IARIA

ADVCOMP 2010

The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences

October 25-30, 2010 - Florence, Italy

Editors

Wolfgang Gentzsch

Pascal Lorenz

Oana Dini

ADVCOMP 2010

Foreword

The Fourth International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP 2010) held from October 25 to October 30, 2010 in Florence, Italy, was a multi-track event covering a large spectrum of topics related to advanced engineering computing and applications in sciences.

With the advent of high performance computing environments, virtualization, distributed and parallel computing, as well as the increasing memory, storage and computational power, processing particularly complex scientific applications and voluminous data is more affordable. With the current computing software, hardware and distributed platforms effective use of advanced computing techniques is more achievable.

The goal of ADVCOMP 2010 was to bring together researchers from the academia and practitioners from the industry in order to address fundamentals of advanced scientific computing and specific mechanisms and algorithms for particular sciences. The conference provided a forum where researchers were able to present recent research results and new research problems and directions related to them. The conference sought contributions presenting novel research in all aspects of new scientific methods for computing and hybrid methods for computing optimization, as well as advanced algorithms and computational procedures, software and hardware solutions dealing with specific domains of science.

We take here the opportunity to warmly thank all the members of the ADVCOMP 2010 technical program committee as well as the numerous reviewers. The creation of such a broad and high quality conference program would not have been possible without their involvement. We also kindly thank all the authors that dedicated much of their time and efforts to contribute to ADVCOMP 2010. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

This event could also not have been a reality without the support of many individuals, organizations and sponsors. We also gratefully thank the members of the ADVCOMP 2010 organizing committee for their help in handling the logistics and for their work that is making this professional meeting a success. We gratefully appreciate to the technical program committee co-chairs that contributed to identify the appropriate groups to submit contributions.

We hope Florence provided a pleasant environment during the conference and everyone saved some time for exploring this historic city.

ADVCOMP 2010 Chairs:

Petre Dini, IARIA / Concordia University, Canada
Simon Fabri, University of Malta, Malta

Wolfgang Gentzsch, EU DEISA Project & Open Grid Forum, Germany
Chih-Cheng Hung, Southern Polytechnic State University, USA
Flavio Oquendo, European University of Brittany - UBS/VALORIA, France
Juha Röning, Oulu University, Finland
Jameleddine Hassine, Cisco Systems, Inc., Canada
Kurt Rohloff, BBN Technologies, USA
Jorge Ejarque Artigas, Barcelona Supercomputing Center (BSC-CNS), Spain
Markus Kunde, German Aerospace Center - Koeln, Germany
Helmut Reiser, Leibniz Supercomputing Centre (LRZ) - Garching, Germany
Xinyu Xu, Sharp Labs of America, Inc., USA

ADVCOMP 2010

Committee

ADVCOMP Advisory Chairs
Petre Dini, IARIA / Concordia University, Canada
Simon Fabri, University of Malta, Malta
Wolfgang Gentzsch, EU DEISA Project & Open Grid Forum, Germany
Chih-Cheng Hung, Southern Polytechnic State University, USA
Flavio Oquendo, European University of Brittany - UBS/VALORIA, France
Juha Röning, Oulu University, Finland

ADVCOMP 2010 Industry Liaison Chairs
Jameleddine Hassine, Cisco Systems, Inc., Canada
Kurt Rohloff, BBN Technologies, USA

ADVCOMP 2010 Research/Industry Chairs
Jorge Ejarque Artigas, Barcelona Supercomputing Center (BSC-CNS), Spain
Markus Kunde, German Aerospace Center - Koeln, Germany
Helmut Reiser, Leibniz Supercomputing Centre (LRZ) - Garching, Germany
Xinyu Xu, Sharp Labs of America, Inc., USA

