Figure 7.

In terms of the quality of political and economic system in regard to their support of economy and business Figure 6 shows that Hong Kong, Singapore, and Malaysia — the Tiger countries — are the ones that rate the highest.

We have further explored the relation between country ratings of the two models by means of correlation coefficients. Three other ratings were used as well based on the following measures: GDP/capita, average value of criteria computers per capita (C_PC) and information technology (IT) from Knowledge infrastructure model (sel.KI), and average value of criteria government economic policies (POLICIES) and adaptation of political system (ADAPTATION) from System model (sel.SYSTEM). The correlation coefficients are given in Table 6. Note that completely correlated ranks would have a coefficient of 1, and

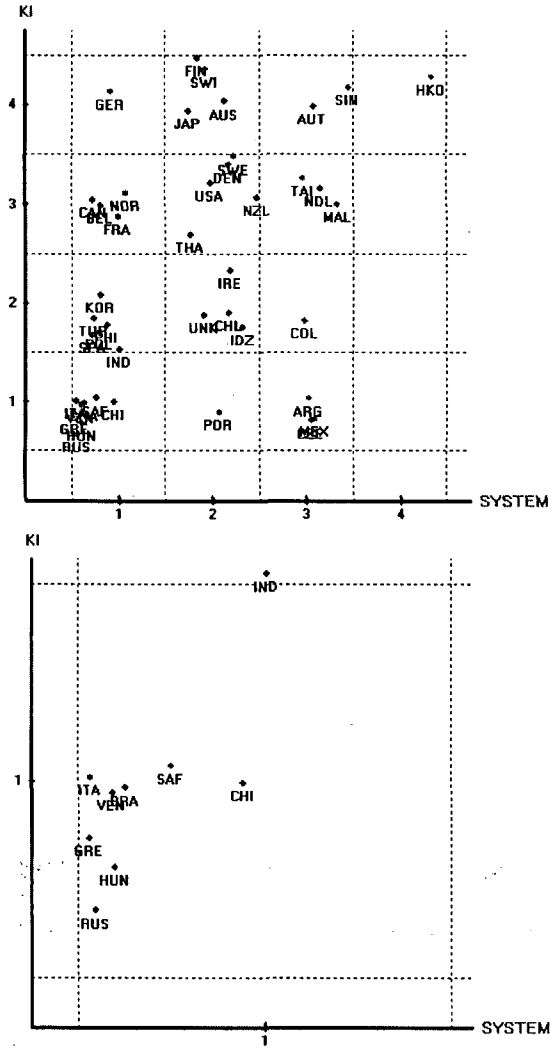


Figure 6: Vredana's graphical representation of the results of the evaluation for Knowledge infrastructure and System model (Figure on the right shows an enlargement of the quadrant KI=1 and SYSTEM=1).

Criteria	USA	Finland
KI	3	4
IT_USAGE	4	4
MANAG_IT	4	4
IT	4	4
EDUCATION	**2**	**4**
C_LIT	**3**	**4**
GEN_EDUCAT	**2**	**4**
EDUC_SYS	**2**	**4**
TRAINING	**2**	**4**
TEC_INFRA	4	4
C_INFRA	4	4
C_USE	4	3
C_PC	4	4
TELECOMM	4	4
TEL_INFR_INV	3	4
INF_REQ	4	4
TEL_INFR	4	4
INVEST	1	2
TELEPHONES	4	4
TEL_LINES	4	4
TEL_COST	3	4

Figure 7: Comparison of criteria for knowledge infrastructure for USA and Finland. The difference between the two countries (3 to 4) can be contributed to the differences of quality of the education (2 to 4), of which the subtree is printed in bold.

	SYSTEM	sel.KI	sel.SYS	GDP/cap
KI	0.452	0.822	0.201	0.658
SYSTEM		0.377	0.754	0.150
sel.KI			0.216	0.730
sel.SYSTEM				0.050

Table 6: Correlation coefficients for ranks obtained from the two socioeconomic models (KI and SYSTEM), selected basic criteria from each model (sel.KI and sel.SYS) and GDP/capita.

uncorrelated a coefficient of 0. The ranks of the two models are found weakly correlated (0.452). Not surprisingly, GDP/capita correlates better with Knowledge infrastructure than with System (0.658 > 0.150). As expected, the outcome of the two models best correlate with the ranks derived from the two averaged selected criteria, *i.e.*, sel.KI and sel.SYSTEM respectively.

A more focused ways of comparing the countries in Vredana in shown in Figure 8. The user selected four countries (Sweden, Austria, Poland, and Slovenia, and three criteria (political and economical system, and knowledge infrastructure) upon which these countries are compared. For Slovenia, the values of the basic criteria were estimated by local experts. For the analysis in Figure 8 we did not replace the unknown values, so one can observe that for Poland and Slovenia the minimal and maximal value for specific criteria is shown (for example, for Slovenia, the value of political system lies within 2 and 3). The radar charts show that there is a balance among knowledge infrastructure, political and economical system for the two highly developed members of the EU, whereas for the two associated countries in transition Poland and Slovenia it is evident that knowledge infrastructure and economical sys-

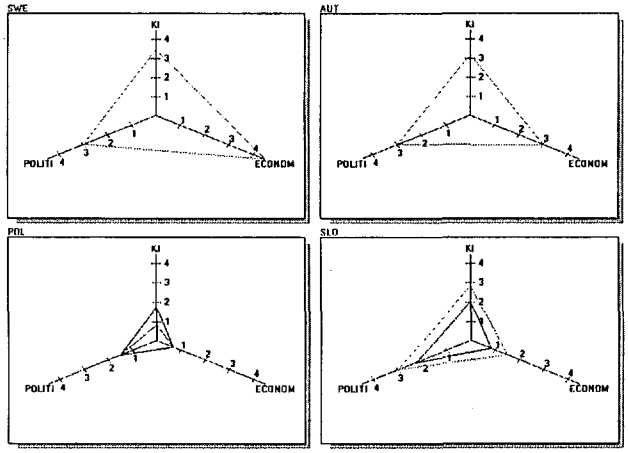


Figure 8: A snapshot of Vredana showing a radar chart that compares four countries with respect to three selected criteria.

tem do not follow yet the positive changes in political system.

4.3 What-if analysis

For an example of "what-if" analysis, we have studied Slovenia through the knowledge infrastructure model. Initially, Slovenia evaluates to "2-3" which ranks it into moderately developed countries in this respect. The question posed to "what-if" analysis is whether its knowledge infrastructure will be improved provided that Slovenia privatizes telecommunications. Namely, at present the Telecom Slovenia is the only telecommunication provider in the country (until 2001), thus holding a complete monopoly. The privatization of telecommunications in the European countries (including those in transition) boosted the development increasing both the quality of infrastructure and services. We have simulated such case and raised the values of basic criteria TEL_INFR and TEL_COST to 4. The two evaluated criteria trees, *i.e.*, the one for original data and the one with adjusted values due to privatization in telecommunications, are shown in Figure 9. According to the model, it is TEL_INFR whose improvement propagates through the intermediate criteria of telecommunication infrastructure all the way up to the target criteria, such that the new value of knowledge infrastructure is 3.

We have additionally attempted to change the value of management and information technology criteria (MANAG_IT). Increasing utilization and familiarity with IT by management is an already undergoing process, so we expect changes it this area in the near future. All other basic criteria being equal, raising MANAG_IT from "2-3" to "3-4" first results in increased IT_USAGE from 2-3 to 3-4, which finally results in improvement of knowledge infrastructure to 3 from previously 2-3.

Overall, we found the "what-if" analysis by DEX as exemplified above a very flexible and useful tool, especially as it provides the explanation through tracing of criteria tree of why and to what degree did the changes influenced the final score. This feature of model's transparency furthermore increases decision maker's confidence to the model and veracity of results.

Criteria	SLO	SLO*
KI	**2:3**	**3**
IT_USAGE	2:3	2:3
MANAG_IT	2:3	2:3
IT	3:4	3:4
EDUCATION	3	3
C_LIT	3:4	3:4
GEN_EDUCAT	3	3
EDUC_SYS	4	4
TRAINING	2	2
TEC_INFRA	**2**	**3**
C_INFRA	2	2
C_USE	1	1
C_PC	3	3
TELECOMM	**2**	**3**
TEL_INFR_INV	**2**	**3**
INF_REQ	2:3	2:3
TEL_INFR	**2**	**4**
INVEST	2	2
TELEPHONES	3	3
TEL_LINES	3	3
TEL_COST	**2**	**4**

Figure 9: Original (SLO) and modified (SLO*) evaluated criteria tree for Slovenia considering the pending changes in privatization of telecommunications. The differences are highlighted (criteria printed in bold).

4.4 Advantages and Disadvantages

Another feature of DEX that can support socioeconomic data analysis is the display of advantages and disadvantages for some selected country. Advantageous criteria are considered to be those that have especially positive effect to the value of the target criteria. Criteria that potentially most lower the final outcome are considered as disadvantages.

An example of advantages/disadvantages analysis for Japan using knowledge infrastructure model is shown in Figure 10. One can see that the major advantages of this country are in the area of usage of IT and in education, while the only disadvantage is the international telephone cost. Note that both disadvantages and advantages are shown as the criteria subtrees, so one can easily trace the propagation of positive (negative) effects through the criteria tree. For Japan, we can see that the advantageous criteria propagated all the way up in the IT usage and education subtrees, but these advantages were not strong enough to make the final outcome of maximal grade of the highest grade (the value of knowledge infrastructure evaluates to 3).

5 Conclusion

We have described the DEX paradigm to construction of hierarchical decision-support models and presented a case study to show how it can enable the efficient construction and application of socioeconomic models. In particular,

Advantages:	
IT_USAGE	4
MANAG_IT	4
EDUCATION	4
C_LIT	4
GEN_EDUCAT	4
EDUC_SYS	4
TRAINING	4
C_USE	4
INF_REQ	4
Disadvantages:	
TEL_COST	1

Figure 10: Advantages and disadvantages for Japan.

- DEX enables an efficient model construction that consist of identification of hierarchical structure and construction of rules for aggregated criteria. Since the original problem (mapping of many basic criteria to final criteria) is decomposed by introduction of intermediate criteria, the aggregation functions include only a few attributes and can be efficiently specified by means of pointwise rules elicited from the experts.

- The use of intermediate criteria not only decomposes a problem of model construction to simpler subproblems, but also makes these intermediate criteria observable — this is specifically useful in application of the model, since it can provide structured explanation and can ease the process of data analysis.

- Once the models are built, DEX can provide further inspection to aggregated functions by means of visualization and of presenting rules in an aggregated way. Moreover, the models can be used to study the overall relevance of the basic criteria.

- Data is provided to DEX models in terms of values for the basic criteria. DEX evaluates the data (derives the values of aggregated criteria) and can additionally be used to answer what-if questions, compare options (data for different countries), and structurally outline the advantages and disadvantages of each option.

- In addition to DEX, a Vredana tool can be used to visualize the data and compare the options.

When constructing and applying the socioeconomic models for knowledge infrastructure and value of political and economical system, we found all of the above advantages of the DEX and Vredana approach very useful. Of specific help was Vredana tool, which we believe should be the tool of the choice for performing what-if analysis and comparative studies. The weakness of the proposed approach is the fact that DEX and Vredana are available only as a separate tools that communicate through common model and option definition data file. It is expected that ongoing work on their integration will not only make data analysis more efficient, but will enable a deeper analysis of the model, such that, for example, the effects of changing the aggregation rules on the values of aggregated criteria for some set of options (countries) could be immediately observed through visualization.

Another possible methodological improvement is a function decomposition technique [18] to model development. Namely, in the case where a dataset exists that gives the values of the target criterion for a number of combinations of basic criteria, the aggregated functions can be automatically induced from the dataset. This data mining approach can potentially shorten the model development time as well as maintain the integrity of the model with some preexisting classified data. A pilot study that used this framework for construction of knowledge infrastructure model is described in [12].

We have proposed two different socioeconomic models, first modeling the value of country's knowledge infrastructure and second modeling the quality of political and economic system in regard to their support of economy and business. There are of course many other interesting socioeconomic models that could be employed in drilling in the country's socioeconomic data, getting insight to its present state and constructing and evaluating its potential future development scenarios. In our further work, we plan to extend the existing and construct new socioeconomic decision support models and correspondingly extend the data base of basic criteria using different sources, including the Yearbook of International Institute for Management Development, World Bank, International Monetary Found, and Institute for Economic Research from Ljubljana, Slovenia. As proposed in this article, these models will be built, integrated in and applied to support decision making and data analysis within DEX-Vredana framework.

References

[1] M. Bohanec, B. Cestnik, and V. Rajkovič. A management decision support system for allocating housing loans. In P. Humphreys, L. Bannon, A. McCosh, and P. Migliarese, editors, *Implementing System for Supporting Management Decisions*, pages 34–43. Chapman & Hall, London, 1996.

[2] M. Bohanec and V. Rajkovič. DEX: An expert system shell for decision support. *Sistemica*, 1(1):145–157, 1990.

[3] M. Bohanec, V. Rajkovič, B. Semolič, and A. Pogačnik. Knowledge-based portfolio analysis for project evaluation. *Information & Management*, 28:293–302, 1995.

[4] M. Bohanec, B. Urh, and V. Rajkovič. Evaluating options by combined qualitative and quantitative methods. *Acta Psychologica*, 80:67–89, 1992.

[5] V. Chankong and Y. Y. Haimes. *Multiattribute decision making: Theory and methodology*. North-Holland, 1983.

[6] C. Décosterd, S. Garelli, M. Hediger, M. Linard de Guertechin, and C. Travers. *The World Competitiveness Yearbook.* International Institute for Management Development, Lausanne, Switzerland, 1996.

[7] J. Delor. Growth, competitiveness, employment: The challenges and ways forward into 21st century. White Book, 1997.

[8] J. Efstathiou and V. Rajkovič. Multiattribute decisionmaking using a fuzzy heuristic approach. *IEEE Trans. on Systems, Man and Cybernetics*, 9:326–333, 1979.

[9] M. Krisper, V. Bukvič, V. Rajkovič, and T. Sagadin. Strategic planning with expert systems based portfolio analysis. In *Proc. IITT International Conference EXPERSYS-91*, pages 283–288, Paris, France, 1991.

[10] M. Krisper and L. Sočan. Heuristic modelling of socio-economic scenarios. In Z. Kaltnekar, editor, *Proc. International Conference on Organisation and Information Systems*, pages 64–72, Bled, Slovenia, 1992.

[11] M. Krisper, L. Sočan, and T. Zrimec. Expert system for the measurement and forecasting of socioeconomic development. In *Proc. IASTED International Conference*, Geneve, Switzerland, 1987.

[12] M. Krisper and B. Zupan. Synthesis of hierarchical decision support models from socioeconomic data. In C. Bavec and M. Gams, editors, *Proc. Information Society Conference*, pages 60–63, Ljubljana, Slovenia, 1998.

[13] J. Mingers. An empirical evaluation of selection measures for decision-tree induction. *Machine Learning*, 3:319–342, 1989.

[14] M. Olave, V. Rajkovič, and M. Bohanec. An application for admission in public school systems. In I. Th. M. Snellen, W. B. H. J. van de Donk, and J.-P. Baquiast, editors, *Expert Systems in Public Administration*, pages 145–160. Elsevier Science Publishers (North Holland), 1989.

[15] R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[16] V. Rajkovič, M. Krisper, T. Sagadin, and S. Grubar. An expert systems-based portfolio analysis. In *Proc. IFIP Congress*, Madrid, Spain, 1992.

[17] A. Šet, M. Bohanec, and M. Krisper. Vredana: A program for evaluation and analysis of options in multiattribute decision making (in slovene). In F. Solina and B. Zajc, editors, *Prof. Fourth Electrotechnical and Computer Conference ERK'95*, pages 157–160, Portorož, Slovenia, 1995.

[18] B. Zupan, M. Bohanec, J. Demšar, and I. Bratko. Feature transformation by function decomposition. *IEEE Intelligent Systems & Their Applications*, 13(2):38–43, March/April 1998.

A concurrent implementation of the simulated annealing by the method of multiple trials

Agnieszka Debudaj-Grabysz
Zakład Oprogramowania, Instytut Informatyki, Politechnika Śląska
Gliwice, Poland

Keywords: concurrent computing, Message Passing Interface, simulated annealing

Received: November 22, 2000

It is known, that concurrent computing can be applied to heuristic methods for combinatorial optimization to shorten the time of computation. Among others there are parallel algorithms for the simulated annealing. These algorithms assume however, the processes running in parallel can interrupt each other to exchange information, when one of them detects an appropriate condition to be fulfilled. The interruption is not to achieve in distributed environments like MPI straightforwardly. Instead of a querying done every step, a novel idea of interval is introduced in the paper. The interval is a predicted, varying during the computation, number of steps between two consecutive moments, when exchange of information between processes is needed. In this way, there is less idle probing for messages to come.

1 Introduction

The simulated annealing is one of heuristic methods to solve combinatorial optimization problems [5]. The optimal state, i.e. described by a maximal or minimal value of the cost function, is searched for in the method. It is achieved by comparing a current solution with another one, randomly selected from a neighborhood. In order not to converge to a local optimum, transitions to worse solutions are accepted with some probability, which is the most prominent feature of the method. The probability of the acceptance of a deteriorated solution is influenced by two factors. First, it goes down during the annealing together with the falling down of a control parameter called — by the analogy to the real annealing — temperature. Secondly, it is inversely proportional to the growth (further we will investigate only minimization) of the cost function. Techniques to select the neighborhood, compute the probability and control the temperature (the cooling schedule) are parameters of the method.

In concurrent computing separate runs of the program (jobs), fragments of the program (tasks), single instructions or even parts of instructions are carried out simultaneously by separate processors or independent computing units of a processor [4]. The development of concurrent versions of algorithms and performing the computing concurrently are motivated by the demand to get results faster. The present work concerns the version of the simulated annealing algorithm with the concurrency at the task level, i.e. parts of the program run in parallel.

The delivery problem has been chosen as a subject of optimization. The solution means here to minimize the length of the route, one has to cover traveling from a depot to consecutive cities. Formally the problem can be stated as follows [3]: for a given set C of n cities, where the cost $f(S_i)$ is defined for every subset S_i and the function o having values from the two-element set $\{0, 1\}$ is known, find such a set of subsets $\{S_i\}$, that their intersection is the empty set, their union equals C, $o(S_i) = 1$ for every S_i and the sum of $f(S_i)$ is minimal. The present work is limited to the case, where the function $o(S_i)$ equals 1, when the number of elements of the subset is less than or equal to 3, which means the supplier is allowed to visit no more than 3 cities during a single journey.

2 Parallel techniques of the simulated annealing

The authors of [1] set apart two general strategies to apply parallelism to the simulated annealing. They call them single and multiple trial parallelisms. The trial is the sequence of the following actions:

1. a random selection of a new solution from the neighborhood,
2. calculation of the value of the cost function for the new solution,
3. acceptance or rejection of this solution,
4. setting the new solution as the current one in case of the acceptance.

The single trial method applies a functional decomposition. The computational effort to perform a trial is divided

into subtasks and mapped to separate processors. Unfortunately in practice only few problems can adopt this strategy efficiently. It is why the multiple trial method is more popular. In this method whole trials are carried out by independent processors. This strategy will be now described in details.

2.1 The multiple trial method

The formal definition of the multiple trial strategy can be found, among others, in the work [2]. It is assumed, that p processors, denoted $P_1, P_2, \ldots, P_p$, are available and that they can work in parallel. At a given moment of time i, the annealing process creates a configuration X_i, being a member of the solution space E ($X_i \in E$). Every processor P_j performs a trial and generates a proposed configuration $Y_{i,j} \in E$. The new configuration X_{i+1} of the process is chosen as

$$X_{i+1} = Y_{i,k} \quad \text{for } k = \inf\{j = 1, 2, 3, \ldots, p | Y_{i,j} \neq X_i\} \tag{1}$$

So whatever solution among proposed and accepted ones is picked, provided it is different from the current solution. If there is no such solution, then the current solution remains unchanged.

The described strategy is efficient especially in low temperatures, when the acceptance of the proposed solution is rare. The increase of the number of processors allows to improve efficiency of the algorithm, understood as a ratio between the number of investigated configurations and the number of the accepted ones. It is quite in contrary to the high temperatures, where as an assumption a new configuration is accepted very often. So the number of processors could be lesser. Roussel-Ragot and Dreyfus [2] took advantage of this observation in practice. The details of their implementation are explained in the next section.

2.2 The parallel multiple trial method with a division of the temperature range into low and high temperatures

Roussel-Ragot and Dreyfus assumed that:

- the number p of processors to be used in the simulation is constant,
- in every step of the annealing the processors perform independent trials (moves),
- the acceptance coefficient $\chi(T)$ is defined as a ratio between the number of the accepted moves and the number of the investigated ones,
- two subranges of temperature are distinguished: range of high temperatures, where $\chi(T) > 1/p$ — here at least one move among all moves done by p processors in a single step is accepted and range of low temperatures, where $\chi(T) < 1/p$ — in this range statistically less than one move is accepted in a single step.

In high temperatures after having done one move by every processor, they are synchronized and one move is randomly accepted. The chosen configuration is broadcast to all processors and becomes the initial configuration for the next step.

In low temperatures the processors work asynchronously, communicating only when a processor accepts a move. Then the processors are synchronized and the accepted configuration broadcast.

The shown strategy allows to avoid useless communication in the low temperatures, improving so the overall efficiency of the algorithm.

2.3 The parallel multiple trial method in MPI environment

Roussel-Ragot and Dreyfus implemented the shown algorithm for the shared memory architecture, making only a remark about possible porting to a distributed environment. MPI (Message Passing Interface) has been selected as the computing environment for the research described in the paper. Nevertheless the MPI standard comes up with no appropriate means to code directly the low temperature strategy. It is not assumed, that one process can interrupt another one and moments of communication have to be set in advance.

The implementation could be approximated by a querying. Processors would check in every step, whether a new solution has been accepted to broadcast. It would be however very inefficient. It would reduce the amount of sent data but would not eliminate the redundant communication, influencing the total run time. It is the reason to propose another method of communication, to be shown in the next section.

3 The parallel multiple trial method with the varying communication interval

The proposed method refers to the feature of the simulated annealing, that the number of accepted moves is lesser and lesser together with the decline of temperature, approaching null at the end. So the cooperating processors could be ordered to communicate every period of time, called further the interval I, which gets longer as the number of accepted moves decreases. The interval, measured in the number of investigated moves, depicts by how many steps the processors should communicate, because there is a substantial probability of the acceptance of a solution during this period. If in the given interval at least one of processors accepted a new solution, then it is broadcast to other processors.

The interval is changed at the moment, when the temperature is reduced. The length of the interval for the next (reduced) temperature is approximated according to the acceptance coefficient in the previous temperature. In every temperature the average of L_a/L_b moves is accepted, where L_a denotes the number of the accepted moves and L_b — the number of investigated moves. If there is p available processors working in parallel, then they should communicate every I investigated configurations, where I is defined as follows:

$$I = \frac{L_b}{pL_a} \tag{2}$$

In this way the whole range of temperatures is divided not into two subranges, but into number of subranges equal to the number of distinct temperatures. Every subrange has its own moments of communication, set by the assumed interval.

Having experimental results analyzed, it was observed, that the length of the interval changes in an irregular manner: it "jumps". When the interval was getting longer, as it should be during the annealing, then the acceptance of a few solutions in a subrange shortened substantially the interval for the next temperature. Usually such an enlarged acceptance reveals to be mere a temporary disturbance, having no influence on the convergence, but the shortening of the interval intensifies the communication, which in turn makes the time of running the algorithm longer. It is why the length of the interval calculated according to (2) is not taken directly. Instead the weighed average of two lengths: the newly calculated one and one from the previous stage is used:

$$I_{i+1} = \frac{2}{3}I_i + \frac{1}{3}\frac{L_b}{pL_a} \tag{3}$$

The length of the interval from the previous subrange is weighed twice as much as the newly calculated length for the next subrange. The ratio of two between the weights has been assumed arbitrarily. No investigation of the influence of the ratio on results has been carried out. After having this "smoothing" applied, the length of the interval ceased to be so sensitive to abrupt changes of the number of the accepted moves.

4 Experimental results

Prior to the development of the parallel version, the sequential one has been implemented. The sequential version served as a base to code the parallel program and is a reference for comparison. Experiments have been conducted for sets of 100, 200 and 500 cities, running a program for specific parameters several times. The concurrent version has been investigated for various number of processors. The test has been conducted on IBM RS/6000 SP (SP2) — the computing server of Wrocławskie Centrum Sieciowo-Superkomputerowe (Center for Networking and Supercomputers in Wrocław, a 15 node shared memory machine) and Sun Enterprise in Centrum Komputerowe Politechniki Śląskiej (Computer Center of Silesian University of Technology, 12 processors)

4.1 The sequential algorithm

The parameters of the sequential algorithm have been tuned partially according to the work [3]. Specifically:

The definition of the neighborhood: two solutions are neighboring if they differ on the position of one city (the city from one route is transferred to another route) or if they can be transformed into each other by a swap of two cities from two routes, where at least one route includes three cities.