ADVCOMP 2010 Technical Program Committee
Witold Abramowicz, The Poznan University of Economics, Poland
Sónia Maria Almeida da Luz, Polytechnic Institute of Leiria, Portugal | University of Extremadura, Spain
Vincenzo Ambriola, Università di Pisa, Italy
Renato Amorim, University of London- Birkbeck, UK
Gabriel Amorós, Universitat de València, Spain
Basavaraj Anami, KLE Institute of Technology - Hubli, India
Stefan Andrei, Lamar University - Beaumont, USA
Plamen Angelov, Lancaster University, UK
Kamalrulnizam Abu Bakar, Universiti Teknologi Malaysia, Malaysia
Simona Bernardi , Università di Torino, Italy
Ateet Bhalla, Technocrats Institute of Technology - Bhopal, India
Pierre Borne, Ecole Centrale de Lille - Villeneuve d'Ascq, France
Kenneth P. Camilleri, University of Malta - Msida, Malta
Marco Campi, University of Brescia, Italy
Juan Vicente Capella Hernández, Universidad Politécnica de Valencia, Spain
Antonio Casimiro Costa, University of Lisbon, Portugal
Marisa da Silva Maximiano, Polytechnic Institute of Leiria, Portugal | University of Extremadura - Cáceres, Spain
Mirela Damian, Villanova University, USA
Michel Dayde, University of Toulouse / IRIT, France
Vieri del Bianco, University College Dublin, Ireland
Javier Diaz, Universidad de Castilla-La Mancha - Ciudad Real, Spain
Walter Didimo, University of Perugia, Italy
Petre Dini, Cisco Systems, Inc., USA / Concordia University, Canada

Jorge Ejarque Artigas, Barcelona Supercomputing Center (BSC-CNS), Spain
Simon Fabri, University of Malta, Malta
Umar Farooq, Smart Technologies ULC - Calgary, Canada
Mehdi Farshbaf-Sahih-Sorkhabi, Azad University / Fanavaran co. - Tehran, Iran
Mohammad-Reza Feizi-Derakhshi, University of Tabriz, Iran
Pablo Garcia Bringas, University of Deusto - Bilbao, Spain
Leonardo Garrido, Tec de Monterrey, Mexico
Matthieu Geist, Supélec / ArcelorMittal / INRIA, France
Wolfgang Gentzsch, EU DEISA Project & Open Grid Forum, Germany
Luis Gomes, Universidade Nova de Lisboa, Portugal
Teofilo Gonzales, University of California, Santa Barbara, USA
Bernard Grabot, ENIT, France
Daniela Grigori, University of Versailles, France
Maki K. Habib, The American University in Cairo, Egypt
Liangxiu Han, University of Edinburgh, UK
Jameleddine Hassine, Cisco Systems, Inc., Canada
Janko Heilgeist, Fraunhofer Institute for Algorithms and Scientific Computing (SCAI)- Sankt Augustin, Germany
Eckhard Hitzer, University of Fukui, Japan
Wladyslaw Homenda, Warsaw University of Technology, Poland
Eduardo Huedo Cuesta, Universidad Complutense de Madrid, Spain
Chih-Cheng Hung, Southern Polytechnic State University, USA
Eshref Januzaj, Mali Information Technologies, Kosova
Hai Jin, Huazhong University of Science and Technology - Wuhan, China
Rajkumar Kannan, Bishop Heber College (Autonomous) - Trichy, India
Dimitrios A. Karras, Chalkis Institute of Technology, Hellas
Marcel Karnstedt, DERI / National University of Ireland - Galway, Ireland
Jana Katreniakova, Comenius University Bratislava, Slovakia
Hans-Joachim Klein, University of Kiel, Germany
William Knottenbelt, Imperial College London, UK
Evangelos Kranakis, Carleton University, Canada
Danny Krizanc, Wesleyan University, USA
Markus Kunde, German Aerospace Center - Koeln, Germany
Luigi Lavazza, Università dell'Insubria - Varese, Italy
Clement Leung, Hong Kong Baptist University, Hong Kong
Juan Pablo López-Grao, University of Zaragoza, Spain
Lau Cheuk Lung, Federal University of Santa Catarina, Brazil
Anthony A. Maciejewski, Colorado State University - Fort Collins, USA
Shikharesh Majumdar, Carleton University - Ottawa, Canada
Sunilkumar S. Manvi, REVA Institute of Technology and Management - Bangalore, India
Leonardo Mariani, University of Milano Bicocca, Italy
Atif Memon, University of Maryland - College Park, USA
Seyede Leili Mirtaheri, University of Science and Technology, Iran
Henning Müller, University Hospitals of Geneva, Switzerland
Camelia Muñoz-Caro, Universidad de Castilla-La Mancha, Spain
Adrian Muscat, University of Malta, Malta
Tomoharu Nakashima, Osaka Prefecture University, Japan
Raghunath Nambiar, Hewlett-Packard, USA

Toan Nguyen, INRIA, France
Alfonso Niño, Universidad de Castilla-La Mancha, Spain
Sascha Opletal, Universität Stuttgart, Germany
Flavio Oquendo, European University of Brittany - UBS/VALORIA, France
Igor Paromtchik, INRIA - Saint Izmier, France
Witold Pedrycz, University of Alberta, Canada
Meikel Poess, Oracle, USA
Radu-Emil Precup, Politehnica University of Timisoara, Romania
Helmut Reiser, Leibniz Supercomputing Centre (LRZ)-Garching, Germany
Harald Richter, Clausthal University, Germany
Yong Man Ro, KAIST (Korea advanced Institute of Science and Technology) - Daejeon 305-701, Republic of Korea
Ivan Rodero, Rutgers State University of New Jersey / NSF Center for Autonomic Computing, USA
Kurt Rohloff , BBN Technologies, USA
Dieter Roller, Universität Stuttgart, Germany
Juha Röning, Oulu University, Finland
Necip Sahinkaya University of Bath, UK
Antonio Sala, Universidad Politecnica Valencia, Spain
José Francisco Salt Cairols, University of Valencia, Spain
Kenneth Scerri, University of Malta, Malta
Bruno Schulze, National Laboratory for Scientific Computing - LNCC - Petropolis, Brazil
Erich Schweighofer, University of Vienna, Austria
Vladimir Stantchev, Berlin Institute of Technology, Germany
Ryszard Tadeusiewicz, AGH University of Science and Technology - Krakow, Poland
Saïd Tazi, LAAS-CNRS, Université de Toulouse / Université Toulouse1, France
Daniel Thalmann, EPFL - Lausanne, Switzerland
Simon Tsang, Telcordia Technologies, Inc. - Piscataway, USA
José Valente de Oliveira, Universidade do Algarve, Portugal
Peter Vojtas, Charles University Prague, Czech Republic
Xinyu Xu, Sharp Labs of America, Inc., USA
Michael Zapf, Universität Kassel, Germany
Marek Zaremba, University of Quebec, Canada
Nadia Zerida, Paris 8 University, France

Copyright Information

Table of Contents

Network and Trust Model for Dynamic Federation

Yang Xiang, John Alan Kennedy, Matthias Egger
Rechenzentrum Garching
Max-Planck-Society
Garching, Germany
{yang.xiang, jkennedy, matthias.egger}@rzg.mpg.de

Harald Richter
Department of Informatics
Clausthal University of Technology
Clausthal-Zellerfeld, Germany
hri@tu-clausthal.de

***Abstract*—Most existing approaches to identity federation are based on static relationships. This leads to problems with scalability and deployment in real-time environment such as mobile networks. This paper introduces an underlying network and trust model for dynamic federation. We present a modified Dijkstra algorithm to calculate the trust value and apply a distributed reputation calculated based on the PageRank algorithm from Google to each entity in order to increase the attack resistance of the system.**

***Keywords*-AAI; dynamic; federation; trust; reputation;**

I. Introduction

Existing solutions for authentication and authorization infrastructure (AAI) like Shibboleth [1] are generally based on static federation. In a static federation, relationships among identity providers (IdPs) and Service providers (SPs) are manually pre-configured in their meta data. The question of whether an entity can trust another depends on if they can find each other in the meta data, thus this question can not be answered in a dynamic manner due to the static nature of the meta data.

The static structure of AAI leads to problems with scalability and interoperability for the following reasons: every new relationship between any two entities must be added manually as such a static federation can not be quickly and easily expanded to an AAI with hundreds or even thousands of IdPs and SPs. Furthermore it is difficult to connect two or more independent federations beyond their borders to form a con-federation because an entity of a certain federation does not know and hence does not trust entities from other federations. Finally, a static AAI can not be deployed in a real-time environment like a mobile network where users access the services of any provider at any time.