The temperature range: (differently then in [3]) the starting temperature is a fraction of the initial cost. Detailed investigation showed, that the value of the initial temperature, when adjusted individually for sets of varying sizes is of a significant importance as for the quality of obtained results (the bigger difference of sizes, the bigger influence of the initial temperature) — see table 4.1.

The cooling schedule: the temperature varies according to the relation:

$$T_{i+1} = \alpha T_i \tag{4}$$

where α depends on the so called pattern temperature, which changes too:

$$\alpha = \frac{500 + 2T_w}{500 + 3T_w} \tag{5}$$

$$T_{w+1} = \alpha T_w \tag{6}$$

The pattern temperature is set initially to 1000. For such schedule there is 515 distinct values of temperature during the annealing.

The end of the algorithm: when $T_w < 1$.

The probability of the cost growth: the probability to accept the worse solution is prescribed by the formula:

$$P(\delta F) = \frac{T_i}{T_i + \delta F} \tag{7}$$

where δF denotes the cost growth.

The initial solution is generated as a random order of N cities in L_r routes (differently then in [3]), where

$$L_r = \begin{cases} \frac{N}{3} & \text{where } N\%3 = 0 \\ \frac{N}{3} + 1 & \text{where } N\%3 = 1 \\ \frac{N}{3} + 2 & \text{where } N\%3 = 2 \end{cases} \tag{8}$$

Test results for different numbers of cities and different initial temperatures are shown in the Table 4.1. The given numbers denote values of the cost function (the total length of routes) obtained in series of tests.

no. of cities	starting temp.	min	max	mean	deviation
100	cost/1k	1835.3474	1843.7628	1838.048	1.43
	cost/5k	1835.3475	1845.4676	1838.682	1.96
	cost/10k	1835.3475	1844.2694	1838.612	1.91
	cost/50k	1835.348	1845.945	1839.1	2.43
200	cost/1k	3504.4226	3536.8674	3524.484	6.13
	cost/5k	3499.124	3512.9392	3506.896	3.00
	cost/10k	3498.1535	3513.0773	3506.625	2.97
	cost/50k	3498.942	3513.752	3505.902	2.61
500	cost/10k	8373.873	8407.265	8520.743	8.05
	cost/50k	8343.395	8372.781	8364.725	5.47
	cost/100k	8352.041	8374.236	8363.176	5.26

Table 1: Minimal and maximal values of the cost function obtained with the sequential program for different starting temperatures (fraction of the initial cost), k means 1000

We can observe, that for an assumed cooling schedule, the setting of the initial temperature (controlled by the denominator of the cost of initial solution) influences the quality of the result and the influence is more substantial, when there are more cities in the set. In the case of 100 cities the variation is rather small. The same best solution has been found by tests carried out for every investigated denominator (1000, 5000, 10000, 50000). The maximal values found in a series of tests differ slightly by 0.1%.

Advantages of lowering the initial temperature are more visible for the set of 200 cities. We observe the better and better minimal solution in series corresponding to the increasing value of the denominator (1000, 5000 and 10000). The value of the worst solution found in a series decreases too. If we keep lowering the temperature (to 50000), then the minimal and maximal values in a series are slightly worse, but the mean value is even better and the deviation also goes down.

For 500 cities, changing the denominator from 10000 to 50000, we can see, that the value of the best solution decreases by 0.4%. Changing the denominator even more, we have the same phenomenon as in the case of 200 cities: the minimal and maximal values in a series worsen slightly, but the results are usually closer to the mean, which itself is better.

4.2 The parallel algorithm

The parameters for the parallel version were as for the sequential one (see section 4.1). The results of the tests performed for different numbers of cooperating processors and different initial temperatures are collected in Table 4.2. Because of problems to measure run time in multi-user systems, the shown values of time are the real times of the execution of the program, found by UNIX time command. The processes had the highest priority during the measurements. It reflects the situation, as if we were an exclusive user of the system.

The first observation concerns the run time. The graph in Fig. 1 shows the speed-up (defined as the ratio of the single processor run time and the time of the execution in the parallel system), increasing together with the number of used processors. The biggest speed-up of 2 is not huge, but let us mind, that for 500 cities the sequential run time is about 10 minutes. Halving it is a substantial saving. The second observation is, that the speed-up depends heavily on the initial temperature.

Relatively small values of the speed-up can be explained by two reasons:

1. The time of communication has been added to the run time of the sequential program. It is a cost, which reduces the profit of decomposition.

2. There are situations during the execution, when the work of applied processors is not used, because there are too many of them. It happens so in high temperatures, when

$$pL_a > L_b \tag{9}$$

It means, that too many solutions are accepted when only one of them can be selected and broadcast. This phenomenon intensifies when there are more cities.

More detailed conclusions can be drawn, investigating the length of the interval in consecutive temperatures (Fig. 2). It is visible, the length increases when iterations go on. Nevertheless, in spite of the smoothing, there are still remarkable fluctuations and we have to remember, the shorter interval means more communication.

We should find the reason of the fluctuations. A monitoring of the number of accepted moves in iterations, such as shown in Fig. 3 and 4, is helpful, because the length of the interval depends on this number. It is clearly seen, the graph reveals situation being far from perfect: the acceptance number should tend to null.

The number of accepted moves depends only on the parameters set for the sequential version, the parallelization has nothing to do here. It means, there is a need for further research on the sequential algorithm, especially on setting its parameters. There are pairs of graphs in Fig. 2, 3 and

no. of proc.	start. temp.	min	max	mean	dev.	quality	time [s]	speed-up
100 cities								
1	cost/5k	1835.3475	1845.4676	1838.682	1.96	0.181	23.346	1
2		1835.347	1845.512	1838.261	1.88	0.158	20.638	1.13
3		1835.347	1845.814	1838.458	2.14	0.169	17.753	1.31
4		1835.348	1845.222	1838.506	1.99	0.172	16.126	1.44
5		1835.347	1842.726	1838.477	1.82	0.17	14.616	1.59
6		1835.347	1844.287	1838.436	1.91	0.168	14.132	1.65
100 cities								
1	cost/10k	1835.3475	1844.2694	1838.612	1.91	0.177	23.612	1
2		1835.3475	1845.1201	1838.589	2.11	0.176	19.83	1.19
3		1835.3475	1845.287	1839.133	2.23	0.206	16.485	1.43
4		1835.347	1843.659	1838.749	1.73	0.185	14.637	1.61
5		1835.347	1846.58	1838.889	2.12	0.192	12.915	1.82
6		1835.348	1844.502	1838.687	1.98	0.182	12.404	1.90
200 cities								
1	cost/5k	3499.124	3512.9392	3506.896	3.0	0.222	92.796	1
2		3500.927	3513.472	3507.739	3.08	0.246	84.975	1.09
3		3500.151	3515.958	3507.771	2.97	0.247	70.439	1.31
4		3500.931	3513.499	3507.422	2.95	0.237	67.812	1.36
5		3500.862	3513.532	3508.002	2.612	0.254	63.683	1.45
200 cities								
1	cost/10k	3498.1535	3513.0773	3506.625	2.975	0.242	94.314	1
2		3500.944	3512.683	3506.715	2.8	0.245	83.286	1.13
3		3500.164	3513.22	3506.588	2.68	0.241	69.374	1.35
4		3498.271	3511.438	3506.062	2.87	0.226	64.092	1.47
5		3501.475	3513.205	3507.185	2.68	0.258	58.382	1.61
500 cities								
1	cost/10k	N/A	N/A	N/A	N/A	N/A	601	1
2		N/A	N/A	N/A	N/A	N/A	551	1.09
3		N/A	N/A	N/A	N/A	N/A	458	1.31
4		N/A	N/A	N/A	N/A	N/A	453	1.32
500 cities								
1	cost/50k	N/A	N/A	N/A	N/A	N/A	585.7	1
2		N/A	N/A	N/A	N/A	N/A	495.5	1.21
3		N/A	N/A	N/A	N/A	N/A	375.5	1.6
4		N/A	N/A	N/A	N/A	N/A	329.73	1.82
5		N/A	N/A	N/A	N/A	N/A	290.37	2.07

Table 2: Results of parallel executions. For 500 cities only monitoring of time has been performed

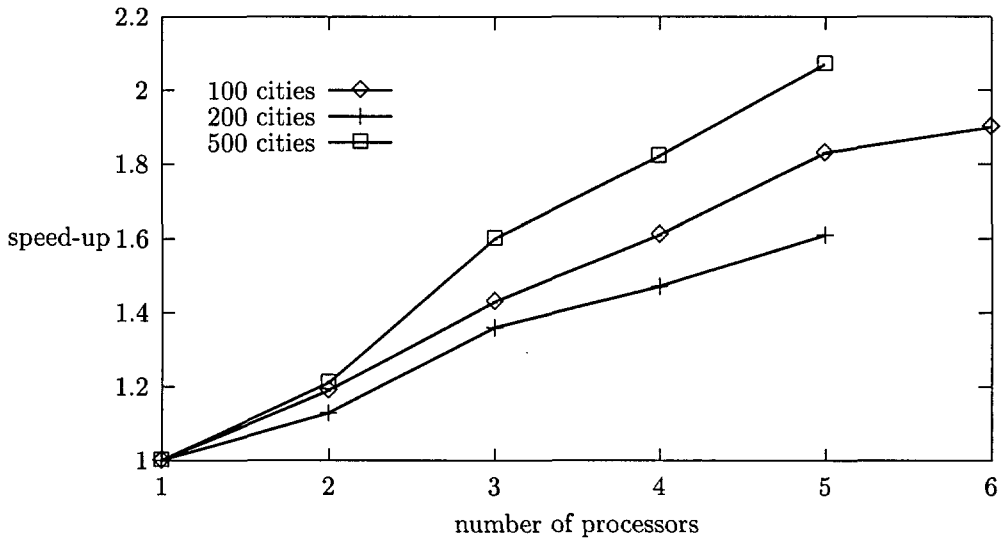


Figure 1: The relationship between the speed-up and the number of used processors

4, concerning two simulations for different initial temperatures. With no doubt we can notice, a too high initial temperature causes the increase of the accepted moves (Fig. 3 and 4) and shortening of the interval (Fig. 2). The shorten interval, the more often inter-process communication and the run time grows.

The third observation touches the quality of obtained results, defined as follows:

$$Q = \frac{f_{avg} - f_{min}}{f_{min}} \cdot 100 \qquad (10)$$

when f_{avg} and denotes the mean value of the cost in a series of parallel tests and f_{min} denotes the best result found by the sequential program. It can be stated, that the quality does not change with respect to the number of processors. Similarly the standard deviation, describing the scattering of the results, is nearly constant.

5 Conclusions

1. The MPI environment does not deliver appropriate methods to implement the concurrent algorithm of the simulated annealing, in the literature called the multiple trial method with a division of the temperature range into low and high temperatures.

2. The present work proposes a modification of the algorithm, rejecting the division into two subranges. Instead, for every temperature the number of trials necessary to find an accepted solution in the neighborhood is approximated. When the assumed number of trials is done, then processors communicate. The intention was to minimize the amount of the wasteful exchange of data in the environment, where processes communicate in prescribed moments or their work.

3. Observing the relationship between the run time and the number of employed processors, one can put a hypothesis, that the profit of the decomposition is substantially reduced by the cost of inter-process communication.

4. Because the partial problem to be solved between communications (i.e. the traveling salesman problem for 3 cities) is trivial, we can expect bigger speed-ups in the case of other problems, specifically in the case of the delivery problem, when the supplier is allowed to visit more then 3 cities. Then the burden of the primary computation should greatly exceed activities linked with the exchange of data, which according to the law of Gustafson [6] should positively influence the speed-up.

5. There is a need to tune the sequential algorithm in order to approach the ideal annealing process. The better parameters will improve the performance of the parallel version.

Acknowledgement

The work has been supported by the grant BK-245/RAu-2/99 of Polish State Committee for Scientific Research.

References

[1] Aarts E., Korst J. (1989) *Simulated Annealing and Boltzman Machines*, John Willey & Sons.

[2] Azencott R. (ed.) (1992) *Simulated Annealing. Parallelization Techniques*, John Willey & Sons.

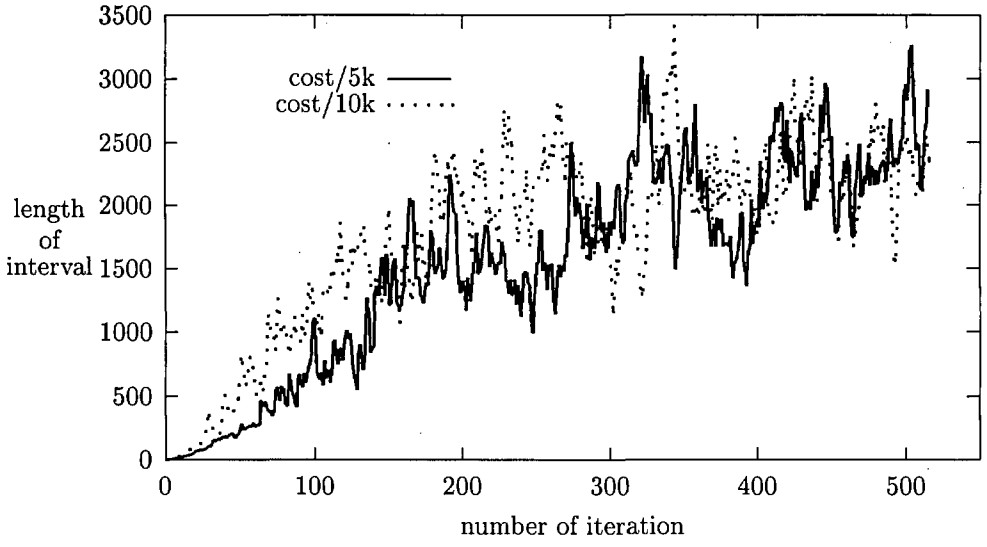


Figure 2: The length of the interval in consecutive iterations (100 cities, 2 processors)

[3] Cichoński S. (1999) *Sekwencyjne i równoległe algorytmy symulowanego wyżarzania*. MSc thesis, Politechnika Śląska, Gliwice.

[4] Foster I. (1995) *Designing and Building Parallel Programs. Concepts and Tools for Parallel Software Engineering*, Addison-Wesley.

[5] Reeves Collin R. (ed.) (1995) *Modern Heuristic Techniques for Combinatorial Problems*, McGraw-Hill.

[6] Wilson G.V. (1993) A glossary of parallel computing terminology. *IEEE Parallel & Distributed Technology*, Feb 1993, 52-66.

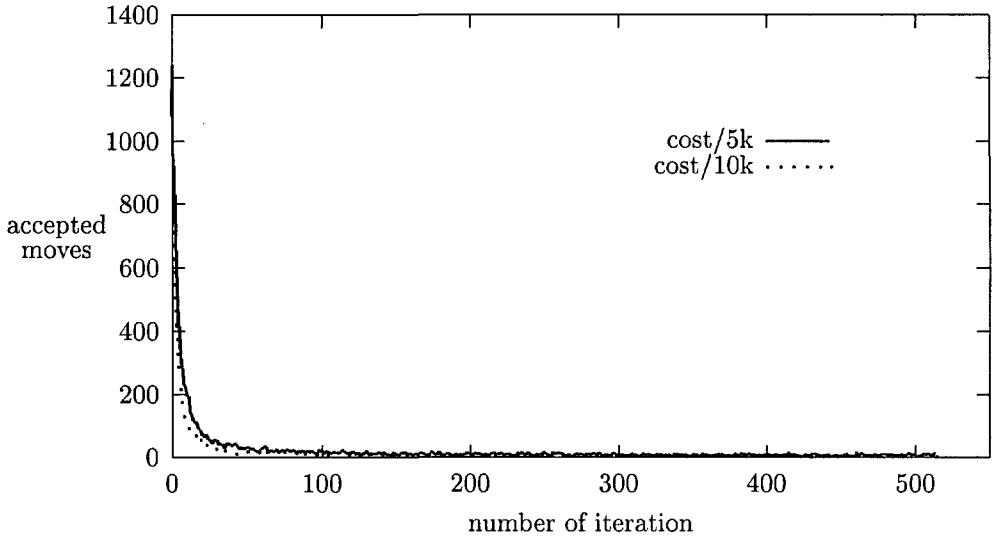


Figure 3: The total number of accepted moves in consecutive iterations (100 cities)

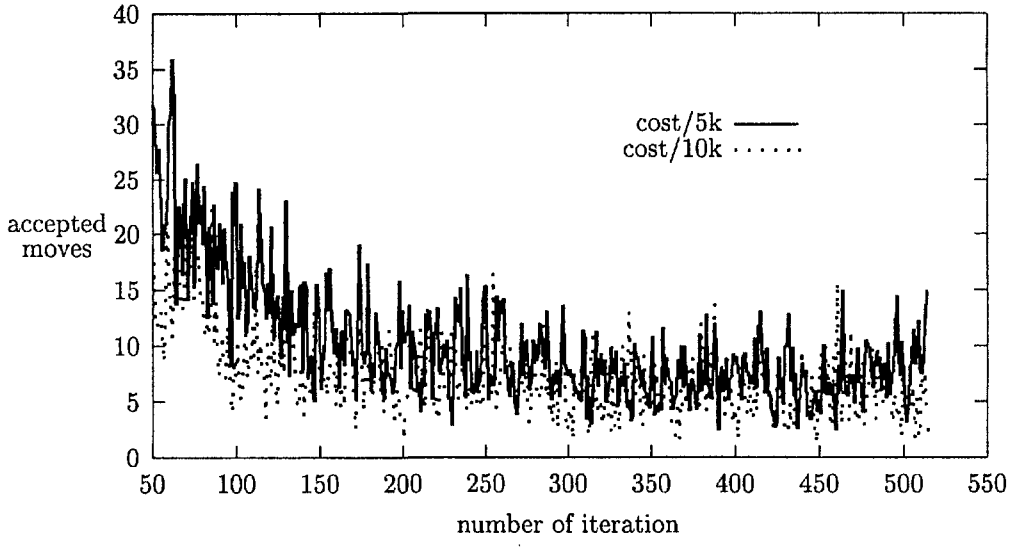


Figure 4: The total number of accepted moves with the first 50 iterations not shown

Efficient parallel clustering algorithms

Amitava Datta
Department of Computer Science & Software Engineering
The University of Western Australia
Perth, WA 6009, Australia
Phone : +61 8 9380 3449, Fax: +61 8 9380 1089
E-mail: datta@cs.uwa.edu.au

Keywords: parallel algorithm, pattern recognition, clustering problems, CREW PRAM

Received: February 23, 2000

Given a set P of n points in two dimensions, the aim of a clustering algorithm is to find a subset of k points such that some measure for this subset is minimized. We present efficient parallel algorithms for two clustering problems, namely, **(i)** *a k-point subset with minimum L_∞-diameter and* **(ii)** *a k-point subset with minimum L_∞-perimeter. We give a unified framework for solving these problems in parallel. We decompose the original problem into $O(n/k)$ subproblems, each of size $O(k)$ by imposing a grid on the point set. We solve all these subproblems simultaneously in parallel and obtain the global optimal solution from the solutions of the subproblems. For the L_∞-perimeter case, our algorithm runs in $O(\log^2 n)$ time and requires $O(n \log^2 n + nk^2 \log^2 k)$ work. For the L_∞-diameter case, our algorithm runs in $O(\log^2 n + \log^2 k \log\log k \log^* k)$ time and requires $O(n \log^2 n)$ work. For each problem, the work done by our algorithm is close to the complexity of the best known sequential algorithm. Our algorithms are designed for the CREW PRAM model of parallel computation.*

1 Introduction

Clustering problems are important in *pattern recognition* and *cluster analysis* [1, 14]. In this paper, we consider clustering problems of the following type. Given a point set P with n points, the problem is to compute a subset of k points such that some *closeness measure* is minimized. For an example, one may want to minimize the perimeter of the convex hull of the k-point subset. This measure was considered by Dobkin *et al* [10]. Aggarwal *et al* [3] considered closeness measures like diameter, side length of the enclosing square, the perimeter of enclosing rectangle, etc. They gave algorithms for these problems on the plane based on higher order Voronoi diagrams. Eppstein and Erickson [11] gave a general framework for solving k-point clustering problems based on computing rectilinear nearest neighbors for the points in the set P. The underlying strategy for their algorithms is that a k-point subset satisfying any of the closeness measures must be in the set of nearest neighbors for a particular point. The best known sequential algorithms for many of these problems were presented by Datta *et al* [9]. They improved upon the strategy of Eppstein and Erickson [11] by imposing a grid on the point set and solving the problems for smaller subsets of points. Efrat *et al* [12] have given sequential solutions for some of these problems using the parametric search technique of Megiddo [19]. Matoušek [18] has given a simple randomized algorithm for finding a smallest circle which encloses k points of a point set.

The applications in pattern recognition and cluster analysis typically involve large sets of data. In many situations, it is important to solve such problems fast for real time applications. The sequential algorithms may not be fast enough for such problems and hence it is important to design parallel algorithms for these problems. Currently, no parallel algorithm is known for any of the clustering problems. In this paper, we initiate the investigation of parallel complexity of these problems. Our model of computation is the *Parallel Random Access Machine (PRAM)*. We are interested in the *Concurrent Read Exclusive Write (CREW)* PRAM. In this model, two or more processors can read from the same memory location simultaneously, but only one processor can write to a memory location at a time. The details of this model are given in the book by JáJá [15].

In this paper, we give efficient parallel algorithms for two of the k-clustering problems for which the best known sequential algorithms are given by Datta *et al* [9]. Our algorithms are based on nontrivial parallelization of the algorithms in [9]. In the first problem, we compute the minimum L_∞-perimeter k-point subset (or the minimum perimeter axes-parallel rectangle that encloses k points). The sequential algorithm for the minimum L_∞-perimeter problem runs in $O(n \log n + nk^2)$ time. Our parallel algorithm for this problem runs in $O(\log^2 n)$ time and requires $O(n \log^2 n + nk^2 \log^2 k)$ work. The work done by our algorithm compares well with the time complexity of the best sequential algorithm and is away by a factor of $O(\log^2 n)$ when $k = O(n)$.

In the second problem, we compute a minimum L_∞-diameter k-point subset (or the minimum side-length axes-

parallel square that encloses k points). The sequential algorithm for the minimum L_∞-diameter k-point subset problem runs in $O(n \log n + n \log^2 k)$ time. We present a parallel algorithm which runs in $O(\log^2 n + \log^2 k \log\log k \log^* k)$ time and requires $O(n \log^2 n)$ work. So, the work done by our algorithm (processor-time product) matches the time complexity of the sequential algorithm in [9], when $k = O(n)$. For small k, the work done by our algorithm is only a factor of $O(\log n)$ away from the time complexity of the sequential algorithm.

The rest of the paper is organized as follows. In sections 2, we discuss the general strategy for solving the clustering problems in parallel. In section 3, we discuss how to construct a degraded δ-grid in parallel. In section 4, we present the algorithm for finding the minimum L_∞-perimeter k-point subset. And in section 5, we present the algorithm for computing the minimum L_∞- diameter k-point subset. We conclude with some comments and open problems in section 6.

2 The general strategy for solving clustering problems

Let P be a set of n points on the plane. The points in the set P are represented by $p_1, p_2, \ldots, p_n$. The x (resp. y) coordinate of the point p_i is represented by x_i (resp. y_i). We use some terminology from [9]. Let k be an integer such that $1 \leq k \leq n$. By a *box*, we mean a 2-dimensional axes-parallel rectangle of the form $\Pi_{i=1}^{2}[a_i : b_i)$. If $b_i = a_i + \delta$ for $i = 1, 2$, then we call the box as a δ-box. The closure of a box, i.e., the product of 2 closed intervals $[a_i : b_i]$, $i = 1, 2$ is called a *closed box*. We use the following notations.

- μ denotes a function which maps a set V of points in 2-dimensions to a real number $\mu(V)$.
- $S(P, k)$ denotes the problem of finding a subset of P of size k whose measure is minimal among all k-point subsets.
- $\mu_{opt}(P)$ denotes the minimal measure.
- P_{opt} denotes a k- point subset of P such that $\mu(P_{opt}) = \mu_{opt}(P)$.
- $\mathcal{A}$ denotes a CREW PRAM algorithm that solves problem $S(P', k)$, where $P' \subset P$ and $|P'| = O(k)$.
- $T(n, k)$ (resp. $W(n, k)$) denotes the time (resp. work) complexity of algorithm $\mathcal{A}$.

For example, if $\mu(B)$ is the diameter of set B, then $S(B, k)$ is the problem of finding a subset of size k whose diameter is minimal among all k-point subsets. We make the following assumptions about the measure μ which are proved later for each of the problems considered in this paper.

Assumption 2.1 *There exists a closed $\mu_{opt}(P)$ box that contains an optimal solution P_{opt}.*

Assumption 2.2 *There exists an integer constant c such that for any $\delta < \mu_{opt}(P)/c$, any closed δ box contains less than k points of P.*

The constant c will depend on the particular problem and we will fix this constant later for each problem. For example, c is 4 for the problem of finding the minimum L_∞-perimeter k-point subset. Our algorithms are based on the following simple lemma.

Lemma 2.1 *Let δ be a real number. Assume there exists a closed δ box that contains at least k points of P. Then $\mu_{opt}(P) \leq c\delta$ and there exists a closed $(c\delta)$ box that contains the optimal solution P_{opt}.*

Our algorithms work in the following way. We reduce problem $S(P, k)$ to $O(n/k)$ subproblems $S(P', k)$ for subsets $P' \subset P$ of size $O(k)$. All the subproblems are solved simultaneously in parallel. Each subproblem is solved by a CREW PRAM algorithm $\mathcal{A}$. We need the following definition for discussing the decomposition of our problems.

Definition 2.2 *Let δ be a positive real number, let α and β $(\alpha \leq \beta)$ be positive integers, and let $\mathcal{R}$ be a collection of δ-boxes such that*

1. *each box in $\mathcal{R}$ contains at least one point of P,*
2. *each point of P is contained in exactly one box of $\mathcal{R}$,*
3. *there is a box in $\mathcal{R}$ that contains at least α points of P,*
4. *each box in $\mathcal{R}$ contains at most β points of P.*

Then $\mathcal{R}$ is called an $(\alpha, \beta; \delta)$-covering of P.

We now give the generic algorithm $\mathcal{G}$ for both of our problems.

Algorithm 1 *(main steps of the generic algorithm $\mathcal{G}$)*

1. Compute a positive real number δ together with a $(k, 4k; \delta)$ -covering $\mathcal{R}$ of P.

   In the next section, we show that such a δ and such a covering $\mathcal{R}$ exist. We compute this covering in $O(\log^2 n)$ time using $O(n)$ processors in the CREW PRAM. We will also store this collection in a data structure of size $O(n)$ such that point location queries can be answered in $O(\log n)$ time by a single processor. We will build this data structure also in $O(\log n)$ time, using $O(n)$ processors.

2. For each box $B \in \mathcal{R}$, do the following two sub steps. The justification for step (a) is given later in Lemma 3.3. More specifically, we show in Lemma 3.3 that an optimal solution can be found within a box B and the $(2c+1)^2 - 1$ boxes that surround B. The constant c depends on a particular problem.

   (a) Find all boxes in $\mathcal{R}$ that overlap the $(2c+1)\delta$-box that is centered at B. Note that the total number of δ-boxes in a $(2c+1)\delta$-box centered at B is $(2c+1)^2$. These boxes are found as follows:

Let (b_1, b_2) be the lower-left corner of B. Then in the data structure for $\mathcal{R}$, we locate the $(2c+1)^2$ points $(b_1 + \epsilon_1\delta, b_2 + \epsilon_2\delta)$, $\epsilon_i \in \{-c, -c+1, \ldots, c-1, c\}, i = 1, 2$.

(b) Let P' be the subset of points of P that are contained in the boxes found in the previous step. If $|P'| \geq k$, solve problem $S(P', k)$ using algorithm $\mathcal{A}$.

3. Find the optimal solution out of $O(n/k)$ solutions found in Step **2**. Output μ_{opt} and P_{opt}.

Theorem 2.3 *The algorithm correctly solves the problem $S(P, k)$. Moreover, there is a constant c' such that the algorithm takes $O(\log^2 n + T(c'k, k))$ time and $O(n \log^2 n + (n/k)W(c'k, k))$ work.*

Proof: By Lemma 2.1, there is a closed $(c\delta)$-box that contains the optimal solution. It is clear that this box must be contained in the $(2c+1)\delta$-box that is centered at some box of $\mathcal{R}$. The algorithm checks all these $(2c+1)\delta$-boxes. If there are less than k points in such a box, then it does not contain the optimal solution. Hence, the correctness of the algorithm follows.

Each box in $\mathcal{R}$ contains at most $4k$ points. Moreover, the point location queries in Step $2(a)$ find at most $(2c+1)^2$ boxes of $\mathcal{R}$. Therefore, the set P' in step $2(b)$ has size at most $(2c+1)^2 4k$. There are at most $(2c+1)^2(n/k)$ boxes $B \in \mathcal{R}$ that give rise to a subset P' of size at least k. Hence the algorithm $\mathcal{A}$ is applied in parallel to at most $(2c+1)^2(n/k)$ different subsets.

As mentioned already, we will show in section 3 that the real number δ and the covering $\mathcal{R}$ can be computed in $O(\log^2 n)$ time using $O(n)$ processors. Moreover, in $O(\log n)$ time and using $O(n)$ processors, $\mathcal{R}$ can be stored in a data structure such that a point location query can be answered by a single processor in $O(\log n)$ time. In Step 3, we have to compute the maximum of $O(n/k)$ quantities. This can be easily done in $O(\log n)$ time using $O(n)$ processors [15] i.e., $O(n \log n)$ work. Also, μ_{opt} and P_{opt} can be reported within this time and work. So, the running time of our algorithm is bounded by :

$$O(\log^2 n + T((2c+1)^2 4k, k)) \quad (1)$$

and the work done(processor-time product) by the algorithm is :

$$O(n \log^2 n + (n/k)(2c+1)^2 W((2c+1)^2 4k, k)) \quad (2)$$

This completes the proof. □

3 Parallel algorithm for computing a $(k, 4k; \delta)$ covering

In the previous section, we have seen that we need a real number δ and a $(k, 4k; \delta)$ covering $\mathcal{R}$ for the point set P. We also need a data structure for these boxes that supports point location queries. Suppose the value of δ and a δ-box containing at least k points are known. Then we can take a grid with mesh size δ containing this box and take for $\mathcal{R}$ the set of all non-empty grid cells. We call such an $\mathcal{R}$ as a *perfect δ-grid*. The problem in using the perfect δ-grid is that that we need the *floor function* for distributing the points among the grid cells. We can explain this in the following way. Consider a point set such that a small δ-box contains k-points, but there exist two points with large difference of x or y coordinates. In this case, the number of delta slabs may be enormously large compared to the number of points in the set. Hence in a perfect δ-grid (shown in Figure 1), we cannot do binary search to place individual points in their respective δ-boxes. The only way to place the points in the δ-boxes would be to use the non algebraic floor function. If we follow this approach, our algorithm will fall outside the *algebraic decision tree* model of computation.

In this section, we introduce so-called *degraded grids*, that have basically the same properties as perfect grids. We can build a degraded grid and search in it without using the floor function and the number of δ-slabs in such a grid is bounded by the number of points in the set.

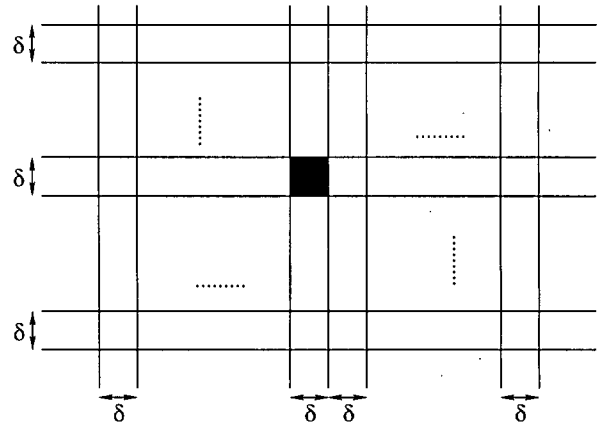


Figure 1: A perfect δ-grid. The shaded box with k-points determines the widths of the δ-slabs.

3.1 Degraded grids

In a standard δ-grid, we divide the plane into slabs of width exactly δ. In such a grid, if we fix a lattice point, all other lattice points are automatically fixed. For example, if $(0, 0)$ is a lattice point, then a slab along the x axis consists of all points that have their x coordinate between $j\delta$ and $(j+1)\delta$ for some integer j. In a degraded δ-grid, the slabs do not start and end at integer multiples of δ. The slabs have width at least δ and the slabs that contain points have width exactly δ. That is, while a δ-grid may be defined independently of the point set by fixing an arbitrary point on the

plane to be a lattice point, the degraded δ-grid is defined in terms of the point set stored in it. Note that the empty slabs in a degraded δ-grid can have arbitrary width.

We first give a formal definition of a degraded one-dimensional δ-grid.

Definition 3.1 *[9] Let P be a set of n real numbers and let δ be a positive real number. Let $a_1, a_2, \ldots, a_l$ be a sequence of real numbers such that*

1. *for all $i \leq j < l$, $a_{j+1} \geq a_j + \delta$,*
2. *for all $p \in P$, $a_1 \leq p < a_l$,*
3. *for all $1 \leq j < l$, if there is a point $p \in P$ such that $a_j \leq p < a_{j+1}$, then $a_{j+1} = a_j + \delta$.*

The collection of intervals $[a_j, a_{j+1})$, $1 \leq j < l$, is called a one-dimensional degraded δ-grid for P.

3.1.1 Construction of one-dimensional δ-grid in parallel

We first discuss the construction of a one-dimensional δ-grid, for a given real number δ. We assume in this section that δ is known and we show in Section 3.2 how to choose a correct value for δ. We first sort the elements in $P = \{p_i | 1 \leq i \leq n\}$ according to the increasing x coordinates and store them in an array $X = \{x_i | 1 \leq i \leq n\}$. Now, we associate a processor with each point in the sorted array X. The processor associated with point x_i checks whether $|x_{i+1} - x_i| \geq \delta$. If this check succeeds, the processor marks the point x_{i+1} with 1. The point x_{i+1} is the left boundary of a δ-slab. If the check fails, the processor marks the point x_{i+1} with 0. The left-most element in the array X, i.e., x_1 is marked with 1. This step clearly can be done by $O(n)$ processors in $O(\log n)$ time using the sorting algorithm by Cole [5].

After this, for every point x_i, we find the nearest point which is marked 1 to its left. This can be done by the algorithm for finding the *all nearest larger value* by Berkman *et al* [4]. This computation takes $O(n/\log n)$ processors and $O(\log n)$ time in the CREW PRAM.

Now, every processor knows the nearest point marked with 1 to its left. Suppose, for the point x_m, the nearest point marked with 1 to its left is the point x_n. The *floor function* $\lfloor K \rfloor$ computes the largest integer that is smaller than a real number K. The processor associated with x_m computes the quantity

$$c = x_n + \delta \times \lfloor |x_m - x_n|/\delta \rfloor$$

The reason for computing the quantity c is the following. For every point x_m, we need to know the left boundary of its δ-slab. Consider the points x_m and x_n, with $x_n < x_m$. If the gap $|m_m - x_n|$ is larger than δ, then we should start the left boundary of the δ-slab for x_m at the point x_m itself, since there are no points within a distance δ to the left of x_m. This is the case when $|x_m - x_n|/\delta$ is greater than 1, since the quantity c evaluates to x_m. On the other hand, if there are other points to the left of x_m, in between x_m and x_n and at a distance less than δ from x_m, then a point like that will be the starting point of the δ-slab for x_m. This is the case when $|x_m - x_n|/\delta \leq 1$. Note that, $|x_m - x_n|/\delta$ can be either 1 or 0 in this case. If $|x_m - x_n|/\delta = 0$, c evaluates to x_n and the δ-slab for x_m starts at x_n. If $|x_m - x_n|/\delta = 1$, c evaluates to $x_n + \delta$ and the δ-slab for x_m starts at $x_n + \delta$.

Note that, $c \leq x_m \leq c + \delta$. As explained above, c is the left boundary of the δ-slab to which x_m belongs. We can compute the quantity c without using the non-algebraic *floor function* in the following way. We know that the difference of x coordinates of two consecutive elements between x_n and x_m is less than δ. In other words, $0 \leq |x_m - x_n| \leq (m-n)\delta$ or, $0 \leq (|x_m - x_n|/\delta) \leq (m-n)$. So, we can find the integer $\lfloor |x_m - x_n|/\delta \rfloor$ by doing a binary search in the range 0 to $(m-n)$. This step again takes $O(n)$ processors and $O(\log n)$ time.

At this stage, every element knows the left boundary of its δ-slab. The elements within a δ-slab can be counted easily since we know the boundaries of the δ-slabs. The difference of indices of two consecutive boundaries is the number of points inside that δ-slab. This step also can be done within $O(\log n)$ time using $O(n)$ processors.

3.1.2 Construction of two-dimensional δ-grid in parallel

A two-dimensional degraded δ-grid(Figure 2) can be defined recursively in the following way. We construct a δ-grid for each one-dimensional δ-slab found in Section 3.1.1. We sort the points inside every one-dimensional δ-slab according to decreasing y coordinate. This takes overall $O(n)$ processors and $O(\log n)$ time [5]. Now we apply the algorithm for constructing one-dimensional δ-grid within each slab. Again, the processor and time requirements are $O(n)$ and $O(\log n)$ respectively. After this computation is over, we have the two-dimensional δ-grid and we know the number of points in each occupied cell of the grid from the grid boundaries.

We construct the data structure for point location as follows. The boundaries of the delta slabs perpendicular to the x axis are kept in a balanced binary tree $\mathcal{T}$ in sorted order. This tree can be built in $O(\log n)$ time, using $O(n)$ processors. Within each such slab, the boundaries of the δ-slabs perpendicular to the y axis can be kept in balanced binary trees. Any point location query can be performed by first searching in the tree $\mathcal{T}$ for locating the δ-slab perpendicular to the x axis and then searching in the balanced tree for this δ-slab. So, for locating a point within a slab, we need to do two binary searches. This takes $O(\log n)$ time by a single processor.

Lemma 3.2 *Given a real number δ, a two-dimensional degraded δ-grid can be constructed in $O(\log n)$ time, using $O(n)$ processors in the CREW PRAM. Moreover, this grid can be stored in a data structure of size $O(n)$, such that*

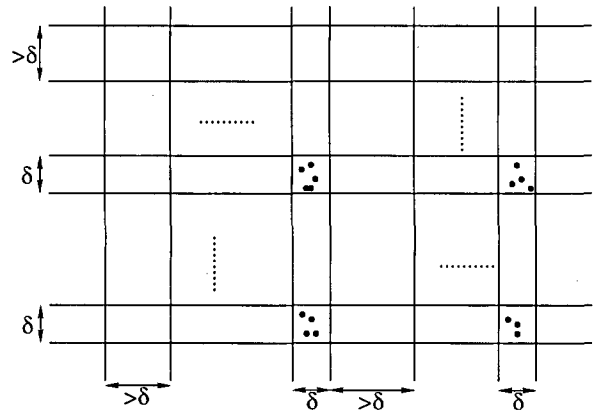


Figure 2: A degraded δ-grid. The height and width of a box are exactly δ if there are points in it. The height and width of empty boxes are $> \delta$ and can be arbitrarily large in general. The points in the set are shown as solid circles.

a single processor can answer point location queries in $O(\log n)$ *time.*

We need the following Lemma for justifying step 2(a) in Algorithm 1.

Lemma 3.3 *Let* $p \in P$ *and let* B *be the degraded* δ*-box for* P *that contains* p*. Let* c *be an integer. All the points of* P *that are within distance* $c\delta$ *from* p *are contained in* B *and in the* $(2c+1)^2$ *boxes (including* B*) that surround* B*.*

Proof: We prove this by contradiction and refer to Figure 3. Let p be a point in the box B and q be a point at a distance $c\delta$ from p. Assume that q is outside the $(2c+1)^2$ boxes surrounding B. We shift the origin of the coordinate system at p and assume that the line $\overline{pq}$ makes an angle θ with the positive x-axis. Since q is outside the $(2c+1)^2$ boxes, at least one of the projections of $\overline{pq}$ on the coordinate axes should be outside the $(2c+1)^2$ boxes. We assume that the x-projection $c\delta|\cos\theta|$ is outside. Therefore, $c\delta|\cos\theta| > c\delta$ and hence $|\cos\theta| > 1$ which is a contradiction. The lemma can be proved in a similar way for other positions of the point q. □

3.2 Parallel construction of a degraded δ-grid with $O(k)$ points per cell

We have assumed a correct value for δ in the previous section and shown how to construct a two-dimensional degraded δ grid. In this section, we first show how to choose a correct δ and then we discuss the complete algorithm for constructing a $(k, 4k; \delta)$ covering.

We first use some notations from [9] and prove the existence of such a δ.

Definition 3.4 *Suppose* P *is a set of* n *points on the plane.*

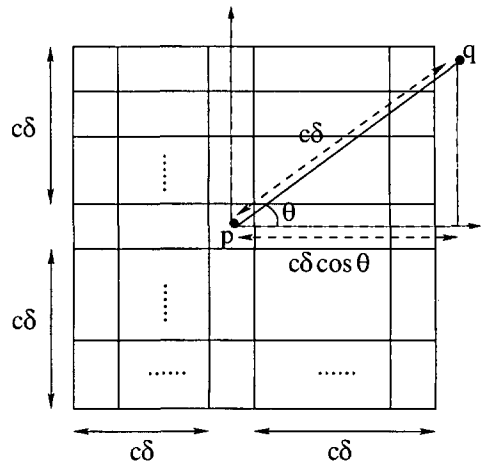


Figure 3: Proof of Lemma 3.3.

- *Assume* δ *is a real number and* $\mathcal{R}$ *is a degraded* δ*-grid for* P*. We number the boxes arbitrarily from* $1, 2, \ldots, r$*, where the number of cells in* $\mathcal{R}$ *is* r*. We define* n_i *to be the number of points of* P*, that are contained in the* i*-th box in* $\mathcal{R}$*. Then we denote* $M(\mathcal{R}) = max_{1 \le i \le r} n_i$*.*

- *Let* P' *be a subset of* P *of size* $4k$ *with minimal* L_∞ *diameter among all* $(4k)$*-point subsets. Then,* δ^* *denotes the* L_∞*-diameter of* P'*.*

Lemma 3.5 *Using the notation above, the following holds:*

1. *For any* $\delta \ge \delta^*$ *and any degraded* δ*-grid* $\mathcal{R}$ *for* P*, we have* $M(\mathcal{R}) \ge k$*.*

2. *For any* $\delta \le \delta^*$ *and any degraded* δ*-grid* $\mathcal{R}$ *of* P*, we have* $M(\mathcal{R}) \le 4k$*.*

Proof: For proving condition 1, let $\delta \ge \delta^*$ and let $\mathcal{R}$ be a degraded δ-grid for P. The set P' is contained in an axes-parallel square $\mathcal{S}$ of side-length δ^*. Since $\delta \ge \delta^*$, $\mathcal{S}$ overlaps at most 4 boxes of $\mathcal{R}$. Since the size of P' is $4k$, there must be at least one box in $\mathcal{R}$ which contains at least k points of P. This shows that $M(\mathcal{R}) \ge k$.

For proving condition 2, we assume that $\delta \le \delta^*$ and $\mathcal{R}$ is a degraded δ-grid for P. Assume that $M(\mathcal{R}) > 4k$. Then there is a box in $\mathcal{R}$ which contains more than $4k$ points of P. Since this box has side lengths δ, there are $4k$ points in P which have L_∞-diameter less than δ. But $\delta^* > \delta$ and hence there is at least one $4k$-point subset with L_∞-diameter less than δ^*. This clearly contradicts our definition of δ^*. Hence the assumption $M(\mathcal{R}) > 4k$ must be false. □

The parallel algorithm searches for a real number δ together with a degraded δ-grid $\mathcal{R}$ of P such that $k \le M(\mathcal{R}) \le 4k$. This grid is the $(k, 4k; \delta)$-covering we want for our algorithms. Lemma 3.5 implies that there is at

least one such δ for which such a covering exists, namely $\delta = \delta^*$. Note that, such a δ is contained in the set all the L_∞ distances between pairs of points in P. Our search for such a δ is based on the sequential algorithm of Johnson and Mizoguchi [16]. A parallel algorithm based on the algorithm in [16] has been used by Lenhof and Smid [17]. Their algorithm takes $O(\log^2 n \log\log n)$ expected time and $O(n \log n \log\log n)$ work in the *randomized CRCW PRAM* model of parallel computation. We present a similar but much simpler algorithm for computing a $(k, 4k; \delta)$ covering for the set P in the weaker CREW PRAM.

We recall the notion of weighted medians from [16]. Let, $x_1, x_2, \ldots, x_n$ be a sequence of real numbers such that every element x_i has a weight w_i, and w_i is a positive real number. Let $W = \sum_{j=1}^{n} w_i$. Element x_i is called a weighted median if

$$\sum_{j:x_j<x_i} w_j < W/2 \quad and \quad \sum_{j:x_j \le x_i} w_j \ge W/2 \qquad (3)$$

Lemma 3.6 *[16] The weighted median of a set of n weighted real numbers can be computed in $O(n)$ time.*

In the following discussion, we will use a parallel algorithm for computing weighted medians which requires $O(n \log n)$ work. As mentioned earlier, we search for the appropriate δ in the set of all L_∞ distances between pairs of points. We do not enumerate all candidate differences since that will require $\Omega(n^2)$ work. We maintain the candidate differences in an implicit way. The algorithm maintains two sorted arrays X and Y which store the x and y coordinates of the points $p_i, 1 \le i \le n$ in sorted order. For the j^{th} entry $X[j]$ in the array X, we store an interval $[a_{xj} : b_{xj}]$, where, a_{xj} and b_{xj} are integers, such that $j < a_{xj} \le b_{xj} + 1 \le n + 1$. Similarly, with the j^{th} entry $Y[j]$ in the array Y, we store an interval $[a_{yj} : b_{yj}]$, where a_{yj} and b_{yj} are integers and $j < a_{yj} \le b_{yj}+1 \le n+1$. Let, j and j' $(j < j')$ be such that $X[j] = x_r$ and $X[j'] = x_s$. Then $|x_s - x_r|$ is a candidate difference iff $a_{xj} \le j' \le b_{xj}$.

Hence, the total number of candidate differences is equal to

$$\sum_{m=x,y} \sum_{j=1}^{n-1} (b_{mj} - a_{mj} + 1) \qquad (4)$$

The intervals are initialized as follows $[a_{mj} : b_{mj}] = [j + 1, n]$, where $m = x, y$. So, initially this sum is

$$\sum_{m=x,y} \sum_{j=1}^{n-1} (n - j) = 2\binom{n}{2} \qquad (5)$$

Equation 5 is evaluated as follows. The multiplicative factor 2 comes due to the outer summation over x and y. The inner summation can be written as :

$$\sum_{j=1}^{n-1} (n-j) = (n-1) + (n-2) + \ldots (n-1) terms$$

$$= n(n-1) - (1 + 2 + \ldots (n-1) terms)$$

$$= n(n-1) - \frac{n(n-1)}{2} = \frac{n(n-1)}{2} = \binom{n}{2}$$

Hence, we get $\sum_{m=x,y} \sum_{j=1}^{n-1} (n-j) = 2\binom{n}{2}$. We use a technique similar to that by Johnson and Mizoguchi [16] for computing the weighted medians. The idea in the algorithm in [16] is to reduce the interval where the weighted median lies in a sequence of iterations. As in [16], the parallel algorithm performs a sequence of iterations and in each iteration, the summation in equation (4) is decreased by a constant factor. The algorithm maintains the following invariant.

Invariant: *At each step of the iteration, the value δ^* is contained in the set of candidate differences.*

We describe the main steps for constructing the $(k, 4k; \delta)$-covering in Algorithm 2.

Algorithm 2 *(Main steps in the algorithm for choosing a $(k, 4k; \delta)$-covering)*

1. Sort the points according to x and y coordinates and store them in the arrays X and Y. For every entry $X[j]$ (resp. $Y[j]$) $(1 \le j \le n)$, initialize the intervals associated with $X[j]$ and $Y[j]$ in the following way. $[a_{mj} : b_{mj}] = [j + 1, n]$, where, $m = x, y$. This initialization including sorting can be done in $O(\log n)$ time using $O(n)$ processors.

2. For each $i \le j < n$, if $a_{xj} \le b_{xj}$, take the pair $X[\lfloor (a_{xj} + b_{xj})/2 \rfloor]$ and $X[j]$, and take the absolute difference of their x coordinates. Give weight $(b_{xj} - a_{xj} + 1)$ to this difference. Do a similar computation for the elements in the array Y. This computation gives at most $2(n-1)$ weighted differences and can be done in $O(1)$ time using $O(n)$ processors.

3. Let $s_1, s_2, \ldots, s_r$ be the union of X and Y sequences of numbers obtained from Step 2. Note that $r = O(n)$. We sort the s_i's according to increasing order and find a weighted median δ by using a prefix sum algorithm and equation (3). This can be done in $O(\log n)$ time, using $O(n)$ processors [5, 15].

4. Construct a δ-covering $\mathcal{R}$ of P by using the algorithm in Section 3.1 and compute $M(\mathcal{R})$. This construction takes $O(n)$ processors and $O(\log n)$ time. There are three possible cases (which can be done in $O(\log n)$ time, using $O(n)$ processors):

   (a) $k \le M(\mathcal{R}) \le 4k$, then output δ and $\mathcal{R}$ and stop.