Hence we introduce a concept of a dynamic federation, in which the IdPs and SPs will be regarded as peers of a trusted network that evolves over time. A trust relationship between two entities is regarded as a network connection. In such a dynamic federation, an SP does not need to know an IdP beforehand. A trust relationship will be created on demand and the trust value, namely how much an IdP can be trusted will be determined on the fly.

We consider the following use case as illustrated in Figure 1:

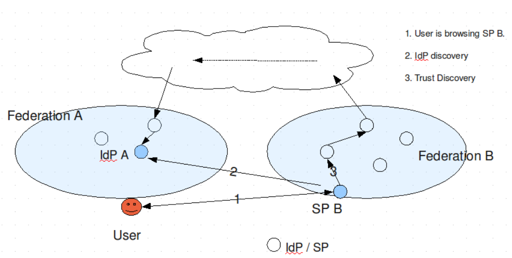


Figure 1. Use case of dynamic federation

1) A User is registered with an IdP A of federation A.
2) He is browsing SP B belonging to federation B.
3) SP B detects that IdP A is the preferred IdP for the browsing user by using a discovery service.
4) SP B gets the meta data of IdP A in order to determine if IdP A can be trusted and to what extent.
5) With a positive result SP B requests IdP A to authenticate the user.
6) IdP A authenticates the user and returns an assertion to SP B.
7) SP B authorizes the user to access the requested service.

In this case SP B does not need to know IdP A beforehand. The trust relationship between SP B and IdP A is created on the fly.

For the dynamic federation to be well defined, we give the following definitions:

Definition 1.1 (entity): An entity is an IdP or SP in a dynamic federation. When the dynamic federation is regarded as a network, we call an IdP or SP a peer of the network. Sometimes we also use the term *vertex* or *node* from graph theory.

Definition 1.2 (IdP discovery): IdP discovery is the process, by which an SP detects the preferred IdP of a user. The result of this process is the endpoint of the IdP.

Definition 1.3 (Trust discovery): Trust discovery is the process, by which an SP determines whether an unknown IdP can be trusted and vice versa. The result of this process is a binary decision or a trust value in the range of (0, 1).

Definition 1.4 (Trust value): The trust value is a subjec-

ISBN: 978-1-61208-101-4

tive quantification of how much an entity can trust another. The trust value depends on the path from truster to trustee, thus it is not a global value.

Definition 1.5 (Reputation value): The reputation value of an entity is a quantity derived from the underlying federation, which is globally visible to all members of the federation. An entity has in general different reputation values as seen by others, we call this personalized reputation. The reputation of an entity depends on the number of its incoming edges in the connection graph and on the reputation value propagated from its neighbors along these edges.

The topic *IdP discovery* is beyond the scope of this paper, in which we firstly focus on the issue *trust discovery*. If we treat a federation consisting of IdPs and SPs as a graph where the IdPs and SPs are vertices and the trust relationship are edges, we can reduce the problem of trust discovery to a question of pathfinding. In the following, we will compare different network models and analyze different pathfinding strategies.

II. Related work

OASIS defined two mechanisms for dynamic metadata publication and resolution [10]. However, it refers to the topic of IdP discovery rather than trust discovery. The trust model of the OASIS mechanisms still relies on X.509. The same drawback is encountered in approaches to dynamic SAML and dynamic metadata exchange [13]. To enable SAML for dynamic federations a generic solution with a SAML extension and a dynamic trust list was introduced [5]. However, there are no details regarding the SAML extension and dynamic trust list described in the paper. In addition, there was an approach to build dynamic federations in Grids [3]. This approach is based on WS-Trust 1.1 [4] and WS-Federation 1.0 [6] and hence, is not SAML-compatible.

III. Strategies of pathfinding

A. P2P networks

When we regard IdPs and SPs as peers of a network, they essentially form a type of peer-to-peer (P2P) trust network, because each entity provides and consumes trust information. In recent years, P2P networks were studied intensively. P2P networks have evolved in terms of their architecture from unstructured networks like Gnutella and Napster [17] to structured networks using distributed hash tables (DHT) like Chord [20]. Structured networks are more efficient and scalable, therefore almost all modern P2P networks belong to this type.