   (b) If $M(\mathcal{R}) < k$, then for each pair $X[\lfloor (a_{xj} + b_{xj})/2 \rfloor]$ and $X[j]$ (reps. $Y[\lfloor (a_{yj} + b_{yj})/2 \rfloor]$ and $Y[j]$) selected in Step 2 such that the difference of their x (resp. y) coordinate is at most δ, set $a_{xj} := \lfloor (a_{xj} + b_{xj})/2 \rfloor + 1$ (resp. $a_{yj} := \lfloor (a_{yj} + b_{yj})/2 \rfloor + 1$). Goto Step 2.

(c) If $M(\mathcal{R}) > 4k$, then for each pair $X[\lfloor(a_{xj} + b_{xj})/2\rfloor]$ and $X[j]$ (resp. $Y[\lfloor(a_{yj}+b_{yj})/2\rfloor]$ and $Y[j]$) selected in Step 2 such that the difference of their x (resp. y) coordinate is at least δ, set $b_{xj} := \lfloor(a_{xj}+b_{xj})/2\rfloor - 1$ (resp. $a_{yj} := \lfloor(a_{yj}+b_{yj})/2\rfloor - 1$). Goto Step 2.

Lemma 3.7 *The algorithm correctly maintains the invariant at every iteration.*

Proof: After the initialization, the total number of candidate differences is equal to $2\binom{n}{2}$ as indicated in equation (5). Obviously, δ^* is included in this set. So, the invariant holds initially. Consider an iteration and assume that Step 4(b) occurs. i.e., $M(R) < k$. Then, from Lemma 3.5, we know that $\delta < \delta^*$. The algorithm removes differences from the set of candidate differences which are at most δ. Hence, the invariant holds after such an iteration. If Step 4(c) occurs, then from Lemma 3.5, $\delta > \delta^*$ and the algorithm removes differences which are at least equal to δ. The invariant holds in this case also. □

Lemma 3.8 *The algorithm makes at most $O(\log n)$ iterations.*

Proof: Let W and W' be the number of candidate differences immediately before and after an iteration. Also, the current weighted median found by the algorithm is δ. In the following discussion, the subscript m takes two values, namely, x and y. Suppose a_{mj} and b_{mj} (resp. a'_{mj} and b'_{mj}) denote the intervals immediately before (resp. after) the iteration. We consider Step 4(b). If $a'_{mj} \neq a_{mj}$, then $a'_{mj} = \lfloor(a_{mj} + b_{mj})/2\rfloor + 1$ and $b'_{mj} = b_{mj}$. Since a_{mj} and b_{mj} are integers, we have $a'_{mj} \geq (a_{mj} + b_{mj} + 1)/2$. So,

$$\begin{aligned} W &= W' + \sum_{m=x,y} \sum_{j} (a'_{mj} - a_{mj}), \\ \text{where } a'_{mj} &\neq a_{mj} \\ &\geq W' + \frac{1}{2} \sum_{m=x,y} \sum_{j} (b_{mj} - a_{mj} + 1) \\ &\geq W' + \frac{1}{2}\frac{W}{2} \end{aligned}$$

The last inequality follows from the fact that δ is a weighted median. Hence,

$$W' \leq \frac{3}{4}W. \tag{6}$$

If Step 4(c) applies, the inequality can be proved in a similar way. We know from Lemma 3.7 that the invariant is always maintained correctly, hence the algorithm terminates. From equation (6), it is clear that at each iteration, at least (1/4)th of the candidate differences are discarded. This completes the proof. □

Lemma 3.9 *A $(k, 4k; \delta)$ covering for the set P can be constructed in $O(\log^2 n)$ time using $O(n)$ processors in the CREW PRAM.*

Proof: It is clear from the description of Algorithm 2 that each step of the algorithm takes $O(\log n)$ time and $O(n)$ processors. From Lemma 3.8, we know that the algorithm terminates within $O(\log n)$ steps. This completes the proof. □

4 Parallel algorithm for computing minimum L_∞-perimeter k point subset

For the L_∞ metric, the distance function d_∞ is defined as $d_\infty((x_1, y_1), (x_2, y_2)) = max(|x_1 - x_2|, |y_1 - y_2|)$. In this section, we discuss the algorithm for finding a k-point subset with minimum L_∞-perimeter from a set of $O(k)$ points. This will imply a solution for the problem $S(P, k)$ as discussed in Theorem 2.3. We first prove the following lemma.

Lemma 4.1 *For μ the L_∞-perimeter measure, Assumptions 2.1 and 2.2 hold with $c = 4$.*

Proof: Suppose there is no closed $\mu_{opt}(P)$-box that contains the optimal solution P_{opt}. Then the L_∞ perimeter of P_{opt} must be greater than $\mu_{opt}(P)$. This contradicts the definition of $\mu_{opt}(P)$. This shows that Assumption 2.1 holds.

For proving Assumption 2.2, let $\delta < \mu_{opt}(P)/4$. Suppose there is a closed δ-box that still contains at least k points. The perimeter of this δ-box is 4δ. This implies that there are k points that have L_∞ perimeter at most 4δ. Now, from our assumption on δ, $4\delta < \mu_{opt}(P)$. Hence, there is a k-point subset with perimeter less than $\mu_{opt}(P)$ which contradicts the definition of $\mu_{opt}(P)$. □

We need the following property for our algorithm $\mathcal{A}$.

Property 4.1 *The four sides of the minimum L_∞-perimeter k-point subset pass through four points from the set.*

Proof: Otherwise, we can reduce the perimeter by moving the sides. □

We also need the following result by Datta [8] for parallel range searching.

Lemma 4.2 *Given a set of n points on the plane, it is possible to construct a data structure called a* range tree *(T) in $O(\log n)$ time by using $O(n)$ processors in the CREW PRAM. Given an orthogonal query rectangle R, a single processor can count the number of points as well as find the point with maximum x-coordinate inside R in $O(\log n)$ time using the range tree T.*

Now we discuss our algorithm $\mathcal{A}$. We first sort the points in the $O(k)$ subset according to increasing x coordinate and store them in an array X. This requires $O(\log k)$ time and $O(k)$ processors by Cole's [5] algorithm. We also construct a range tree T as described in Lemma 4.2 using the algorithm by Datta [8]. A range tree is a balanced binary tree that supports orthogonal range queries. Given a set of points S on the plane and an axes-parallel rectangle R, an orthogonal range query may ask for either the number of points of S which lie inside R, or the enumeration of these points. We refer to the book by Preparata and Shamos [20] for details of this data structure. Datta [8] constructs this data structure by using Cole's [5] parallel merge sort algorithm. The details of this construction can be found in [8].

If the left side of a rectangle passes through a point p_k, we say that p_k is the left support for the rectangle. Consider a pair of points (p_i, p_j) such that $x_i < x_j$ and $y_i > y_j$. Such a pair (p_i, p_j) fixes the bottom-left corner of a rectangle. We search for a k-point subset enclosed in a rectangle with the bottom-left corner fixed in this way. The left (resp. bottom) support of such a rectangle is the point p_i (resp. p_j). The top support is a point p_k such that $x_k > x_i$ and $y_k > y_i$. We associate $O(k)$ processors with every pair (p_i, p_j). One processor is associated with a point p_k which satisfies the condition above. Now, three supports of the rectangle are fixed and we have to fix the right support such that the rectangle contains k points.

The processor associated with p_k does a binary search for finding the right support. Note that, during this binary search, we should consider a point p_m as a candidate for the right support such that, $x_m > x_i$, $x_m > max(x_k, x_j)$ and $y_k > y_m > y_j$. For this purpose, it will be convenient if we have all the points in the semi-unbounded rectangle defined by p_k, p_i and p_j sorted according to increasing x order. But sorting these points for every such semi-unbounded rectangle requires too much work. Instead we do a binary search in the array X. Suppose the current candidate for the right support is the point p_n. There are two possibilities. In the first case, $y_k > y_n > y_j$. We do a range search in the range tree T. The query range is a rectangle R such that its three sides are fixed as explained before and the right side is the point p_n. If the number of points inside this rectangle is k, we store this rectangle as a possible candidate rectangle. Otherwise, if the number of points is less than k (resp. greater than k), we shift the right support towards higher (resp. lower) x coordinate points.

In the second case, y_n may not be inside the interval $[y_k, y_j]$. We do a range search again with the right side at the position x_n. If there are exactly k points inside this rectangle, we can find the right side of this rectangle by doing a range searching in the tree T and finding the maximum x coordinate point inside this rectangle.

There are $O(k^2)$ pairs like (p_i, p_j). For each such pair, we allocate $O(k)$ processors. So, the overall processor requirement is $O(k^3)$. The binary search takes $O(\log k)$ time and at every position during the binary search, we do a range search which takes $O(\log k)$ time. So, the overall time requirement is $O(\log^2 k)$. After this, there are at most $O(k^3)$ candidate rectangles left and among them the minimum perimeter rectangle can be found easily in $O(\log k)$ time, using $O(k^3)$ processors. This concludes the discussion of our algorithm. The main steps are given in Algorithm 3.

Algorithm 3

1. Sort the points according to increasing x coordinates and store them in an array X. This can be done in $O(\log k)$ time using $O(k)$ processors by Cole's [5] algorithm.

2. Construct a range tree T for answering orthogonal range counting and maxima queries by Datta's [8] algorithm. This step takes $O(\log k)$ time and $O(k)$ processors, by Lemma 4.2.

3. For every pair of points (p_i, p_j) such that $x_i < x_j$ and $y_i > y_j$, we allocate $O(k)$ processors. This group of processors find the minimum L_∞-perimeter k-point subset with p_i (resp. p_j) as the left (resp. bottom) support. This step takes $O(\log^2 k)$ time and $O(k^3)$ processors, as discussed above.

4. Among the possible $O(k^3)$ rectangles found in Step 3, find the minimum perimeter one. This can be done in $O(\log k)$ time, using $O(k^3)$ processors.

Lemma 4.3 *The overall time and work requirements of Algorithm* 3 *is* $O(\log^2 k)$ *and* $O(k^3 \log^2 k)$.

Algorithm 3 is applied simultaneously to all the (n/k) subsets of size $O(k)$ each. We state the result in the following theorem.

Theorem 4.4 *The minimum* L_∞*-perimeter* k*-point subset can be computed in* $O(\log^2 n)$ *time using* $O(n \log^2 n + nk^2 \log^2 k)$ *work.*

Proof: We get the stated processor and work bounds by plugging in the values from Lemma 4.3 into equations (1) and (2) in Theorem 2.3. From Lemma 4.3, $T((2c+1)^2 4k, k) = O(\log^2 k)$ and hence, the total time requirement is $O(\log^2 n + \log^2 k)$, *i.e.*, $O(\log^2 n)$, since $n \geq k$. Similarly, $W((2c+1)^2 4k, k) = O(k^3 \log^2 k)$ and hence the total work done by the algorithm is $O(n \log^2 n + (n/k)k^3 \log^2 k) = O(n \log^2 n + nk^2 \log^2 k)$. □

5 Parallel algorithm for computing a k-point subset with minimum L_∞-diameter

In this section, we discuss the algorithm for finding a k-point subset with minimum L_∞-diameter. We apply this algorithm simultaneously in parallel on small sets of $O(k)$ points each. This will imply a solution for problem $S(P,k)$ by Theorem 2.3. First, we state the following lemma.

Lemma 5.1 *For μ the L_∞-diameter measure, Assumptions 2.1 and 2.2 hold with $c = 1$.*

Proof: Suppose there is no closed $\mu_{opt}(P)$-box that contains the optimal solution. Then there must be two points in P_{opt} that have L_∞- distance greater than μ_{opt}, a contradiction. This shows that Assumption 2.1 holds.

For proving Assumption 2.2, let $\delta < \mu_{opt}(P)$. Assume that there is a closed δ-box that still contains k points. Then this subset of k points has L_∞-diameter less than $\mu_{opt}(P)$. This contradicts the definition of $\mu_{opt}(P)$. □

Our algorithm is based on the following lemma.

Lemma 5.2 *The minimum L_∞-diameter of a k-point subset is determined either by $|x_i - x_j|$ or by $|y_i - y_j|$ for two points $p_i, p_j \in P$. Moreover, two opposite sides of the minimum L_∞-diameter set must be supported by these two points.*

Proof: The first part of the Lemma is obvious. If two opposite sides are not supported by the pair of points determining the L_∞-diameter, the L_∞ diameter of the set can be decreased. □

We also need the following result.

Lemma 5.3 *Given a point set P and a fixed size square S, in $O(\log n)$ time and using $O(n)$ processors, we can find an axes-parallel placement of S such that S contains the maximum number of points from the set P. This computation can be done in the CREW PRAM.*

Proof: We replace each point in the set P by a square of size S. The square S_i corresponding to the point $p_i \in P$ is placed such that its bottom-left corner coincides with the point p_i and its sides are parallel to the axes. Consider the arrangement of this set of squares $S_1, S_2, \ldots, S_n$. We define a graph $\mathcal{H}$ such that the set of vertices for this graph is the set of squares and two vertices are connected by an edge if the two corresponding squares intersect. Consider the maximum clique in the graph $\mathcal{H}$. It is easy to see that all the squares in this clique have a common point p inside them. We place the square S with its top-right corner at this point. This placement of S will contain all the points in P which gave rise to the clique $\mathcal{H}$.

The details of an algorithm are as follows. We can transform the point set into squares of size S in $O(1)$ time, using $O(n)$ processors. Then we apply an algorithm by Chandran *et al.* [6] for finding a maximum clique and a deepest point in the arrangement of squares. This algorithm takes $O(\log n)$ time and $O(n)$ processors in the CREW PRAM. □

Now, we give the details of our algorithm $\mathcal{A}$. From Lemma 5.2, it is clear that the L_∞-diameter for the minimum k-point set is present in the set of all coordinate differences. In our algorithm, we search for the optimal such difference along each axis. We first sort the points along each coordinate axis x and y and store them in two arrays X and Y. The differences among the coordinates along each axis can be represented as a triangular matrix. There are two $O(k) \times O(k)$ triangular matrices in our case. We represent them as M_x and M_y. The first row of M_x has the difference of x coordinate between the first point and all the other points. The matrix M_y is also defined in a similar way. Note that both rows and columns are in sorted order in these matrices. For example, the first two elements in the j^{th} column of matrix M_x actually denote the differences $|x_1 - x_j|$ and $|x_2 - x_j|$. Obviously, the first quantity is greater than the second, since the numbers $x_1, x_2, \ldots, x_{O(k)}$ are sorted in increasing order.

We do not actually construct the matrices M_x and M_y, since that will require $\Omega(k^2)$ work. But we can generate any element in $O(1)$ time from the sorted arrays X and Y. So, we can effectively assume as if the complete matrices are available. We have to repeatedly select elements from these matrices for doing a binary search to find an appropriate L_∞-diameter. An $O(k)$ time sequential algorithm has been presented by Frederickson and Johnson [13] for selection from matrices with sorted rows and columns. The best known parallel algorithm for selecting a specified element from sorted matrices is by Sarnath and He [21]. Their algorithm runs in $O(\log k \log\log k \log^* k)$ time using $O(k/\log k \log^* k)$ processors in the EREW PRAM. The work done by the algorithm is $O(k \log\log k)$. We use their algorithm for our binary search.

At every step of our binary search, we select an L_∞ distance from the matrices M_x and M_y. We compute a placement of a square with side lengths equal to this L_∞ distance such that this square contains maximum number of points from the set. This computation can be done by the algorithm in Lemma 5.3, in $O(\log k)$ time and $O(k)$ processors. We stop the binary search when such a square contains k points. Clearly, the time for every step of the binary search is dominated by the complexity of the selection algorithm in [21]. The overall time requirement for the complete binary search is $O(\log^2 k \log\log k \log^* k)$. The total work done by the selection algorithm during the binary search is $O(k \log k \log\log k)$ and that by the algorithm in Lemma 5.3 is $O(k \log^2 k)$. So, the overall work is $O(k \log^2 k)$. The main steps are given in Algorithm 4.

Algorithm 4

1. Sort the points according to increasing x and y coordinates and store them in two arrays X and Y by using Cole's [5] algorithm.
2. Do a binary search in the matrices M_x and M_y as explained above. At each step of the binary search, find an optimal placement of the square with the current L_∞ distance as its side length. This is done by the algorithm in Lemma 5.3.
3. Stop when an L_∞ distance is found such that the corresponding square contains k points inside it.

Lemma 5.4 *The minimum L_∞-diameter k-point subset within a set of $O(k)$ points can be computed in $O(\log^2 k \log\log k \log^* k)$ time and $O(k \log^2 k)$ work in the CREW PRAM.*

As mentioned earlier, this algorithm is applied in parallel to all the $O(k)$ subsets in the grid. So, the overall complexity of the algorithm is stated in the following theorem.

Theorem 5.5 *The minimum L_∞-diameter k-point subset within a set of n points can be found in $O(\log^2 n + \log^2 k \log\log k \log^* k)$ time and $O(n \log^2 n)$ work.*

Proof: We get the stated time complexity by plugging in the values from Lemma 5.4 into equations (1) and (2) in Theorem 2.3. From Lemma 5.4, $T((2c+1)^2 4k, k) = O(\log^2 k \log\log k \log^* k)$ and $W((2c+1)^2 4k, k) = O(k \log^2 k)$. Hence, the total time requirement is $O(\log^2 n + \log^2 k \log\log k \log^* k)$ and the total work is $O(n \log^2 n + n \log^2 k)$, *i.e.*, $O(n \log^2 n)$, since $n \geq k$. □

6 Conclusion

We have developed a general scheme for designing parallel algorithms for two point set clustering problems. This scheme allowed us to decompose the overall problem of size n into $O(n/k)$ smaller problems of size $O(k)$ each. Our algorithms are efficient in terms of overall time and work requirements even though we use relatively expensive algorithms for solving each of the $O(n/k)$ subproblems in parallel. We have been able to design efficient parallel algorithms for solving each of these $O(n/k)$ subproblems. The L_∞ metric is simpler compared to the other metrics like the Euclidean metric or the general L_p metric. However, currently there is no known parallel algorithms that solve these problems in the Euclidean metric.

The main advantage of our scheme is that once we have even a relatively inefficient parallel algorithm for solving a particular clustering problem, we can plug in that algorithm for solving the $O(n/k)$ subproblems. As a result, overall we will get a more efficient parallel algorithm for the problem. Since we solve all the smaller subproblems in parallel, we achieve overall efficiency.

Datta *et al.* [9] have shown that the general technique of partitioning a point set by imposing a grid on it can be used for designing fast sequential algorithms for clustering problems in higher dimensions. However, in this paper we have considered parallel algorithms for point sets only on the plane. It will be interesting to see whether similar techniques can be used for designing parallel algorithms for clustering problems in higher dimensions as well. The computation of the degraded δ-grid in parallel can be extended to higher dimensions. But the main problem is to design parallel algorithms for the clustering problems in higher dimensions which can be used for solving the $O(n/k)$ subproblems. Currently no such parallel algorithm is known. Datta [8] has shown that the range-tree data structure is very useful for designing parallel algorithms for clustering problems. However, the method for constructing the *range tree* data structure in parallel is limited to two dimensions. A parallel algorithm for constructing the range tree in higher dimensions will help us to design fast parallel algorithms for several clustering problems.

We have designed our algorithms for the CREW PRAM model of parallel computation. However, this model assumes concurrent read when multiple processors can read from the same memory location in the same clock cycle. It is an interesting open problem whether we can design parallel algorithms with similar complexities for the weaker EREW PRAM model where only one processor can read from a memory cell in a single clock cycle. The main difficulty is the access to the range tree data structure. In the algorithms in this paper, we solve the smaller $O(n/k)$ subproblems in parallel and hence all the processors need concurrent access to the range tree data structure which results in concurrent reading of the same cell by multiple processors. We need some new techniques for creating multiple copies of the range tree in parallel so that each processor can search in its own copy of the range tree.

Acknowledgments : The author would like to thank an anonymous reviewer for many helpful comments and suggestions which improved the presentation of the paper considerably. The author would like to thank S. Soundaralakshmi for reading an earlier draft of this paper and suggesting many improvements. Part of this work was done when the author was a visiting professor at the Institut für Informatik, Universität Freiburg, Germany, supported by a grant from Deutscher Akademischer Austauschdienst (DAAD).

References

[1] H. C. Andrews, *Introduction to Mathematical Techniques in Pattern Recognition* Wiley-Interscience, New York, 1972.

[2] T. Asano, B. Bhattacharya, M. Keil, and F. Yao, "Clustering algorithms based on minimum and maximum spanning trees," *Proc. 4th ACM Symp. on Comp. Geom.* (1988), pp. 252-257.

[3] A. Aggarwal, H. Imai, N. Katoh, and S. Suri, "Finding k points with minimum diameter and related problems," *J. Algorithms*, **12** (1991), pp. 38-56.

[4] O. Berkman, D. Breslauer, Z. Galil, B. Scheiber, and U. Vishkin, "Highly parallelizable problems," *Proc. of 21st Annual ACM Symp. on Theory of Computing*, 1989, pp. 309-319.

[5] R. Cole, "Parallel merge sort," *SIAM J. Comput.*, **17**, (1988), pp. 770-785.

[6] S. Chandran, S. K. Kim, and D. M. Mount, "Parallel computational geometry of rectangles," *Algorithmica*, **7**, (1992), pp. 25-49.

[7] S. Chandran, *Merging in Parallel Computational Geometry,* PhD thesis, Department of Computer Science, University of Maryland, College Park, MD, 1989.

[8] A. Datta, "Efficient parallel range searching and partitioning algorithms", *Parallel Algorithms and Applications*, **16**, (2001), pp. 301-316.

[9] A. Datta, H. P. Lenhof, C. Schwarz, and M. Smid, "Static and dynamic algorithms for k-point clustering problems," *Journal of Algorithms*, **19**, (1995), pp. 474-503.

[10] D. P. Dobkin, R. L. Drysdale, and L. J. Guibas, "Finding smallest polygons," In: F. P. Preparata (ed.), Advances in Computing Research, Vol. **1**, Computational Geometry, J. A. I. Press, London, (1983), pp. 181-214.

[11] D. Eppstein and J. Erickson, "Iterated nearest neighbors and finding minimal polytopes," *Proc. 4th ACM-SIAM Symp. on Discrete Algorithms*, (1993), pp. 64-73.

[12] A. Efrat, M. Sharir, and A. Ziv, "Computing the smallest k-enclosing circle and related problems," *Proc. Workshop on Algorithms and Data Structures (WADS), Lecture Notes in Computer Science*, **Vol. 709**, (1993), pp. 325-336.

[13] G. N. Frederickson and D. B. Johnson, "The complexity of selection and ranking in $X + Y$ and matrices with sorted columns," *Journal of Computer and System Sciences*, **24**, (1982), pp. 197-208.

[14] J. A. Hartigan, *Clustering Algorithms*(John-Wiley, New York, 1975).

[15] J. JáJá, *An introduction to Parallel Algorithms* (Addison-Wesley, 1992).

[16] D. B. Johnson and T. Mizoguchi, "Selecting the Kth element in $X + Y$ and $X_1 + X_2 + ... + X_m$," *SIAM J. Comput.*, **7**, (1978), pp. 147-157.

[17] H. P. Lenhof and M. Smid, "Sequential and parallel algorithms for the k-closest pairs problem," Max Planck Institut für Informatik, Technical Report, MPI-I-92-134. August 1992. A preliminary version appeared in *Proc. 33rd Annual IEEE Symp. on Foundations of Computer Science*, pp. 380-386.

[18] J. Matoušek. "On enclosing k points by a circle," *Info. Proc. Let.*, **53**, (1995), pp. 217-221.

[19] N. Megiddo, "Applying parallel computation algorithms in the design of serial algorithms," *J. ACM*, **30**, (1993), pp. 852-865.

[20] F. P. Preparata and M. I. Shamos, *Computational Geometry : an Introduction*, Springer, New York, 1985.

[21] R. Sarnath and X. He, "Efficient parallel algorithms for selection and searching on sorted matrices," *Proc. 6th International Parallel Processing Symposium*, (1992), pp. 108-111.

Deriving biased classifiers for better ROC performance

Hendrik Blockeel and Jan Struyf
Katholieke Universiteit Leuven, Department of Computer Science
Celestijnenlaan 200A, B-3001 Leuven, Belgium
{hendrik.blockeel,jan.struyf}@cs.kuleuven.ac.be

Keywords: data mining, machine learning, ROC analysis

Received: July 15, 2000

Induction of classifiers is an important task in the field of data mining. Classifiers are often evaluated based on their predictive accuracy, but there are disadvantages associated with this measure: it may not be appropriate for the context in which the classifier will be deployed. ROC analysis is an alternative evaluation technique that makes it possible to evaluate how well classifiers will perform given certain misclassification costs and class distributions. Given a set of classifiers, it also provides a method for constructing a hybrid classifier that optimally uses the available classifiers according to specific properties of the deployment context. Now in some cases it is possible to derive multiple classifiers from a single one, in a cheap way, and such that these classifiers focus on different areas of the ROC diagram, such that a hybrid classifier with better overall ROC performance can be constructed. This principle is quite generally applicable; here we describe a method to apply it to decision tree classifiers. An experimental evaluation illustrates the usefulness of the technique.

1 Introduction

During the last decennium, there has been an increasing interest in the fields of *data mining* and *knowledge discovery*. Frawley et al. [9] define knowledge discovery as the non-trivial extraction of implicit, previously unknown, and potentially useful knowledge from data. Data mining is then considered a sub-process of knowledge discovery: the process that involves the actual analysis of the data. Other sub-processes include preparation and cleaning of data, post-processing of data mining results (e.g., visualisation) etc.

The motivation for the increased interest in these fields from both the academic and commercial world is the following. Software and hardware technology have been evolving up to the point where huge amounts of data can now efficiently be collected, stored, and accessed. Consequently, companies and institutions try to store as much data as they can get (building so-called data warehouses), in the hope that these data will somehow be useful later. But while collecting and storing data may be simple, the extraction of useful knowledge from such a data warehouse is not a trivial task. This knowledge is often implicitly available in the data, in the form of patterns that occur, relationships that hold. For instance, assume a store keeps data about visits of clients: when they visited the store, what they bought, etcetera. The knowledge that certain products are often bought together by people, or that certain products are typically bought on specific days or hours, is not explicitly stored in the database but is implicitly available and can be extracted by carefully analysing the data. It is basically this task that knowledge discovery focuses on.

Certain types of knowledge allow one to make predictions, and these are typically consolidated in a so-called *predictive model*. Building predictive models by analysing data is an important task within data mining, and many different approaches have been developed. Linear regression is perhaps the best known example, typically being included in even introductory statistics courses. Other techniques include neural networks [1], induction of decision trees [12, 6], etc.

Predictive modelling approaches typically build a model that minimizes a certain error criterion. For numerical prediction this might be the mean squared error of the predictions, for prediction of categorical values (further referred to as classification) it is typically the proportion of the predictions that is incorrect. In the classification context, the term "predictive accuracy" is popular; this is the proportion of correct predictions (i.e., 1 minus the error).

In some cases, however, the error measure minimised by a data mining algorithm may not be very relevant to the task at hand. We will see further in this text that there are cases where predictive accuracy is not a suitable evaluation criterion. Alternative criteria are then needed. Given that many data mining algorithms optimise predictive accuracy, the question arises whether predictive models produced by standard data mining algorithms can be improved so that they work better according to these alternative criteria. This article provides an answer to that question for one of these alternative criteria: ROC analysis.

The remainder of this article is structured as follows. In Section 2 we describe some of the shortcomings of predictive accuracy as an evaluation measure. In Section 3 we briefly explain the principles of ROC analysis, which is an alternative and more generally applicable evaluation

criterion. We then show in Section 4 how one can improve performance according to ROC analysis by deriving from a single classifier a set of biased classifiers. Experimental results are shown in Section 5, and Section 6 concludes the paper.

2 Evaluation of Predictive Models

In many cases measures such as mean squared error or accuracy adequately reflect the model's usefulness for making predictions, but this is certainly not always the case. To see this, consider the following example. Suppose some disease is spreading in a population, to which 5% of the population is susceptible, while the other 95% is resistant (and it is not obvious who is susceptible and who is not). A cheap vaccine against the disease is available, and ideally it should be provided to all the people who are susceptible to the disease, and to nobody else.

Now compare the following two predictive models. Model 1 predicts everyone to be resistant. Model 2 identifies a subgroup of 20% of the population as susceptible, and there is reason to believe that the 5% truly susceptible persons are all part of this subgroup. The predictive accuracy of Model 1 is then 95%, that of Model 2 only 85%. Thus, accuracy-wise, Model 1 is better than Model 2, even though one might feel that Model 2 contains more information about who is susceptible than Model 1. Consequently, most off-the-shelf data mining systems would tend to build a predictive model that is closer to Model 1 than to Model 2, simply because they aim at maximal accuracy.

The problem here is that not all errors are equally bad. We can express this by assigning a certain cost to different types of errors. For instance, the cost assigned to not giving the vaccine to a susceptible person may be much higher than that of giving the vaccine to a resistant person. One should then aim at constructing models that have a low expected cost. If in the above example the cost of not providing the vaccine to a susceptible person is 10 times that of providing the vaccine to a resistant person, then Model 2 has a lower expected misclassification cost than Model 1.

The expected (average) misclassification cost of a model will of course depend on the costs associated with certain types of errors as well as on the probability of making them. Therefore, this cost depends not only on the model but also on the context in which it will be deployed. This context is not always known in advance; e.g., misclassification costs may not be known in advance, or might vary during deployment.

If the exact parameters of the deployment context are not known in advance, a comparison between models may not give a clear-cut result: whether one model is better than the other may depend on the deployment context. In such a case, it is best to keep both models, and decide which one to use during deployment (possibly switching from one model to another during the deployment phase).

The above considerations provide motivation for the use of a more complex evaluation procedure for predictive models than just their predictive accuracy. Such a procedure is the so-called ROC analysis [11], which has become very popular during recent years. ROC analysis allows us to decide in which contexts, if any, one model will be better than another. It also allows us to clearly see how typical predictive modelling approaches, by aiming for high accuracy models, are concentrating on a small part of ROC space. The main contribution of this paper is that it proposes a computationally cheap method for deriving, from a single model, so-called "biased" models which aim at different deployment contexts than the original model and thus cover a different area in the ROC diagram. The method is worked out in more detail for decision trees, and evaluated in that context.

3 ROC diagrams

The use of predictive accuracy as an evaluation criterion for predictive models has the following two disadvantages: (a) accuracy is unstable with respect to changing class distributions, and (b) it assumes a symmetric misclassification cost (i.e., misclassifying an object of class A as B is an equally bad mistake as misclassifying an object of class B as A).

ROC analysis [11] has been proposed as an alternative evaluation criterion that does not suffer from these problems. A ROC diagram shows how a classifier can be expected to perform in an environment with a given class distribution (not necessarily the one of the training set) and given misclassification costs. On a ROC diagram one can easily see under which circumstances one classifier is better than another (and whether a classifier is better than another classifier in all possible environments). Given that different classifiers may perform better in different environments, it is natural to use ROC analysis to build a hybrid classifier that is optimal in the sense that in each environment it will employ the best classifier from all those available.

First we introduce some notation and terminology. We assume a binary classification problem: objects are to be classified as positive or negative. We represent the true classes as $+$ and $-$ and the predictions as pos and neg. $C_{pos|-}$ represents the cost of misclassifying a negative object as positive; $C_{neg|+}$ the cost of misclassifying a positive object as negative. We assume $C_{pos|+} = C_{neg|-} = 0$. $P(+)$ is the probability that an unseen instance is positive; $P(-)$ is the probability that it is negative. $P(pos|+)$ is the probability that an instance is classified as positive given that it actually is positive; we similarly use $P(neg|+)$, $P(pos|-)$, $P(neg|-)$.

The expected misclassification cost for a single object is then

$$EC = C_{pos|-}P(pos|-)P(-) + C_{neg|+}P(neg|+)P(+)$$

The *true positive rate* TP estimates $P(pos|+)$ and is computed from a test set as $n_{pos,+}/n_+$, i.e., the number of actual positives predicted positive over the number of actual

positives. The *false positive rate* FP estimates $P(pos|-)$ and is computed as $n_{pos,-}/n_-$. Using these estimates, the formula for expected misclassification costs becomes

$$EC = C_{pos|-}P(-) \cdot FP + C_{neg|+}P(+) \cdot (1 - TP)$$

In a ROC diagram the horizontal axis represents FP, the vertical axis TP. Points with the same expected classification cost (iso-cost lines) are on a straight line with slope $C_{pos|-}P(-)/C_{neg|+}P(+)$. Points with the same expected accuracy are on a straight line with slope $P(-)/P(+)$. Note that given $P(+)$ and $P(-)$, expected accuracy can be computed from TP and FP, but not the other way around. Also note that these slopes cannot be negative, assuming positive costs.

The upper left corner of a ROC diagram is the ideal situation ($FP = 0, TP = 1$; i.e., all positives and no negatives are returned). A classifier A is better than B in a certain situation if there exists an iso-cost line for that situation such that A is above the line and B is below it. A *dominates* B if no situations exist where B is better than A; on the diagram A is then to the left and above B. A *set of classifiers S dominates a classifier B* if in every situation there exists a classifier in S that is better than B; on the diagram B is then below the *convex hull* of S.[1] For instance, on the ROC diagram shown in Figure 1 B dominates F (F is to the lower right of B). E is not dominated by B, nor by C, but it is dominated by {B,C}.

For each set of classifiers S, a minimal subset S' can be defined such that no elements e of S' are dominated by $S' - \{e\}$; this is the set of all the classifiers that lie on the convex hull of S. S' can be seen as a hybrid classifier of minimal complexity that still performs at least as good as any of its component classifiers. In Figure 1, the convex hull is formed by the classifiers A, B, and C (together with the points (0,0) and (1,1), which represent classifiers that never, respectively always, predict positive).

When the deployment context is not known in advance, the area under the convex hull is a good measure of the overall quality of this hybrid classifier. In such a case it is reasonable to try to find classifiers that are complementary with respect to their application context; i.e., try to find some that will appear to the far left in the ROC diagram, and others that appear close to the top.

Note that a line of equal accuracy (iso-accuracy line) on the ROC diagram is just a special case of an iso-cost line, one with slope $P(-)/P(+)$. Suppose that an iso-accuracy line in Figure 1 would be relatively steep. Then any learning algorithm aiming at high accuracy would produce a classifier in the neighbourhood of A. It may not make much sense to use many different learning algorithms to improve the ROC convex hull, since all of them will be clustered together near A and an important opportunity for improving the ROC convex hull at the top of the diagram is just not considered. This effect is illustrated in Figure 2.

[1]The convex hull of a set of points is the smallest convex polygon that contains all the points.

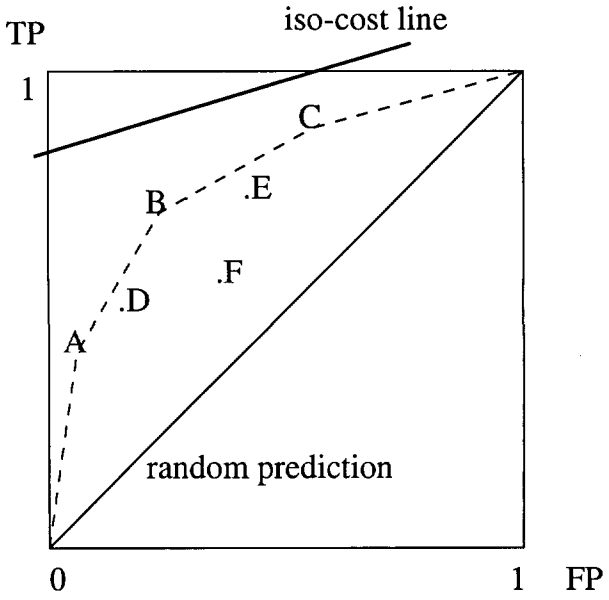


Figure 1: An example ROC diagram.

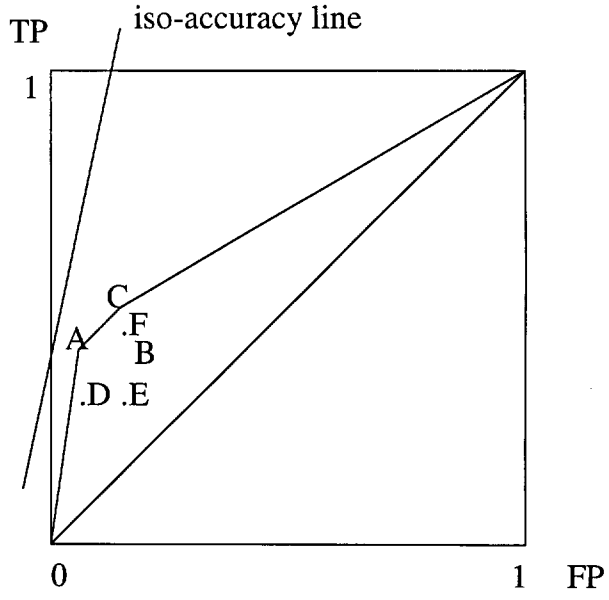


Figure 2: The effect of maximising accuracy, from the ROC point of view. All classifiers are clustering together in the left part of the ROC diagram.

4 Deriving Differently Biased Classifiers from a Single Classifier

The aim of ROC analysis is to optimally employ the information contained in a set of predictive models, so that as good as possible predictions are made in all circumstances. In many cases these classifiers carry more information than just their (FP,TP) coordinates, and this extra information is ignored when just computing the convex hull for these classifiers. The main point of this paper is that in many cases it is easy to exploit this extra information, and this leads to better overall performance.

A classifier A can sometimes be decomposed in such a way that from the pieces new classifiers can be obtained. A trivial example of this is a rule set $S = \{R_1, \ldots, R_n\}$ with rules R_i of the form "IF $cond_1$ AND $cond_2$ AND ... THEN $class$. Any subset of S is a new classifier. As the body of each rule is a set of conditions, the same principle could be used on those. Note that when a rule predicts positive, then removing the rule will decrease TP and FP, while removing a condition from the rule will increase them.

In a similar fashion, a decision tree gives rise to subtrees that are valid classifiers by themselves. This is discussed in more detail below. First we wish to remark that this derivation of new classifiers from the original one bears some resemblance to pruning decision trees or rule sets (see e.g. [12]). Note however that the aim is quite different: pruning a classifier aims at finding a new classifier that is simpler but has comparable or better predictive accuracy, and in this sense improves upon the original one. In our context we are looking for classifiers with properties quite different from the original one.

When a classifier A gives rise to a set of derived classifiers $A_1, \ldots, A_n$, it makes sense to include these classifiers as well on the ROC diagram. In the worst case A dominates all the A_i, but it is probable that some of the A_i are not dominated by A; in fact, the method for deriving classifiers from A could be constructed in such a way that it is unlikely that all A_i are dominated by A.

For decision trees we propose the following method. Given a tree with positive and negative predictions, order the leaves according to some estimated probability of an unseen example sorted into that leaf being positive. This probability estimate could e.g. be the m-estimate [7], a linear interpolation between the ratio of positives in the leaf and some a priori probability p, which is given a weight $m : e = \frac{n_+ + pm}{n_+ + n_- + m}$, with n_+ and n_- the number of positive/negative examples in the leaf. As a value for p it makes sense to use the proportion of positives in the training set. We further choose $m = 1$, which means the estimate is mainly influenced by the proportion of positives in the leaf and only slightly by the size of the leaf.

Now from the decision tree A new decision trees can be constructed as follows. A new tree A_i is the same tree as A, except that for the first i leaves according to the ordering just described (i.e., the i leaves that predict pos with the highest certainty), pos is substituted for the leaf's prediction, and for all other leaves neg is substituted for their prediction. A_0 always predicts neg, A_l (with l the number of leaves in the tree) always predicts pos. The A_i can be pruned separately (for instance, A_1 is essentially just one rule). Figure 3 illustrates this process (the most extreme cases, all positive or all negative predictions, are not shown, and trees are not pruned in the figure).

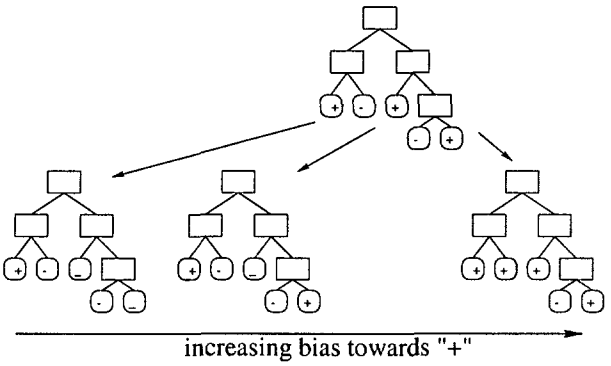


Figure 3: A decision tree, with some biased trees that can be derived from it. Trees more to the left are more biased towards negative predictions.

This idea is not really new; it was used, e.g., in [3] (although not with an m-estimate), from the point of view that decision trees can be considered rank classifiers (i.e. classifiers that give a classification together with some degree of confidence; such classifiers give rise to a curve in the ROC diagram). A similar technique has been used by Gärtner [10] in experiments with the Naive Bayes classifier; different versions of that classifier are constructed, with varying a priori probabilities of having a positive. Since the probability estimates used by Naive Bayes need not be recomputed, we can consider this another example of deriving a set of classifiers from a single classifier without further access to the data.

Thus, there have been approaches similar to the technique we propose here; but to our knowledge this technique has not been described as a methodology as such. Our main point here is that the approach seems quite generally applicable, and should probably be applied in a systematic way. It seems worth investigating, not only for decision trees but also for other representations, what is the best way of deriving a set of classifiers from a single one such that the set will exhibit maximal performance on a ROC diagram. (Note that even for decision trees, what we describe is a rather simple approach and better ones might be devised.)

5 Experimental results

We have evaluated the proposed method for deriving classifiers from a single one on a data set donated by the insurance company SwissLife: the so-called Sisyphus data set. This data set was studied in the context of the European project SolEuNet. We here present some representative results on two classification tasks (a more detailed account

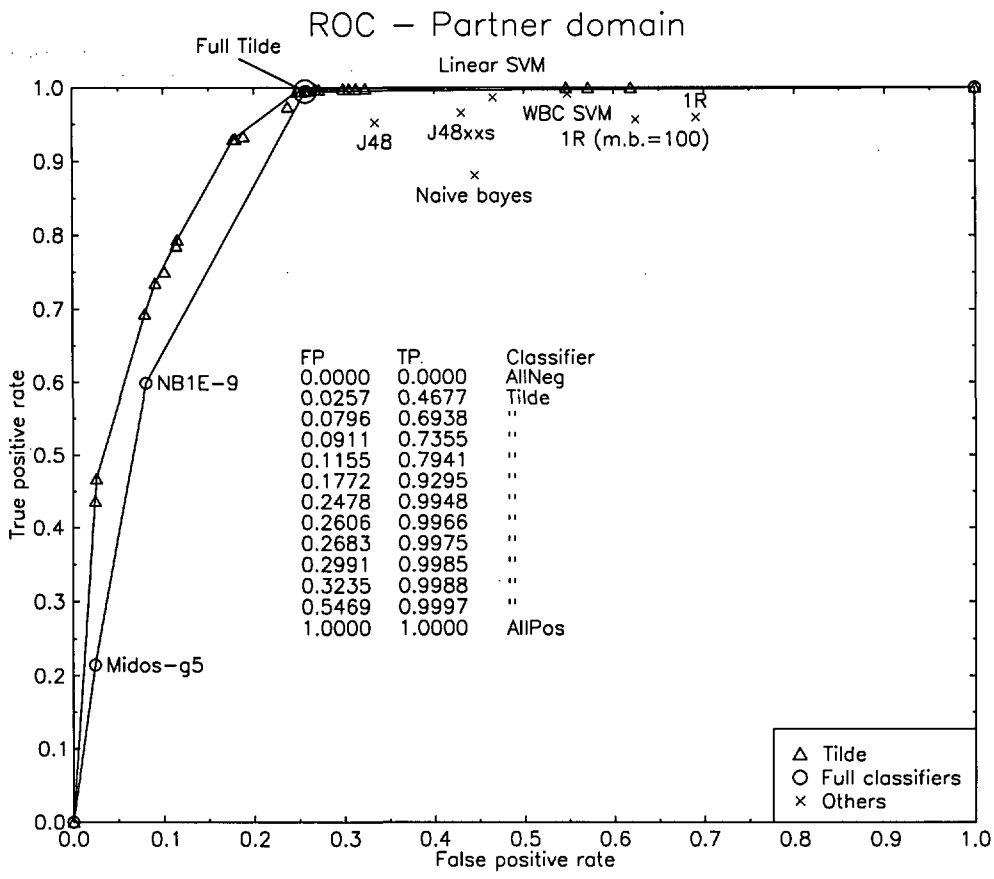


Figure 4: ROC convex hull formed by classifiers derived from a single decision tree, compared with the convex hull formed by another set of classifiers as reported in [10].

of the experiments can be found in [5]). Figure 4 shows a ROC diagram with two convex hulls indicated. One (earlier presented in [10]) is the convex hull of a set of classifiers produced using multiple learning algorithms, without the above methodology. On the second one, classifiers derived from a single decision tree induced by the Tilde system [2] on the original convex hull were included in the diagram. The second convex hull clearly dominates the first one; what's more, in this case it even consists entirely of derived classifiers.

A second result is shown in Figure 5. Here the original tree returned by Tilde was not on the convex hull; it was dominated by a tree produced by the J48 system available in the Weka data mining tool [15]. Yet, the derived classifiers do improve the convex hull.

The proposed methodology was also used in a submission to the Predictive Toxicology Challenge 2000-2001, which was organised in the context of the ECML/PKDD-2001 conference.[2] A more detailed account of the experiments is beyond the scope of this paper but can be found in [4]. The ROC diagrams constructed by the organisers of the challenge (a representative example is shown in Figure 6) are consistent with our claim that learners aiming for maximal accuracy tend to cluster together in the ROC diagram, and that our methodology alleviates this problem to some extent (even though in this specific case the ROC convex hull was not improved much).

It should be pointed out that the proposed technique for deriving a set of decision trees from a single one can improve but never deteriorate the convex hull (as non-optimal derived classifiers would simply not be included), and it is computationally very cheap. For this reason it can only be advantageous to use this methodology, when performing a ROC analysis to obtain overall optimal performance.

6 Conclusions and Related Work

We have discussed a methodology for deriving classifiers from a single classifier in such a way that the ROC convex hull can be significantly improved by including these classifiers. The method we proposed here for decision trees is very cheap and can only give improvements, hence there do not seem to be good reasons for not using it in practice.

[2] http://www.informatik.uni-freiburg.de/~ml/ptc

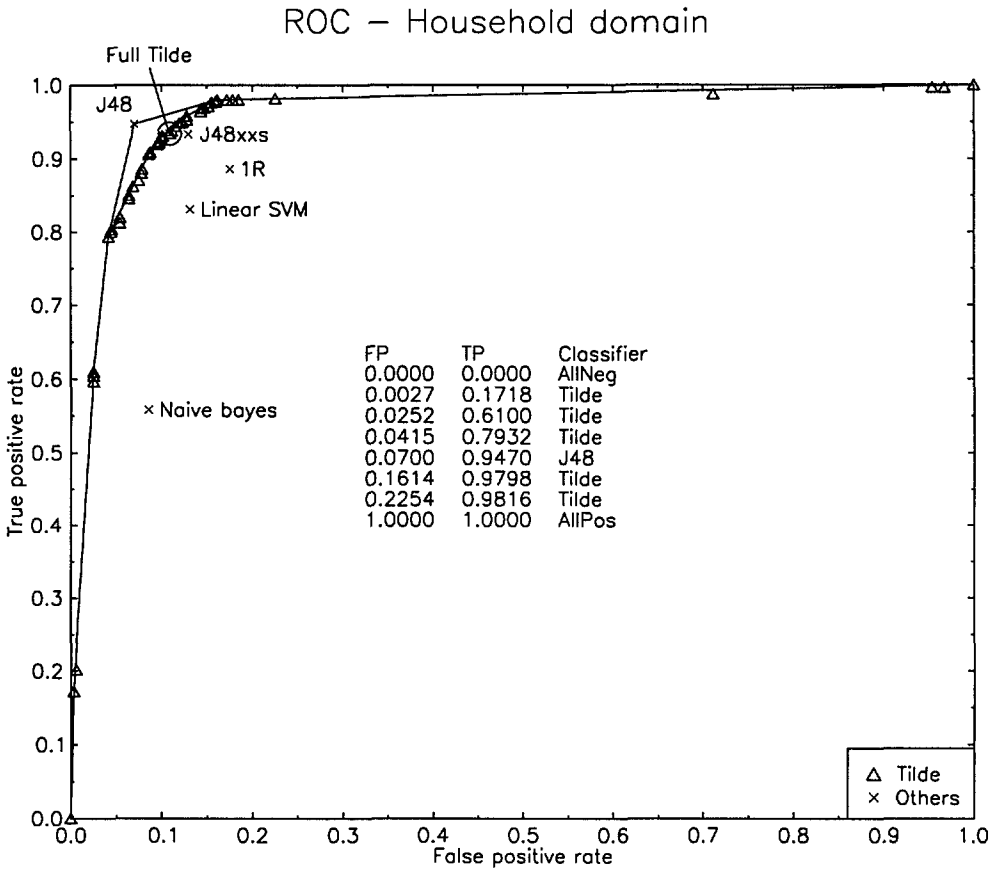


Figure 5: ROC convex hull formed by classifiers derived from a single decision tree, compared with the convex hull formed by another set of classifiers as reported in [10].

The technique of deriving a set of classifiers from a single one has been used before (see, e.g., [3, 10], but in a more or less ad hoc way; it was not explicitly advocated as a methodology. The latter is much more the case in Srinivasan's work on extraction of multiple models in ILP [13], but Srinivasan's work is to a large extent complementary to ours. Srinivasan discusses optimization of a ROC convex hull by running an ILP system repeatedly, varying the background knowledge. This is a quite different (and more expensive) way of producing multiple classifiers. Obviously, both approaches could be combined, producing better results than when either approach is used in isolation.

The problem of cost-sensitive classification has been discussed several times in the literature; see, e.g., [14]. A relatively recent contribution was made by Domingos [8], who discusses a general method for making learning algorithms cost-sensitive. Our method, while motivated from a slightly different perspective, could be seen as serving a similar goal, and a comparison between the methodology proposed here and these other methods seems interesting as future work. Other future work may include more sophisticated methods to derive biased classifiers from trees or from classifier represented in a different format.

Acknowledgements

Hendrik Blockeel and Jan Struyf are respectively a postdoctoral fellow and research assistant of the Fund for Scientific Research of Flanders. The Sisyphus dataset was made available by SwissLife, in the context of the EU's Fifth Framework Project IST-1999-11495 (SolEuNet). The ROC diagram of Figure 6 was constructed and kindly made available to the authors by the organisers of the Predictive Toxicology Challenge 2001 (Stefan Kramer, Christoph Helma, Ashwin Srinivasan, Ross King). Finally, the authors thank Dunja Mladenič and Raymond Kosala for useful comments on this text.

References

[1] C. M. Bishop. *Neural Networks for Pattern Recognition*. University Press, Oxford, 1999.

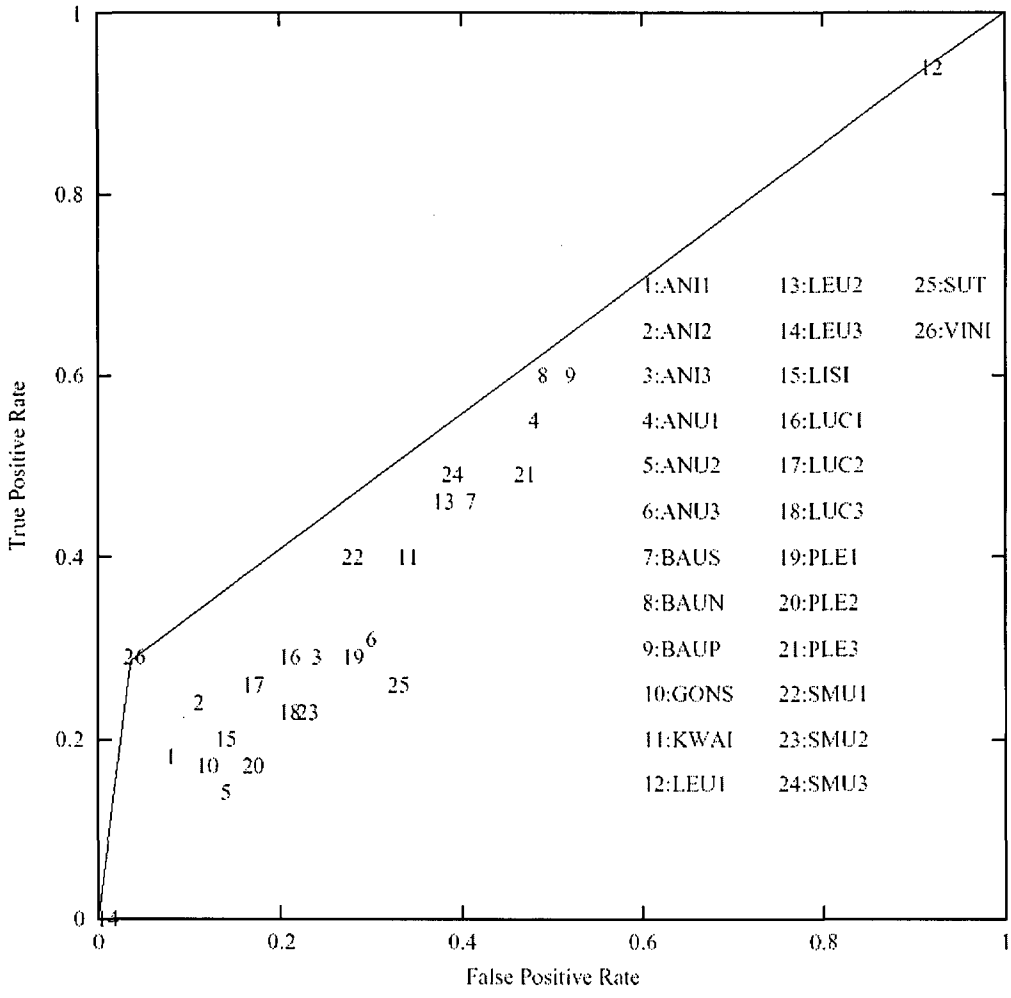


Figure 6: ROC diagram comparing submissions to the Predictive Toxicology Challenge. Note how most classifier are in the lower left corner of the diagram. The most notable exception is number 12, which was produced using the methodology described here.

[2] H. Blockeel and L. De Raedt. Top-down induction of first order logical decision trees. *Artificial Intelligence*, 101(1-2):285–297, June 1998.

[3] H. Blockeel, L. Dehaspe, K. Driessens, N. Jacobs, R. Kosala, J. Ramon, and W. Van Laer. The Leuven submission to the Benelearn-99 competition. In P. van der Putten and M. van Someren, editors, *The Benelearn 1999 Competition*, pages 1–8. Sociaal-wetenschappelijke Informatica, Universiteit van Amsterdam, 1999.

[4] H. Blockeel, K. Driessens, N. Jacobs, R. Kosala, S. Raeymaekers, J. Ramon, J. Struyf, W. Van Laer, and S. Verbaeten. First order models for the Predictive Toxicology Challenge 2001. In *Proceedings of the Predictive Toxicology Challenge Workshop*, Freiburg, Germany, 2001.

[5] H. Blockeel and J. Struyf. Frankenstein classifiers : Some experiments on the Sisyphus data set. In *Proceedings of IDDM-01 - Workshop on Integration of Data Mining, Decision Support, and Meta-Learning*, Freiburg, Germany, 2001.

[6] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

[7] B. Cestnik. Estimating probabilities: A crucial task in machine learning. In *Proceedings of the 9th European Conference on Artificial Intelligence*, pages 147–149, London, 1990. Pitman.

[8] Pedro Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining*, pages 155–164, 1999.

[9] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. Knowledge discovery in databases: an overview. In G. Piatetsky-Shapiro and W. Frawley, editors, *Knowledge Discovery in Databases*, pages 1–27. Cambridge, MA: MIT Press, 1991.

[10] T. Gärtner. Roc analysis on sisyphus data, 2001. Unpublished.

[11] F. Provost and T. Fawcett. Analysis and visualization of classifier performance: comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 43–48. AAAI Press, 1998.

[12] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann series in machine learning. Morgan Kaufmann, 1993.

[13] A. Srinivasan. Extracting context-sensitive models in inductive logic programming. *Machine Learning*, 2001. To appear.

[14] P. Turney. Cost-sensitive learning bibliography, 2000. Institute for Information Technology, National Research Council, Ottawa, Canada.

[15] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, 1999.

A critical note on the role of the quantum mechanical "collapse" in quantum modeling of consciousness

Dejan Raković
Faculty of Electrical Engeneering, P.O.Box 35-54,
11120 Belgrade, Yugoslavia
and
The International Anti-Stress Center (IASC), Belgrade, Yugoslavia
E-mail: rakovic@buef31.etf.bg.ac.yu
AND
Miroljub Dugić
Faculty of Science, Dept. Phys., P.O.Box 60,
34 000 Kragujevac, Yugoslavia and
The International Anti-Stress Center (IASC), Belgrade, Yugoslavia
E-mail: dugic@uis0.uis.kg.ac.yu

Keywords: Consciousness, quantum modeling, neural networks, quantum measurement, "collapse", open quantum systems

Received: September 18, 2000

We give a brief account on the existing strategies in quantum-mechanical approaching the problem of consciousness. To a list of distinguished approaches we add the next plausible notion: when treated quantum-mechanically, consciousness should be modeled as an open quantum system. This notion is tightly connected to the von Neumann's "collapse" ("wave packet reduction"). Here we strongly emphasize that the problem of the "collapse" cannot be considered resolved within the quantum mechanics of open quantum systems (or: decoherence theory). Some clues in this regard are briefly outlined.

1 Introduction

The complex subject of *consciousness* does not stop drawing attention of the general scientiffic community. Particularly, it was suggested that some manifestations of consciousness might have a deeper physical origin, which requires the use of the laws and methods of *quantum physics*. Therefore, the problem of consciousness partially becomes a matter of investigation within, e.g., the quantum-mechanical formalism *and* interpretation, still with the stringent "constraints" coming from the phenomenological data.

The purpose of this paper is to put forward some critical remarks concerning the above distinguished approach to the problem of consciousness, with an emphasis on the role of the quantum-mechanical "wave packet collapse (reduction)" (further: "collapse").

In the next Section we briefly and critically overview the arguments concerning the quantum-mechanical nature of consciousness, adding a new point in this concern. In Section 3 we critically discuss the role of the quantum-mechanical "collapse" within the quantum theory of consciousness. Section 4 is discussion and conclusion.

2 Some critical remarks on the quantum nature of consciousness

There are significant efforts in establishing the quantum (e.g., quantum-mechanical) nature of consciousness. It seems convenient here to give a brief overview of some of the existing results and statements, still without ambition to be exhaustive. (Note: the main criterion for making the list below, is that the authors more-or-less explicitly argue for the use of quantum formalism(s) in modeling of consciousness.)

– The brilliant brief review-article by Jibu and Yasue [10] shares belief with the *"reductionists"*, who (sometimes noncritically) assert that the physical laws of the complex (many-particle) systems can be traced back to the fundamental laws of quantum physics. Originally, the above mentioned "reductionism" is due to Ricciardi and Umezawa [19], and is stated as [10]: "deeper understanding of natural phenomena... [can be obtained] only after having an adequate grasp of the theories derived from the first principles of physics"[1].

- The experience with the Microwave Resonant Therapy (MRT), as a medical cure, points out a rather intriguing results: (1) sharply-resonant sensory response of the disordered organism to small changes of the external microwave (MW) radiation frequency (0.01-0.1%); (2) low-intensity

(down to 10^{-9} W/cm^2) and low-energy (down to 10^{-4} eV) of the biologically efficient MW radiation rather below the thermal noise effect ($\sim 10^{-2}$ W/cm^2 and $\sim 10^{-2}$ eV); and (3) negligible MW energy losses at its propagation for significant distances down acupuncture meridians ($\sim$ 1m) from the exposed acupuncture points. Such *quantum-like* characteristics of MRT inspired Sit'ko [20] and coworkers to *propose a quantum physics of alive*. In the framework of their model, acupuncture meridians might be related to eigenfrequencies and spatio-temporal eigenwaves of every biological quantum system, being its individual characteristic. On these grounds Raković et al [17] suggest that healthy condition might be considered as an absolute minimum (ground state) of the nonlocal self-consistent macroscopic quantum potential of the organism, some disorders of an acupuncture system corresponding to higher minimums of the (spatio-temporally changeable) potential hypersurface in energy-configuration space, which possibly explains the higher sensory responses of the more excited (more disordered) acupuncture system, and poor MRT sensory response of the healthy acupuncture system being already in the ground state - this picture being very close to those of *associative neural networks* in their energy-configuration spaces, and to pattern recognition as convergence of the neural networks memory patterns.

– As Peruš [16] points out, the classical neural-network model of brain activities *does not prove sufficient* for reaching the subtlety and unity of consciousness. Besides, he gives some details with regard to *formal analogy* between the "quantum processes" in Feynman's propagator version of quantum mechanics and "neural-network processes" of Hopfield's oscillatory holographic neural network (cf. Section 6 therein), particularly emphasizing the "uncertainty analogy" - i.e., analogy between the quantum uncertainty, and the uncertainty inherent to the neural-network processing. These similarities of the quantum and neural network pictures might not be only formal, supporting the EM/ionic holographic microwave/ultralowfrequency (MW/ULF) *quantum neural network-like* function of the acupuncture system, and its essential relation to (complex-valued quantum relativistic) *consciousness* - as strongly suggested from Raković's [18] modeling of altered and transitional states of consciousness.

– The lucid and illuminating analysis by Penrose [15], concerning the noncomputable nature of human (mathematical) understanding points to the, so-called, "objective reduction (collapse) - OR", as a possible *physical* basis of human faculty of understanding. Actually, in a long run of argumentation (and speculation), Penrose meets the problem (: where to look for noncomputable physical law(s)), and *makes a guess* in this concern (appointing OR), which is suggested to fall within the quantum gravitation theories.

– The quantum holography modeling of the "brain/mind" by Marcer and Schempp [13] bears *positivistic atitude*. Actually, they point to some advantages of the quantum-mechanical treatment of some speciffic subjects of consciousness. Among else, there are the next "subjects": both digital and analogue[2] computation, the problem of unity of consciousness, and that some *non-local "anomalies" of consciousness* (such as the, so-called, altered states of consciousness) cannot be properly physically explained, *unless* – as advocated by Raković [18] - they are somehow founded on the *quantum-mechanical nonlocality*.[3]

It is worth stressing that, relative to the formulation of quantum mechanics, the above distinguished approaches to quantum nature of consciousness, bear *plausibility*. Actually, the venture of quantum physics begins with inability for obtaining the classical-physics explanations of some experimental data - which is widely known as the "physical basis of quantum mechanics". On the other hand, *still, we do not deal with either complete phenomenology, or classical-physics theory of consciousness*. Therefore, from the purely methodological point of view, the quantum nature of consciousness appears just as a *working hypothesis* in the physical approach to consciuousnness.

On this line of thought and plausibility, and this is the main point of this Section, we add (to the above list) the point given below, still not fully distinguished in the literature. First, we point to the next characteristic of the *neural network models* of the brain-activities (which appear as *ultimate physical origin* of consciousness): it is not entirely clear what should (or even could) be the state (and dynamics) of an isolated neural network, i.e., the dynamics of a neural network is *substantially* determined by external stimulation (cf., e.g., Ref. [10, 16]).

When expressed in physical terms, the above characteristic of neural networks reads as follows: the physical behaviour (i.e. dynamics) - but, probably, even existence (cf. Jibu and Yasue [10]) - of a neural network is substantially determined by interaction (cf. "stimulation" above) with its environment (cf. "external"[4] above). And this justifies the next suggestion: *if treated quantum-mechanically, the neural networks should be modelled as open quantum systems*.

Within quantum mechanics of open quantum systems, the above remark links the subject of consciuosness with the subjects of "decoherence" [1, 11, 24-26, 5, 6], the "collapse" [2, 4, 8, 12, 15-18, 23-25], and with the (quantum) "chaos" [2, 8, 26]. Here we shall only discuss the second issue - as it is stated in title of this paper.

We conclude this Section by noting that, for some phenomenological data, and according to some physical reasoning, it seems plausible to *adopt the hypothesis* that consciousness, i.e., some of its physical manifestations, might have the quantum-mechanical origin. As a new, nontrivial point of our argumentation in this respect appears *suggestion* that, whatever it might be, a physical model of consciusness should be considered as *an open quantum system*.

3 The problem of the "collapse" unresolved

As it was distinguished in Introduction: once adopted, the hypothesis of quantum-mechanical nature (or physical ba-

sis) of consciousness implies that further investigation falls into the "machinery" of the quantum-mechanical formalism. This way naturally comes to scope the problem of the "collapse", which can be considered as virtually independent on a (quantum-mechanical) model of physical system (here: brain activities and, in a long run, of consciousness); i.e., the "collapse" is a general issue of quantum mechanics.

Still, with respect to the told in Section 2, one should note that the "route" for approaching the "collapse" is, at least, two-fold. On one side, one may consider the "collapse" as an *emergent property* of the quantum systems, which can be formally "settled down" by the famous "projection postulate" [23]. On the other side, in Section 2 it was argued that the physical model of consciousness should be considered as an open quantum system - which is substantially different physical situation; let us briefly justify this notion.

As it is well known, the "projection postulate" appears as an *ad hoc* rule for obtaining (reproducing) the results of the generalized (and simplified) "scheme" of the (idealized) quantum measurement situations. This characteristic of the "projection postulate" is the main reason for extensive search for the more physical solution of the problem at hand. As a result of these efforts appears the, so-called, quantum mechanics of open systems [1, 2, 5-8, 11, 12, 14, 24-26]. And it is sometimes (noncritically) adopted the statement, according to which the "collapse" naturally follows as a consequence of interaction of a quantum system with its environment.

Unfortunatelly, and this is the very point of this Section, this statement *cannot* be considered physically correct. More precisely: *the very openess of a quantum system does not prove sufficient for obtaining the "collapse" as an objective physical process*; in terms of the quantum measurement theory [12, 25]: the openess of a quantum system *sharpens*, but *does not resolve* the quantum measurement problem. Those readers interesting in the "technical" details should refer to Appendix I below (and references therein), in which it is implicit that the problem of "collapse" can be traced back to the (von Neumann's [23]) hypothesis of the universal validity of the Schrodinger law. Since, on the other side, this hypothesis seems generally adopted (except in the theory of Dugić [7]), and particularly in the quantum modeling of consciousness [10, 13, 16, 18], one may conclude that the "collapse" does not appear more usefull in quantum modeling of consciousness, than it is the case within the quantum measurement theory.

Therefore, one ends with the "pessimistic" statement that, *as yet, the problem of "collapse" cannot be considered resolved within the quantum mechanics of open systems*, thus necessarily making the statements, such as, that the "collapse" makes explicit the quantum eigenstates which bear apparent macroscopic meaning (e.g., the patterns of the neural networks), somewhat vague.

4 Discussion

The main point of this paper is the next statement: *if* (as it seems to be the case [13, 15, 16, 18]) the role of the "collapse" within the quantum modeling of consciousness should be considered substantial, then the quantum physics of consciousness meets serious obstacles - as distinguished in Section 3.

However, the "pessimistic" conclusion of Section 3 sholud not be considered as "the end" of the "story". Furthermore, the experience with the modern quantum mechanics of open systems, likewise some recent breakthrows [2, 7, 8, 15, 18], might open the new avenues with this regard[5]. It seems very probable that, whatever the "possibilities" might be, they will exhibit the next general characteristics: (i) More-or-less significant change of the physical meaning of "collapse", and (ii) Abandoning of the (hypothesis of) universal validity of the Schrodinger law.

Needless to say, these expectations express some subjective point of view, but, nevertheless, there are some noncontroversial arguments in their support; here we shall briefly distinguish some of them, but the interested reader should refer to Appendix II, below.

The concept of "collapse" (as stated by von Neumann [23]), has already survived nontrivial change, basically for the original "definition" is not physically clear. For instance, nowadays one meets the concept of the "objective reduction (collapse) - OR", as introduced by Penrose [15]. As an intermediate step from von Neumann to Penrose, one can meet the different attempts, e.g., the theory of Raković [18], all of them participating in physical objectification of the "collapse". The "objectification" assumes the possibility of physical *observability* of the "collapse"(i.e., of the OR) - which puts speciffic emphases and limitations on the objectification of the "collapse", and of consciousness (non?)involved in it. Certainly, this is an open task whose perspectives, as yet, can hardly be predicted.

In conclusion, we shortly emphasize that the role of the "collapse" in physical modeling of consciousness bears ambiguities, which come from the unresolved problem of the "collapse" within the quantum measurement theory, and particularly within the quantum mechanics of open systems. Whether the interplay between the physics of consciousness, and of the quantum measurement theory can provide some progress in this concern is, as yet, an open question to be answered, we believe, in relatively near future.

Appendix I

The objective of modern quantum measurement theory is the composite system "(quantum) object+apparatus+environment (o+A+E)". In this system, only the ("classical") apparatus (A) interacts with both, the (quantum) "object (o)", and with the (apparatus') environment (E). Note: interaction with environment is considered *unavoidable and continuous in time*, and is

usually assumed to be strong; this is why the apparatus - within the quantum-mechanical treatment - is refered to as an *open* system. The interaction with the "object" is considered continuous, but only in a finite (macroscopically short) time interval.

As in the von Neumann's quantum measurement theory [23], one adopts the next *postulate/hypothesis*: for the "whole", o+A+E, one should consider the Schrodinger law (equation) valid, i.e. :

$$\Psi_{o+A+E} \equiv \hat{U}\Psi_o \otimes \chi_A \otimes \Phi_E = \sum_i C_i \phi_{oi} \otimes \chi_{Ai} \otimes \Phi_{Ei}, \tag{I.1}$$

where the initial ("pure") quantum state of the object, $\Psi_o = \sum_i C_i \phi_{oi}$, $\hat{U}$ denotes the unitary in time evolution operator (the Schrodinger law), "$\otimes$" denotes the "tensor (direct) product" of states; the states ϕ_{oi} constitute an orthonormalized basis in the Hilbert state space of the "object".

As it is usually distinguished, *linearity* of $\hat{U}$ keeps the linear superposition involved in Ψ_o, and leads to the *correlations* of states of the subsystems, as explixit on the r.h.s. of (I.1).

On the other side, for the practical purposes, each subsystem, o, A, E (or o+A, A+E), can be formally represented by the "reduced density matrix", $\hat{\rho}_o$, $\hat{\rho}_A$, $\hat{\rho}_E$ (and so on), defined as:

$$\hat{\rho}_o = \sum |C_i|^2 \phi_{oi} \phi_{oi}^{\dagger}, \tag{I.2a}$$

$$\hat{\rho}_A = \sum |C_i|^2 \chi_{Ai} \chi_{Ai}^{\dagger}, \tag{I.2b}$$

and so on; "†" denotes the "hermitian conjugate".

Therefore, as regards the "object", quantum measurement formally reduces to the transition:

$$\Psi_o \Psi_o^{\dagger} \rightarrow \hat{\rho}_o, \tag{I.3}$$

which is the *mathematical form of the "colapse"*. As regards the "whole", the same reads:

$$\Psi_{o+A+E}\Psi_{o+A+E}^{\dagger} \rightarrow \hat{\rho}_{o+A+E} =$$

$$\sum_i |C_i|^2 \phi_{oi}\phi_{oi}^{\dagger} \otimes \chi_{Ai}\chi_{Ai}^{\dagger} \otimes \Phi_{Ei}\Phi_{Ei}^{\dagger}. \tag{I.4}$$

[But *neither transition can be obtained via the Schrodinger law (equation)*. And this is the famous "quantum measurement problem".]

The very point of this appendix are the next nonequalities:

$$\Psi_{o+A+E}\Psi_{o+A+E}^{\dagger} \neq \hat{\rho}_o \otimes \hat{\rho}_A \otimes \hat{\rho}_E \tag{I.5a}$$

$$\Psi_{o+A+E}\Psi_{o+A+E}^{\dagger} \neq \hat{\rho}_{o+A+E} \tag{I.5b}$$

$$\hat{\rho}_{o+A+E} \neq \hat{\rho}_o \otimes \hat{\rho}_A \otimes \hat{\rho}_E \tag{I.6}$$

The *physical meaning* of (I.6) is probably obvious, and leads to the main point of Secion 3: After the collapse (cf. the r.h.s. of (I.4)), *no* subsystem (o, A, E) *can be considered to be in a definite quantum state* ($\hat{\rho}_o$, $\hat{\rho}_A$, $\hat{\rho}_E$). Otherwise, one would be able to put equality, "=", in Eq. (I.6): $\hat{\rho}_{o+A+E} = \hat{\rho}_o \otimes \hat{\rho}_A \otimes \hat{\rho}_E$; i.e., the assertion "the object is in a state $\hat{\rho}_o$, *and* the apparatus is in the state $\hat{\rho}_A$, *and* the environment is in the state $\hat{\rho}_E$", is *by definition* equivalent with the assertion "the composite system o+A+E is in the state $\hat{\rho}_o \otimes \hat{\rho}_A \otimes \hat{\rho}_E$.

This fact, well known in the foundations of quantum mechanics and quantum measurement theory (cf. also the recent preprint of d'Espagnat [4]), is a *feature of the "quantum nonseparability"* [3], and justifies to refer to $\hat{\rho}_o$, $\hat{\rho}_A$ and $\hat{\rho}_E$, as to the *"improper mixtures"* [3]; formally exactly the same situation appears in the famous "EPR paradox".

One may therefore conclude that the very openess of a quantum system (apparatus) does not by itself appear sufficient concerning both, establishing the "collapse" as an objective physical process (which would "offer" $\hat{\rho}_o$, $\hat{\rho}_A$, $\hat{\rho}_E$, as the objective quantum states), likewise the resolution of the quantum measurement problem. That is, *as regards the "collapse"*, the quantum mechanics of open systems suffers from exactly the same shortcomings as the original von Neumann's quantum measurement theory [23].

Appendix II

The physical contents of the term "collapse" have changed, especially in recent progress of this topic. And the main object of change is the *role of consciousness*.

In the original von Neumann's theory [23] the "collapse" is ultimatively refered to consciousness. More precisely: to *self-consciousness* of the observer, who is the final element of the von Neumann's [23] "chain", "(quantum) object + apparatus + observer (o+A+O)". And the prefix "self" is substantial in this respect: the ability of the observer to be aware of his own (quantum) state is the unique cause of the "collapse". However, the *necessity* of the *self*-consciousness is usually seen as *nonphysical* solution of the problem.

Recent progress in physical modeling of consciousness distinguishes the physically different point of view. E.g., in the physical model of consciousness of Raković [18], one meets consciousness interpreted as a physical "condition" sufficient for "channeling the collapse". As yet, it is not completely understood if consciousness (note: there is no the *self*-consciousness) appears *necessary*[6] in this respect. We interpret this model as an intermediate step in obtaining a fully physical solution of the problem of "collapse" (i.e., in obtaining full objectification of the "collapse"). To this end, one should note that, once the physical consequences of the model [18] would be fully understood, it might prove that (in a long run) the model appears, by itself, to provide us with the basis of the full physical solution.

As a new prospect in objectification of the "collapse" appears the program formulated by Penrose [15]. Therein one meets the "objective reduction (collapse) - OR", as a *real, objective physical effect*, which *does not call for consciousness, at all*! However, this is, at the moment, just a program, within which the "objective collapse" appears

merely as a convenient phrase.

References

[1] Caldeira, A.O. and A.J. Leggett, 1983, Quantum Tunneling in a Dissipative System, Ann. Phys. (N.Y.) **149**, 374

[2] Cvitanović, P., I. Percival and A. Wirzba (editors), 1991, "Quantum Chaos- Quantum Measurement", Academic Publishers, Dodrecht

[3] D'Espagnat, B., 1971, "Conceptual Foundations of Quantum Mechanics", Reading, MA: Benjamin

[4] D'Espagnat, 1998, Reply to Aharonov and Anandan's "Meaning of the Density Matrix", Los Alamos E-print archive, *quant-ph 9804063* (unpublished)

[5] Dugić, M., 1996, On the Occurrence of Decoherence in Nonrelativistic Quantum Mechanics, Physica Scripta **53**, 9

[6] Dugić M., 1997, On Diagonalization of a Composite-system Observable. Separability, Physica Scripta **56**, 560; J. Res. Phys., **27** (1998) 141.

[7] Dugić, M., 1998, Many Time Interpretation of the Quantum Measurement Process, Los Alamos E-print archive, *quant-ph 9810029* (unpublished)

[8] Grigolini, P., 1993, Quantum Mechanical Irreversibility and Measurement, World Scientific, Singapore

[9] Jahn, R.G., 1982, The persistent paradox of psychic phenomena: An engeneering perspective, Proc. IEEE 70, pp. 136-170; and references therein

[10] Jibu, M. and K. Yasue, 1997, What is Mind? - Quantum Field Theory of Evanescent Photons in Brain as Quantum Theory of Consciousness, Informatica, **21**, 471

[11] Joos, E. and H.D. Zeh, 1985, The Emergence of Classical Behaviour through Interaction with the Environment, Z. Phys. **B59**, 223

[12] Leggett, A. J., The Superposition Principle in Macroscopic Systems, in Foundations of Quantum Mechanics in the Light of New Technology, eds. S. Nakajima et al, (World Scientific, Singapore, 1996) p. 35

[13] Mercer, P.J. and V. Schempp, 1997, Model of the Neuron Working by Quantum Holography, Informatica, **21**, 517

[14] Omnés, R., 1994, The Interpretation of Quantum Mechanics, Princeton University Press, Princeton

[15] Penrose, R., 1994, Shadows of the Mind. A Search for the Missing Science of Consciousness, Oxford University Press, Oxford

[16] Peruš, M., 1997, System-Processual Background of Consciousness, Informatica **21**, 491

[17] Raković, D., and Z. Jovanović-Ignjatić,1998, Microwave resonance therapy and acupuncture : New prospects for traditional medicine, 14th Ann. Int. Symp. Acup. and Elec. Therap., New York, October 22-25; Z. Jovanović-ignjatić and D. Raković, A review of current research in microwave resonance therapy: Novel opportunities in medical treatment, Acunpt. and Electro-Therap. Res., The Int. J., in press (1999); Raković D., Z. Jovanović-Ignjatić, D. Radenović, M. Tomašević, E. Jovanov, V. Radivojević, Ž. Martinović, M. Car, and L. Škarić, 1999, An overview of microwave resonance therapy and EEG correlates of microwave resonant therapy and other consciousness altering techniques, 10th Int. Montreux Congress on Stress, Montreux, February 28 - March 6 (to be published in Electro- and Magnetobiology)

[18] Raković, D., Brainwaves, neural networks, and ionic structures: Byophysical model for altered states of consciousness, in D. Raković and Dj. Koruga, eds., Consciousness, Scientific Challenge of the 21st Century (ECPD, Belgrade, 1995), pp. 291-316; D. Raković, 1997, Hierarchical neural networks and brainwaves: Towards a theory of consciousness, in Lj. Rakić, G. Kostopoulos, D. Raković and Dj. Koruga, eds., Brain and Consciousness: Proc. ECPD Workshop, ECPD, Belgrade, p. 189; D. Raković, 1997, Prospects for conscious brain-like computers: Biophysical arguments, Informatica **21**, p. 507; D. Raković and M. Dugić, 1998, Consciousness mediated quantum gravitational collapse via generated wormholes: >From macroscopic biophysical to microscopic quantum arguments, in P.P. Wang, ed., Proc. Joint Conf. Information Sciences, Vol. 2: 3rd Int. Conf. Comput. Intell. and Neurosci., Durham, NC, p. 265; and references therein

[19] Ricciardi, L.M. and H. Umezawa 1967, Brain and physics of many-body problems, Kybernetica, **4**, 44

[20] Sit'ko, S.P. and L.N.Mkrtchian, 1994, Introduction to Quantum Medicine, Pattern, Kiev

[21] Swami Rama, 1978, Living with the Himalayan Masters, Himalayan Int. Inst. of Yoga Sci. and Phil., Holesdale, PA; K.C. Markides, 1990, Fire in ther Heart. Healers, Sages and Mystics, Paragon House, New York

[22] Thorne, K.S., 1994, Black Holes and Time Warps: Eistein's Outrageous Legacy, Picador, London, Ch. 14; and references therein

[23] V. Neumann, J., 1955, Mathematical Foundations of Quantum Mechanics, Princeton University Press, Princeton

[24] Zurek, W.H., 1982, Environment-induced superselection Rules, Phys. Rev. **D26**, 1862

[25] Zurek, W.H., 1993, Preferred States, Predictability, Classicality and the Environment-Induced Decoherence , Prog. Theor. Phys. **89**, 281

[26] Zurek, W.H. asd J.P. Paz, 1994, Decoherence, Chaos and the Second Law , Phys. Rev. Lett. **72**, 2508

5 Footnotes

[1]Still, although it is strongly suggested by Jibu and Yasue that " 'popular quantum mechanics' ... never provides us with a proper scientific understanding of the deep question 'What is consciousness?' but with a fatal misunderstanding", the formal part of the model of "quantum brain dynamics" *really is quantum mechanical*! Actually, cf. Appendix A therein, the "Quantum Electrodynamics of Ordered Water" is *completely* formulated within the *Hamiltonian formalism*, and really falls within the methodology of *modern quantum mechanics of open systems* - as it can be directly seen by comparing to the papers of, e.g., Zurek [24-26], and especially Dugić [6].

[2]Here, by "analogue computation" we mean the, so-called, "quantum computing", which is really a computational process which *simulates quantum mechanical dynamics* of the actual system.

[3]By (quantum) non-locality we mean non-locality as it is defined within the famous "EPR paradox", and which is a straightforward consequence (with previously postulated the "collapse") of the quantum correlations (cf., e.g, the r.h.s of (I.1)).

[4]Here, "external" should not be literally understood. For instance, for a *sub*network, the other subnetworks of the same, whole network, might appear "external". For this reason, in analogous physical situations, there appear the concepts (the terms - cf. e.g., Omnés [14]) "external environment", and "internal environment", only requireing that "environment" is sufficiently "big" (i.e., that the limit $N \to \infty$ is legitimate; N is the number of particles constituting the "environment").

[5]At the moment, there are the three projects in this concern, coming from the diferent areas of modern physics. The first project refers to the novel results [2, 8] of the encreasing field of quantum chaos. On the other side, the concept of the "objective reduction (collapse) - OR", as defined by Penrose [15], might have the *quantum-gravitational* origin in the, so-called, wormholes, as advocated by Raković [18]. Finally, the recent proposal of Dugić [7] calls for a *new paradigm* - nonunique physical time - and (in paralel to the ergodicity considerations [8]) being refered to the single (individual) physical systems (quantum "objects" and "apparatuses"), as oposed to the stabdard ensemble point of view.

[6] This might be deeply connected with the role of "collective consciousness" (as a composite quantum state Φ of all "individual consciousness" ϕ_k: $\Phi \sim \Pi_k \phi_k$) in quantum theory of measurement [18], where "collective consciousness" with its assembling (equivalent to convergence of Feynman's propagator quantum mechanics to one of its propagators, Φ_i) contributes in channeling reduction of initial wave function Ψ into one of (possible) probabilistic eigenstates Ψ_i - which implies that "collapse" could be related with generation of microparticles' local wormholes in highly noninertial microparticle's interactions in quantum measurement situations [18] (fylly equivalent to extremely strong gravitational fields according to the Einsten's Principle of equivalence, when relativistic generation of wormholes is predicted [22]): so, in Penrose's gravitationally induced collapse [15] the very mechanism for this process could be continuous opening and closing of local microparticle's wormholes, addresses oftheir exits being related (probabilistically) to one of (possible) eigenstates Ψ_i of corresponding quantum system - and everything being related to corresponding (probabilistic) assembling Φ_i of "collective consciousness" [18]. The question how it is possible that these highly noninertial microparticles' processes with inevitable relativistic generation of the wormholes were not taken into account within quantum mechanics which is yet extreemly accurate theory - might be answered as it was, but within the ad hoc von Neumann's "projection postulate" [23] to account for quantum-mechanical "wave packet collapse" in quantum measurement situations (implying also that this ad hoc prcedure is based on quantum gravitational phenomena, being on deeper physical level than quantum mechanical ones)! On the other hand, the nonlocality of "collective consciousness" provides an additional evidence that Quantum mechanics is nonlocal theory - otherwise demonstrated by tests of Bell's inequalities and the Einstein-Podolsky-Rosen "paradox".

It should be also pointed out that the above "collective consciousness"' assembling $\Phi_i (i = 1, 2, 3 \ldots)$ in quantum theory of measurement should be interpreted as purely probabilistic (with relative frequency of their appearance given by quantum-mechanical probability $|a_i|^2$ of realization of corresponding microparticle's eigenstates Ψ_i), depending not on the previous history of the repeatedly prepared quantum system. However, this might not be the case for "individual consciousness" assembling, being history-dependent deterministic one (resulting in deterministic convergence of the consciousness-related-acupuncture EM/ionic MW/ULF oscillatory holographic Hopfield-like associative neural network to the particular attractor in the potential hypersurface, or equivalently to deterministic convergence of Feynman's propagator quantum mechanics to the particular propagator corresponding to ϕ_k, fixedly determined by "individual consciousness"), implying that strong preferences in individual futures might exist, governed by individual mental loads; the same may apply to collective futures too, governed by interpersonal mental loads [18].

Solving path problems on a network of computers

Robert Manger
Department of Mathematics, University of Zagreb
Bijenička cesta 30, 10000 Zagreb, Croatia
Phone: +385 1 460 5750, Fax: +385 1 468 0335
E-mail: manger@math.hr

Keywords: graph theory, path problems, distributed algorithms, network computing, parallel virtual machine (PVM), experiments

Received: October 11, 2000

Path problems are a family of optimization and enumeration problems, which reduce to generation or comparison of paths in graphs. Some examples are: checking the existence of a path between two given nodes, finding the shortest or the most reliable path, finding the path of maximum capacity, listing all paths, etc. In this paper we propose a distributed algorithm for solving path problems in directed graphs. The algorithm can easily be implemented on a network of computers. In the paper we also prove the correctness of the proposed algorithm. Next, we describe an actual network implementation based on the PVM package. Finally, we list some experimental results illustrating the performance of the implemented algorithm.

1 Introduction

Path problems [11] are a family of optimization and enumeration problems, which reduce to generation or comparison of paths in directed or undirected graphs. Such problems are encountered in operations research, computer science, electronics and elsewhere. Some well-known examples are: checking the existence of a path between two given nodes, finding the shortest or the most reliable path, finding the path of maximum capacity, listing all paths, etc.

Path problems can be solved in many ways, by sequential or parallel algorithms. Most of the existing parallel algorithms [4, 6, 8] require exotic computer architectures, such as a shared-memory multiprocessor or a synchronized mesh of processors. Distributed algorithms [7, 10, 12] run on ordinary computer networks, but they are usually specialized for certain types of problems or certain graph configurations.

The aim of this paper is to propose, justify and evaluate yet another *distributed* algorithm for solving path problems. Contrary to the previously cited works, our solution is fairly simple. Also, the solution is general in the following three ways. First, the algorithm has been designed for a ring of processors with private memories [2, 3], which means that it can be implemented on various distributed-memory architectures including an ordinary network of computers. Next, the algorithm works within an abstract algebraic framework, so that it can be applied to different types of path problems (e.g. path existence, shortest distances, ... , etc). Finally, problems in all types of directed and undirected graphs can be solved, although the considered version of the algorithm is primarily intended for directed graphs. A more restricted version suitable for undirected graphs has been studied in [9].

This paper is organized as follows. Section 2 gives a quick overview of the algebraic approach to path problems. Section 3 presents a simple algebraic path-finding method. The next Section 4 explains how this basic method can be transformed to run on a ring of processors. Section 5 demonstrates that the obtained distributed algorithm is indeed correct. Section 6 describes an actual network implementation of the algorithm based on the *PVM package* [5]. Section 7 lists the results of experiments, where the speedup of the implemented algorithm has been measured on randomly generated path problems. The final Section 8 gives conclusions.

2 Path problems

The algebraic approach to path problems tries to find a general (abstract) formulation for the whole family of concrete problems. According to [1], each particular path problem, posed on a graph with n nodes, is represented by a suitable $n \times n$ *adjacency* matrix, let us denote it by A. The entries a_{ij} of A are not necessarily numbers, but elements of a special set P, which is called a *path algebra*. The elements of P are manipulated by two binary operations, $\vee$ (join) and $\circ$ (multiplication), which are analogous to conventional addition and multiplication, respectively. The two operations satisfy the following properties: $\vee$ is idempotent, commutative, and associative; $\circ$ is associative, left-distributive, and right-distributive over $\vee$; there exist a zero element ϕ and a unit element ϵ such that (for any a) $\phi \vee a = a$, $\phi \circ a = \phi = a \circ \phi$, $\epsilon \circ a = a = a \circ \epsilon$. Also, the operation $\vee$ generates an ordering of P, let us denote it by $\preceq$, such that $a \preceq b$ if and only if $a \vee b = b$. Both operations $\vee$ and $\circ$ are

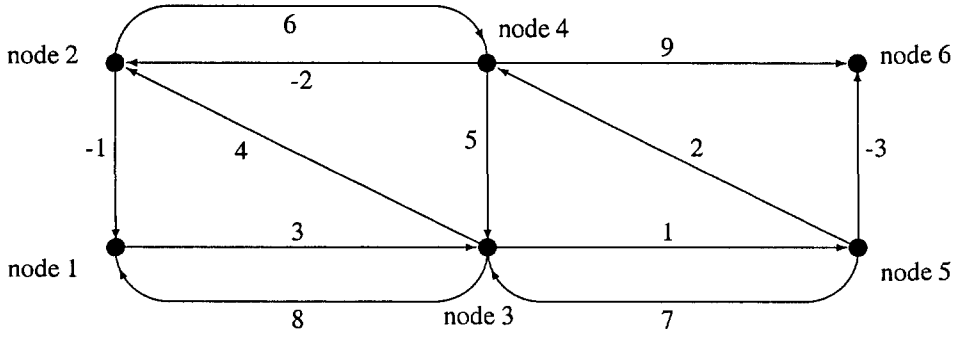


Figure 1: A shortest distance problem

isotone for $\preceq$, and ϕ is the least element.

The solution of the path problem given by A is obtained by computing the so-called *closure* matrix:

$$A^* = E \vee A \vee A^2 \vee A^3 \vee \ldots . \qquad (1)$$

Here, E denotes the unit matrix (ϵ on the diagonal, ϕ elsewhere). The matrix operations $\vee$ and $\circ$ are derived from the corresponding scalar operations, similarly as in ordinary linear algebra. Also, the generated matrix ordering $\preceq$ is consistent with entry-by-entry scalar comparison. In all meaningful path problems, the matrix A is *stable*, i.e. the join (1) becomes saturated after adding enough powers of A. It means that the closure A^* can be computed in a finite number of algebraic operations.

To illustrate these ideas, let us consider the graph in Figure 1 whose arcs are given "lengths". Suppose that we want to solve the *shortest distance* problem, i.e. we want to determine the length of a shortest path between any pair of nodes. Then the matrix A and its closure A^* are:

$$A = \begin{bmatrix} \infty & \infty & 3 & \infty & \infty & \infty \\ -1 & \infty & \infty & 6 & \infty & \infty \\ 8 & 4 & \infty & \infty & 1 & \infty \\ \infty & -2 & 5 & \infty & \infty & 9 \\ \infty & \infty & 7 & 2 & \infty & -3 \\ \infty & \infty & \infty & \infty & \infty & \infty \end{bmatrix},$$

$$A^* = \begin{bmatrix} 0 & 4 & 3 & 6 & 4 & 1 \\ -1 & 0 & 2 & 5 & 3 & 0 \\ 0 & 1 & 0 & 3 & 1 & -2 \\ -3 & -2 & 0 & 0 & 1 & -2 \\ -1 & 0 & 2 & 2 & 0 & -3 \\ \infty & \infty & \infty & \infty & \infty & 0 \end{bmatrix}.$$

A is indeed stable since the corresponding graph does not contain a cycle of negative length. P is here the set of real numbers including ∞, with the standard min as $\vee$, and the standard $+$ as $\circ$. The zero in P is ∞, and the unit element is 0. The (i,j)-th entry of A is the length of the arc from node i to node j, while the (i,j)-th entry of A^* is the length of the shortest path from node i to node j.

As the second example of a path problem, let us consider again the same graph in Figure 1, but suppose now that we want to solve the *path existence* problem, i.e. we only want to determine which pairs of nodes are connected by a path and which are not. Then we can use a much simpler Boolean path algebra P, which consists of the values 0 and 1, with the operations max and min as $\vee$ and $\circ$, respectively. The corresponding matrices A and A^* have the same zero/nonzero structure as before: any ∞ is now replaced with 0 and any other value with 1. The (i,j)-th entry of A specifies the existence of an arc from node i to node j, while the (i,j)-th entry of A^* indicates the existence of a path from node i to node j.

Many other types of path problems can be solved in a similar fashion. The general structure of the problem always stays the same. However, the set P changes, and the operations $\vee$ and $\circ$ can have different meanings. More examples can be found in [1, 11].

3 Basic algorithm

Our basic algorithm for solving path problems is in fact an algorithm that computes the closure A^* of a given stable $n \times n$ matrix A over an arbitrary path algebra P. The expression of the form (1) is evaluated relatively quickly, by squaring a suitable matrix several times. Since the path algebra P is not specified, the algorithm can be applied to many different types of path problems.

More precisely, the following sequence of matrices is computed:

$$\begin{aligned} B^{(0)} &= E \vee A, \\ B^{(k)} &= B^{(k-1)} \circ B^{(k-1)} \quad (k = 1, 2, \ldots). \end{aligned}$$

Computation stops when two consecutive matrices $B^{(k)}$ and $B^{(k-1)}$ are equal. It is easy to check that

$$B^{(k)} = E \vee A \vee A^2 \vee A^3 \vee \ldots \vee A^{2^k}.$$

Thus for a stable matrix A the algorithm terminates, and the final $B^{(k)}$ is equal to A^*.

Note that the (i,j)-th entry $b_{ij}^{(k)}$ of the matrix $B^{(k)}$ is evaluated as the "inner product" of the i-th row and j-th column of the matrix $B^{(k-1)}$:

$$b_{ij}^{(k)} = \bigvee_{l=1}^{n} b_{il}^{(k-1)} \circ b_{lj}^{(k-1)}. \qquad (2)$$

In each iteration, the formula (2) must be evaluated for all $i = 1, 2, \ldots, n$ and all $j = 1, 2, \ldots, n$.

4 Distributed algorithm

Our distributed algorithm for solving path problems can be considered as a modified version of the basic algorithm described in the previous section. This version again computes the closure A^* of a given stable $n \times n$ matrix A over an arbitrary path algebra P. However, the algorithm now employs a ring of m processors, where $1 \leq m \leq n/2$. Each processor is connected by a bidirectional communication link with its predecessor and successor. There is no shared memory, but each processor has its own private memory. Figure 2 refers to a ring of 4 processors.

The distributed algorithm uses two $n \times n$ matrices, denoted by $R = [r_{ij}]$ and $C = [c_{ij}]$. In the beginning, both R and C are equal to $E \vee A$. Through the algorithm, R and C are being simultaneously updated, so that they always remain equal. Finally, both R and C become equal to A^*.

The matrix R is divided into $2m$ blocks, so that every block consists of a roughly equal number of adjacent rows. Similarly, the matrix C is divided into $2m$ blocks of columns. The range of column indices assigned to one particular block of C is the same as the range of row indices assigned to the *corresponding* block of R. The blocks of R and C are distributed among processors, so that one processor keeps exactly two blocks of R and the corresponding two blocks of C.

The algorithm consists of a sufficient number of *sweeps* (iterations). One sweep is carried out in $2m - 1$ *phases*. The algorithm can be stopped as soon the matrices R and C remain unchanged through a whole sweep.

In one phase a processor generates row-column pairs from the available blocks. In each phase, all possible pairs from non-corresponding blocks are generated. In the first phase within a sweep, additionally, all possible pairs from corresponding blocks are also generated. For each generated pair, consisting of, say, the i-th row of R and the j-th column of C, the processor computes the "inner product". The obtained value is used to update both the j-th element of the i-th row of R and the i-th element of the j-th column of C; thus:

$$r_{ij} := c_{ij} := \bigvee_{l=1}^{n} r_{il} \circ c_{lj}. \qquad (3)$$

After any phase the processors exchange blocks, in order to form new combinations of blocks for the next phase. Any block of C moves together with the corresponding block of R. The exchange procedure has to assure that through a single sweep ($2m - 1$ consecutive phases) each block of R meets each of the non-corresponding blocks of C exactly once.

There are many block exchange procedures available. Our chosen procedure requires that after each phase Rules I and II should alternately be applied. Both rules are shown in Figure 2 for the case where $m = 4$, and they are defined analogously for any $m \geq 2$. Each index in Figure 2 denotes one block of R together with the corresponding block of C. Arrows in Figure 2 indicate block moves. Rules I and II for block exchange have originally been developed in the context of matrix eigenvalue problems [3]. It has been proved that the rules are indeed correct, i.e. any unordered pair of indices is formed exactly once in some processor during one sweep. The initial distribution of blocks among processors is established again after two sweeps. Table 1 lists the unordered pairs of indices, formed during the first sweep, for the case where $m = 4$.

Assume now that $n = 16$, $m = 4$. Then the matrix R (or C) is divided into 8 blocks, each block consisting of 2 rows (columns). Block 1 consists of rows (columns) 1 and 2, block 2 consists of rows (columns) 3 and 4, etc. Table 2 lists the row-column pairs (i, j) which are generated in the first two phases of the first sweep. Each pair determines an update operation (3).

5 Proof of correctness

In order to reduce computational and communication costs, our distributed algorithm uses in fact only one $n \times n$ matrix. This matrix is maintained in two copies, R and C, so that one copy is distributed among rows and the other among columns. Entries in the matrices R and C are updated and again used during the same sweep. For that reason, the updating formula (3) is not quite equivalent to the formula (2) from the basic algorithm. Consequently, the values of R and C obtained after each sweep do not follow exactly the sequence $B^{(k)}$ $(k = 1, 2, \ldots)$ from the basic algorithm. Thus we have to prove that the distributed algorithm is still correct.

Lemma 1 *All diagonal entries in the matrices $B^{(k)}$ $(k = 0, 1, 2, \ldots)$ generated by the basic algorithm are $\succeq \epsilon$.*

Proof. Carried out by mathematical induction on k. First, for any i,

$$b_{ii}^{(0)} = \epsilon \vee a_{ii} \succeq \epsilon.$$

Assume that the claim is valid for some $k - 1$. Then, for any i,

$$b_{ii}^{(k)} = \bigvee_{l=1}^{n} b_{il}^{(k-1)} \circ b_{li}^{(k-1)} \succeq b_{ii}^{(k-1)} \circ b_{ii}^{(k-1)} \succeq \epsilon \circ \epsilon = \epsilon.$$

□

Lemma 2 *The matrices $B^{(k)}$ generated by the basic algorithm satisfy the inequalities*

$$B^{(k)} \succeq B^{(k-1)} \quad (k = 1, 2, \ldots).$$

Proof. We check the claim by direct comparison of matrix entries. After choosing any feasible i, j and k, we may write

$$b_{ij}^{(k)} = \bigvee_{l=1}^{n} b_{il}^{(k-1)} \circ b_{lj}^{(k-1)} \succeq b_{ij}^{(k-1)} \circ b_{jj}^{(k-1)} \succeq$$

$$\ldots \text{Lemma 1} \ldots \succeq b_{ij}^{(k-1)} \circ \epsilon = b_{ij}^{(k-1)}.$$

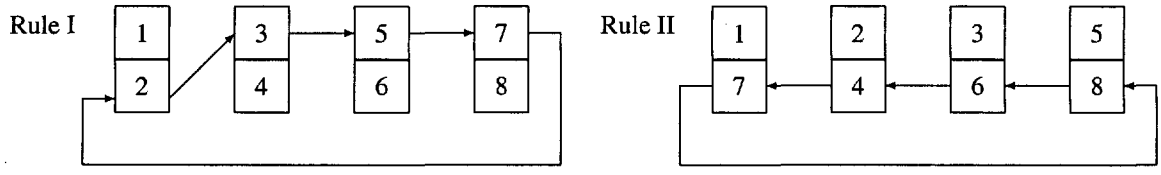


Figure 2: Block exchange rules for $m = 4$.

phase	processor 1	processor 2	processor 3	processor 4
1	{1,2}	{3,4}	{5,6}	{7,8}
2	{1,7}	{2,4}	{3,6}	{5,8}
3	{1,4}	{2,6}	{3,8}	{5,7}
4	{1,5}	{4,6}	{2,8}	{3,7}
5	{1,6}	{4,8}	{2,7}	{3,5}
6	{1,3}	{6,8}	{4,7}	{2,5}
7	{1,8}	{6,7}	{4,5}	{2,3}

Table 1: Block index pairs during one sweep for $m = 4$.

phase	processor 1	processor 2	processor 3	processor 4
1	(1,1) (1,2)	(5,5) (5,6)	(9,9) (9,10)	(13,13) (13,14)
	(2,1) (2,2)	(6,5) (6,6)	(10,9) (10,10)	(14,13) (14,14)
	(3,3) (3,4)	(7,7) (7,8)	(11,11) (11,12)	(15,15) (15,16)
	(4,3) (4,4)	(8,7) (8,8)	(12,11) (12,12)	(16,15) (16,16)
	(1,3) (3,1)	(5,7) (7,5)	(9,11) (11,9)	(13,15) (15,13)
	(1,4) (4,1)	(5,8) (8,5)	(9,12) (12,9)	(13,16) (16,13)
	(2,3) (3,2)	(6,7) (7,6)	(10,11) (11,10)	(14,15) (15,14)
	(2,4) (4,2)	(6,8) (8,6)	(10,12) (12,10)	(14,16) (16,14)
2	(1,13) (13,1)	(3,7) (7,3)	(5,11) (11,5)	(9,15) (15,9)
	(1,14) (14,1)	(3,8) (8,3)	(5,12) (12,5)	(9,16) (16,9)
	(2,13) (13,2)	(4,7) (7,4)	(6,11) (11,6)	(10,15) (15,10)
	(2,14) (14,2)	(4,8) (8,4)	(6,12) (12,6)	(10,16) (16,10)

Table 2: Row-column pairs, first and second phase of the first sweep, $n = 16$, $m = 4$.

□

Lemma 3 *Consider a stable matrix A. Let $R^{(k)}$ and $C^{(k)}$ be the values of the matrices R and C obtained by the distributed algorithm immediately after the k-th sweep. Then these matrices and the matrices $B^{(k)}$ from the basic algorithm satisfy the inequalities*

$$B^{(k)} \preceq R^{(k)} = C^{(k)} \preceq A^* \quad (k = 0, 1, 2, \ldots).$$

Proof. Carried out by induction on time instants when the entries of R and C are updated. Assume the notation: $R^{(k)} = [r_{ij}^{(k)}]$, $C^{(k)} = [c_{ij}^{(k)}]$. The induction *basis*: we check that the claim is true at the time instant 0 (at the beginning). Indeed, for any i and j,

$$b_{ij}^{(0)} = r_{ij}^{(0)} = c_{ij}^{(0)} = (E \vee A)_{ij} \preceq (E \vee A \vee A^2 \vee \ldots)_{ij} = (A^*)_{ij}.$$

The induction *hypothesis*: we fix an instant t and choose any instant $\bar{t} < t$. Let us denote with $\bar{k}$ the sweep to which $\bar{t}$ belongs. Then we assume that for any $r_{ij}^{(\bar{k})}$ and $c_{ij}^{(\bar{k})}$ computed at the instant $\bar{t}$ it holds

$$b_{ij}^{(\bar{k})} \preceq r_{ij}^{(\bar{k})} = c_{ij}^{(\bar{k})} \preceq (A^*)_{ij}.$$

The induction *step*: we consider now the previously fixed instant t. Let us denote with k the sweep to which t belongs. Let us choose an entry $r_{ij}^{(k)} = c_{ij}^{(k)}$ being computed at the instant t. We want to show that

$$b_{ij}^{(k)} \preceq r_{ij}^{(k)} = c_{ij}^{(k)} \preceq (A^*)_{ij}.$$

The value $r_{ij}^{(k)} = c_{ij}^{(k)}$ is obtained according to the formula (3), as the inner product of the i-th row of R and the j-th column of C. Some elements of that row or column have already been updated in the k-th sweep, and some have not. Let S_1 be the set of indices l such that r_{il} has been updated, and let S_2 be the set of indices l such that c_{lj} has been updated. Then

$$r_{ij}^{(k)} = c_{ij}^{(k)} = \bigvee_{l \in S_1 \cap S_2} r_{il}^{(k)} \circ c_{lj}^{(k)} \vee \bigvee_{l \in S_1 \setminus S_2} r_{il}^{(k)} \circ c_{lj}^{(k-1)} \vee \bigvee_{l \in S_2 \setminus S_1} r_{il}^{(k-1)} \circ c_{lj}^{(k)} \vee \bigvee_{l \notin S_1 \cup S_2} r_{il}^{(k-1)} \circ c_{lj}^{(k-1)}. \quad (4)$$

All elements on the right-hand side of (4) have been created before t, thus they are covered by the induction hypothesis. By using lower bounds implied by the induction hypothesis, we get

$$r_{ij}^{(k)} = c_{ij}^{(k)} \succeq \bigvee_{l \in S_1 \cap S_2} b_{il}^{(k)} \circ b_{lj}^{(k)} \vee \bigvee_{l \in S_1 \setminus S_2} b_{il}^{(k)} \circ b_{lj}^{(k-1)} \vee \bigvee_{l \in S_2 \setminus S_1} b_{il}^{(k-1)} \circ b_{lj}^{(k)} \vee \bigvee_{l \notin S_1 \cup S_2} b_{il}^{(k-1)} \circ b_{lj}^{(k-1)}. \quad (5)$$

Next, by applying Lemma 2 to the elements $b_{il}^{(k)}$ and $b_{lj}^{(k)}$ in (5), we obtain

$$r_{ij}^{(k)} = c_{ij}^{(k)} \succeq \bigvee_{l=1}^{n} b_{il}^{(k-1)} \circ b_{lj}^{(k-1)} = b_{ij}^{(k)}. \quad (6)$$

Finally, by combining upper bounds from the induction hypothesis and (4), we get

$$r_{ij}^{(k)} = c_{ij}^{(k)} \preceq \bigvee_{l=1}^{n} (A^*)_{il} \circ (A^*)_{lj} = (A^* \circ A^*)_{ij} = (A^*)_{ij}. \quad (7)$$

The inequalities (6) and (7) together complete the induction step. □

Theorem 1 *Suppose that the given matrix A is stable. Then the distributed algorithm terminates after a finite number of sweeps. Also, the resulting matrices R and C are both equal to A^*.*

Proof. Follows from Lemma 3, by taking into account that $B^{(k)} = A^*$ if k is big enough. □

Note that Lemma 3 also implies that the number of sweeps needed by the distributed algorithm is never greater than the number of iterations needed by the basic algorithm. According to [1] this means that for most path problems the distributed algorithm finishes in $\leq \log_2 n$ sweeps. As we will see in Section 7, our experimental results suggest that the algorithm needs in fact only few sweeps, even for relatively large problems.

6 Network implementation

PVM [5] is a well known software package that transforms a network of UNIX computers into a virtual parallel machine. A distributed program is realized within the PVM framework as a set of sequential tasks (processes); these tasks can be allocated to different hosts (computers), and they can exchange messages (data) through the network.

By using the C language and the PVM library, we have developed an actual network implementation of the distributed algorithm for solving path problems. The obtained program code can easily be adjusted to solve various types of path problems (path existence, shortest distances, ... , etc). Namely, the algorithm is coded in terms of an abstract path algebra whose definition is given in a separate module (abstract data type). By changing that module and by recompilation, one can easily switch from one type of problem to another.

The main purpose of our network implementation is to enable experimental evaluation of the distributed algorithm. Therefore the obtained program not only solves a given path problem, but it also measures the performance of the algorithm. As input data, the program takes: the size n of the matrix A, the size m of the imaginary ring of processors, and the matrix A itself. As output, the program

prints the closure matrix A^* together with the following performance data: the number of sweeps, the total execution time.

The running program consists of $m + 1$ PVM tasks. The first task is called the *master*, and the remaining tasks are m identical *slaves*. Strictly speaking, the master is not the part of the algorithm, and it serves only as an interface to the user. The actual algorithm is done jointly by the slaves. More precisely, one slave performs the part of the algorithm assigned to one processor within the ring. The assumed ring architecture is emulated by appropriate ring-like communication among the slaves. Thanks to this communication, the slaves are to some extent synchronized. For instance, a slave can finish a sweep only if all other slaves have also reached the end of that sweep.

To start the program, one must launch the master task. The master first reads the input file, spawns m slaves on appropriate hosts, and broadcasts auxiliary configuration data to all slaves. The configuration data comprise n, m, and the list of task ids. Next, the master generates the matrices R and C, both equal to $E \vee A$, and divides each of them into $2m$ appropriately sized blocks of rows or columns. The blocks are distributed among the slaves. Each block is represented by a structure that includes not only the rows/columns but also the original row/column indices. After distribution of blocks, the master waits until the slaves accomplish the algorithm. At the end, the master collects the final versions of column blocks and the performance data from the slaves, assembles the obtained closure matrix $C = A^*$, summarizes the performance data, and writes the output file.

As soon as they are spawned, all slaves receive the configuration data from the master. Next, each slave receives its two blocks of R and two blocks of C. After receiving their data, the slaves collectively perform the distributed algorithm and measure the performance. More precisely, each slave executes a nested loop, whose outer part corresponds to sweeps and inner part to phases. A phase includes computation and block transfers as described in Section 4. At the end of a sweep, all slaves together check whether R and C have changed during that sweep: this involves additional communication along the ring, namely the slaves inform each other about changes. If there has been no change, then each slave sends its final blocks of $C = A^*$ and its performance measurements to the master, thus finishing execution.

The performance is monitored so that each slave counts sweeps and measures its own execution time. Only the essential part of the algorithm is considered, i.e. the part between receiving data and sending results. The real physical time is recorded, which comprises computation, communication, and possible additional delays caused by other users of the network. As we have already noted, all slaves are synchronized; therefore each slave should report exactly the same number of sweeps and approximately the same execution time. Small differences in time measurements are possible, and they can be explained by the fact that some slaves receive their data earlier and seemingly work longer. The master finally computes the minimum of all measurements as the most accurate estimate of the execution time.

The program has been tested on a virtual parallel machine assembled of four UNIX computers: a HP 9000/845SE server, a HP 712/80 workstation, a SUN Ultra Enterprise 3000 server with two CPU-s, and a HP 9000/E55 server. Taking into account the technical characteristics of the four computers and their usual loads, we have decided to run the program with at most 8 slaves, so that the first slave should be assigned to HP 9000/845SE, the second slave and the master to HP 712/80, the next four slaves to SUN Ultra Enterprise 3000, and the last two slaves to HP 9000/E55. Note that our four hosts operate at very different speeds. A faster computer can execute more than one slave concurrently, without slowing down the ring of slaves as a whole. Namely, all slave tasks are synchronized, and they adjust their speeds according to the slowest one. In our chosen assignment the slowest slave is always the first one, and it runs on the first host that happens to be very evenly loaded. Consequently, the actual speed of the slowest slave is always roughly the same, and therefore the testing results obtained with the chosen assignment should be fairly consistent.

7 Experimental results

Our results presented in Tables 3 - 6 deal with the performance of the considered distributed algorithm, and they are obtained by running the PVM program on randomly generated path problems. The results are based on the values directly measured by the program (number of sweeps, execution time). More than 600 experiments have been performed.

In our experiments we have used path existence and also shortest distance problems. For both types, we have chosen problems with smaller graphs (size n of the adjacency matrix A equal to 128) and problems with larger graphs (n equal to 256). Each of Tables 3 - 6 summarizes the results for 40 problems of the same type and same size. The problems within the same table have further been divided into groups of 10 according to their graph density (percentage of non-zeros in A), which can be 1%, 3%, 10% or 30%. Each particular problem has been solved four times, i.e. with the number of processors m equal to 1, 2, 4 or 8.

Entries in Tables 3 - 6 are in fact mean values, computed over a whole group of 10 similar problems. A table row shows not only the number of sweeps and the total execution time, but also the average time per sweep. Both times are expressed as absolute values (seconds), and as well as relative values (speedups compared to the sequential case). Thus each table should describe the typical behavior of the algorithm on a problem of a chosen type and size. This behavior is expressed in dependence on the graph density and on the number of available processors.

graph density	# of processors m	# of sweeps	total execution time in seconds (speedup)	avg time per sweep in seconds (speedup)
1%	1	4.0	10.9 (1.00)	2.72 (1.00)
	2	4.0	7.2 (1.51)	1.80 (1.51)
	4	4.0	5.7 (1.91)	1.42 (1.91)
	8	4.0	6.4 (1.70)	1.60 (1.70)
3%	1	3.0	8.2 (1.00)	2.73 (1.00)
	2	3.0	5.4 (1.52)	1.80 (1.52)
	4	3.0	4.3 (1.91)	1.43 (1.91)
	8	3.0	4.8 (1.71)	1.60 (1.71)
10%	1	3.0	8.2 (1.00)	2.73 (1.00)
	2	3.0	5.4 (1.52)	1.80 (1.52)
	4	3.0	4.3 (1.91)	1.43 (1.91)
	8	3.0	4.8 (1.71)	1.60 (1.71)
30%	1	2.0	5.5 (1.00)	2.75 (1.00)
	2	2.0	3.6 (1.53)	1.80 (1.53)
	4	2.0	2.7 (2.04)	1.35 (2.04)
	8	2.0	3.0 (1.83)	1.50 (1.83)

Table 3: Experimental results: path existence problems, size $n = 128$.

graph density	# of processors m	# of sweeps	total execution time in seconds (speedup)	avg time per sweep in seconds (speedup)
1%	1	3.7	88.3 (1.00)	23.85 (1.00)
	2	3.5	46.5 (1.90)	13.28 (1.79)
	4	3.3	26.7 (3.31)	8.09 (2.95)
	8	3.5	20.4 (4.33)	5.83 (4.09)
3%	1	3.0	71.5 (1.00)	23.83 (1.00)
	2	3.0	40.0 (1.79)	13.33 (1.79)
	4	3.0	24.4 (2.93)	8.13 (2.93)
	8	3.0	17.5 (4.09)	5.83 (4.09)
10%	1	2.9	69.0 (1.00)	23.80 (1.00)
	2	3.0	39.7 (1.74)	13.23 (1.80)
	4	3.0	24.3 (2.84)	8.10 (2.94)
	8	3.0	17.4 (3.97)	5.80 (4.10)
30%	1	2.0	47.4 (1.00)	23.70 (1.00)
	2	2.0	26.2 (1.81)	13.10 (1.81)
	4	2.0	16.0 (2.96)	8.00 (2.96)
	8	2.0	11.4 (4.16)	5.70 (4.16)

Table 4: Experimental results: path existence problems, size $n = 256$.

graph density	# of processors m	# of sweeps	total execution time in seconds (speedup)	avg time per sweep in seconds (speedup)
1%	1	4.0	31.3 (1.00)	7.82 (1.00)
	2	4.0	18.1 (1.73)	4.52 (1.73)
	4	4.0	10.4 (3.01)	2.60 (3.01)
	8	4.0	7.5 (4.17)	1.87 (4.17)
3%	1	4.0	38.1 (1.00)	9.52 (1.00)
	2	4.0	20.8 (1.83)	5.20 (1.83)
	4	4.0	11.9 (3.20)	2.97 (3.20)
	8	4.0	8.3 (4.59)	2.07 (4.59)
10%	1	4.0	39.9 (1.00)	9.97 (1.00)
	2	4.0	21.6 (1.85)	5.40 (1.85)
	4	4.0	12.2 (3.27)	3.05 (3.27)
	8	4.0	8.6 (4.64)	2.15 (4.64)
30%	1	4.0	42.7 (1.00)	10.67 (1.00)
	2	4.0	23.0 (1.86)	5.75 (1.86)
	4	4.0	12.7 (3.36)	3.17 (3.36)
	8	4.0	8.8 (4.85)	2.20 (4.85)

Table 5: Experimental results: shortest distance problems, size $n = 128$.

graph density	# of processors m	# of sweeps	total execution time in seconds (speedup)	avg time per sweep in seconds (speedup)
1%	1	4.0	292.7 (1.00)	73.17 (1.00)
	2	4.0	153.3 (1.91)	38.32 (1.91)
	4	4.0	82.1 (3.56)	20.52 (3.56)
	8	4.0	46.3 (6.32)	11.57 (6.32)
3%	1	4.0	316.7 (1.00)	79.17 (1.00)
	2	4.0	164.7 (1.92)	41.17 (1.92)
	4	4.0	88.3 (3.59)	22.07 (3.59)
	8	4.0	49.9 (6.35)	12.47 (6.35)
10%	1	4.0	324.8 (1.00)	81.20 (1.00)
	2	4.0	167.2 (1.94)	41.80 (1.94)
	4	4.0	89.9 (3.61)	22.47 (3.61)
	8	4.0	51.0 (6.37)	12.75 (6.37)
30%	1	4.0	374.3 (1.00)	93.57 (1.00)
	2	4.0	191.6 (1.95)	47.90 (1.95)
	4	4.0	101.9 (3.67)	25.47 (3.67)
	8	4.0	57.7 (6.49)	14.42 (6.49)

Table 6: Experimental results: shortest distance problems, size $n = 256$.

As we see from Tables 3 and 4, our algorithm is not very efficient when applied to path existence problems. Or more precisely, the algorithm is not able to utilize more than few processors. For smaller problems, the execution time with 8 processors is even worse than with 4 processors. The performance for larger problems is somewhat better.

If we restrict to one particular group of path existence problems (fixed size and density), than we can observe that the number of sweeps is only slightly influenced by the number of processors. Therefore the speedup computed by using the total execution time is similar to the speedup obtained by using the average time per sweep.

If we compare different groups of equally large path existence problems (same size, different densities), we observe that the average time per sweep does not depend on the graph density. On the other hand, the number of sweeps becomes smaller when the graph becomes denser. This phenomenon is easy to explain: very dense graphs are toowell connected, and the corresponding path existence problems become solvable in only 2 iterations. Consequently, the results for very dense graphs would look exactly the same as for 30% density. That's the reason why densities greater than 30% have not been included in Tables 3 and 4.

Tables 5 and 6 indicate that our algorithm is much more efficient when applied to shortest distance problems. Satisfactory speedups are already obtained for smaller problems, and even better results are achieved for larger problems. The number of sweeps is apparently not influenced by the number of processors. Also, the number of sweeps does not depend on the graph density.

If we compare different groups of equally large shortest distance problems, we see that the average time per sweep becomes larger when the graph density increases. This fact can be explained in the following way. A dense graph allows construction of many different paths connecting the same pair of nodes. To determine the shortest distance, it is not enough to find only one path between two given nodes as in the case of path existence. Instead, all possible paths should be examined and their lengths should be compared. Consequently, shortest distance problems with denser graphs become computationally very complex.

It would certainly be interesting to consider shortest distance problems with graph densities greater than those shown in Tables 5 and 6. The corresponding speedups would probably be even better than for 30% density. However, we have not included such experiments due to some practical limitations. Namely, shortest distance problems with very dense graphs require unproportionally large computing times, and that causes unpredictable delays in a multi-user environment, thus producing unreliable measurements.

8 Conclusions

Particular ideas presented in this paper are not new. However, those ideas have been combined in a new way. Namely, the paper has shown that certain matrix distribution techniques, originally developed for matrix eigenvalue problems, can successfully be applied to path problems in graphs.

Our theoretical analysis has proved that the considered distributed algorithm is correct. Our PVM implementation has demonstrated that the algorithm can run on an ordinary network of computers, and solve relatively large path problems. According to our experimental results, the actual efficiency with a given number of processors varies considerably, an it depends on the problem type, graph size, and graph density. The best speedups are achieved for computationally intensive tasks, such as finding shortest distances in large dense graphs.

In this paper we have tried to construct a relatively general algorithm that can solve a wide class of path problems in both directed and undirected graphs. Of course, this general method can be optimized for certain smaller classes of problems. For instance, we have already proposed a simplified version of the algorithm, which works correctly with most undirected graphs, and which runs approximately two times faster than the general version. We believe that similar improvements are also possible for some other classes of problems. Our future research will concentrate on further optimization of the algorithm.

References

[1] Carré B. (1979) *Graphs and Networks*, Oxford University Press, Oxford.

[2] Crichlow J.M. (1997) *An Introduction to Distributed and Parallel Programming*, Prentice-Hall, Englewood Cliffs, NJ.

[3] Eberlein P.J. and Park H. (1990) Efficient implementation of Jacobi algorithms and Jacobi sets on distributed memory architectures, *Journal of Parallel and Distributed Computing*, Vol 8, pp. 358-366.

[4] Gayraud T. and Authie G. (1992) A parallel algorithm for the all pairs shortest path problem, *Parallel Computing '91*, ed. by D.J. Evans, G.R. Joubert and H. Liddell, Advances in Parallel Computing 4, North-Holland, Amsterdam, pp. 107-114.

[5] Geist A., Beguelin A., Dongarra J., Jiang W., Manchek R. and Sunderam V. (1994) *PVM: Parallel Virtual Machine - A Users' Guide and Tutorial for Networked Parallel Computing*, The MIT Press, Cambridge, MA.

[6] Horowitz E., Sahni S. and Rajasekaran S. (1997) *Algorithms / C++*, Computer Science Press, New York.

[7] Humblet P.A. (1991) Another adaptive distributed shortest path algorithm, *IEEE Transactions on Communications*, Vol 39, pp. 995-1000.

[8] Manger R. (1996) A comparison of two parallel iterative algorithms for solving path problems, *Journal of Computing and Information Technology - CIT*, Vol 4, pp. 75-85.

[9] Manger R. and Nogo G. (1999) Experiments with a distributed algorithm for solving path problems, *Proceedings of the 7th International Conference on Operational Research (KOI'98 - Rovinj, Sep 30 - Oct 2, 1998)*, ed. by I. Aganović, T. Hunjak, and R. Scitovski, Croatian Operational Research Society, Osijek, pp. 177-186.

[10] Orda A. and Rom R. (1996) Distributed shortest-path protocols for time-dependent networks, *Distributed Computing*, Vol 10, pp. 49-62.

[11] Rote G. (1990) Path problems in graphs, *Computing Supplement*, Vol 7, pp. 155-189.

[12] Zhu S. and Huang G.M. (1998) A new parallel and distributed shortest path algorithm for hierarchically clustered data networks, *IEEE Transactions on Parallel and Distributed Systems*, Vol 9, pp. 841-855.

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 700 staff, has 500 researchers, about 250 of whom are postgraduates, over 200 of whom have doctorates (Ph.D.), and around 150 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 **Slove**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.

In the last year on the site of 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.

At the present time, part of the Institute is being 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 is being developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park will take the form of a shareholding company and will host an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of 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 Economic Relations and Development, 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 1 219 385
Tlx.:31 296 JOSTIN SI
WWW: http://www.ijs.si
E-mail: matjaz.gams@ijs.si
Contact person for the Park: Iztok Lesjak, M.Sc.
Public relations: Natalija Polenec

EDITORIAL BOARDS, PUBLISHING COUNCIL

Informatica is a journal primarily covering the European computer science and informatics 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 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.

Volume 26 Number 1 May 2002 ISSN 0350-5596

Informatica

An International Journal of Computing and Informatics