However, all of these systems mentioned above focus on the issue of how to find certain data in a P2P network within an acceptable bound such as in $O(logN)$ time and how to minimize the join / leave overhead of peers. Thus, they do not care whether the path to the targeted node is the shortest or the optimal one. In a network of trust an entity can not be trusted solely because of the fact that it is reachable in the network; Rather we need to calculate the trust value for a trust decision and / or for further authorization decisions. This means, we need to know not only whether a node can be found in a network but also the level of trust we associate with this node.

B. Classic routing protocols

Routing protocols in contrast to P2P networks focus on finding the optimal path between nodes. Distance vector routing protocols like RIP [14] are based on Bellman-Ford algorithms [7]. Within distance vector protocols, a route is defined as a vector with length and direction where the vector length is a generalization of the distance between source and destination. The advantage of distance vector routing protocols is their simplicity. They are easy to implement, configure and maintain. However, to avoid routing loops a maximum hop count and a hold-down-timer are introduced, this leads to major problems with scalability and convergence in the routing iterations (count-to-infinity problem).

Link-State Routing Protocols like OSPF [18] are based on the concept that routers flood information about the state of their adjacent neighbors, called link state to all nodes in a sub-network. Hence, each router will have a map of the entire sub-network after a certain time. Generally, a Dijkstra algorithm [12] is used to calculate the shortest path from the source node to a destination in the sub-network. Link-state routing protocols support complex topologies of large sub-networks and scale well. The Dijkstra algorithm used in link-state protocols can be implemented with less run time than the Bellman-Ford algorithm used in distance vector protocols. Because routers using the link-state protocol have a map of the entire sub-network, routing loops are rarely encountered. Generally, link-state protocols converge much more quickly than distance vector protocols and have no limitation on hop count. The trade-offs of link-state protocols are the expense of implementation and maintenance and the high requirement of CPU power and memory.

IV. The network model for dynamic federation

Following the above analysis we can draw the conclusion that routing protocols are better suited to forming a dynamic federation for AAI than P2P network models. This is because both routing protocols and dynamic federation address the same problem, i.e., how to find the best path from a given source node to a destination node in the network.

We decided to create the network model for dynamic federation by using a link-state model similar to OSPF, which consists of the following components:

- **Network Hierarchy**

Like OSPF the network of entities shall be divided into areas to reduce traffic overhead. For example, each federation of

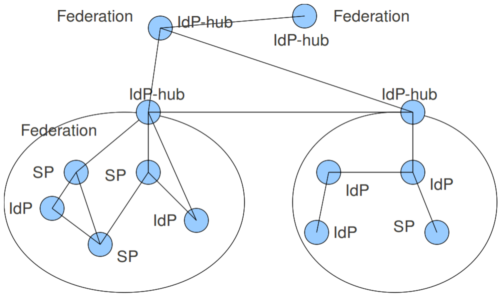


Figure 2. Example structure of dynamic federation

a con-federation can be treated as one area while the con-federation can be treated as the entire network. Each federation has an interface to the outside, which is responsible for inter-federation communication and only this interface is visible from outside. We call this interface the IdP-hub.

As Figure 2 shows, there are two levels in the hierarchy: IdP/SP and IdP-hub where the IdP-hub is considered to be pre-configured.

- **The link-state database**

We borrow the notion of LSA(Link State Advertisement) from OSPF. Each entity describes the link-state of its neighbors with LSAs and sends them to other entities in the network. Actually, an LSA is a list of IDs of all manually linked neighbors. When entity A has B's metadata, then A trusts B and B is an adjacent neighbor of A.

On the con-federation level, the IdP-hub provides federation information by sending federation-LSAs to other IdP-hubs in the con-federation to describe its federation to the outside. The federation-LSA contains at least the Id of its federation and the end location of the IdP-hub who is sending this LSA.

- **Trust table**

The trust table contains the metadata of all entities, it is effectively a metadata repository. In the trust table, all entities of a federation are stored with an entity-ID, a calculated trust value and the type of the entity, which is an IdP or SP, as Table I depicts.

Entity-ID	Value	Type
A	0.5	idp
B	0.8	sp
C	0.7	idp

Table I
EXAMPLE TRUST TABLE OF AN IDP OR SP

The trust table of an IdP-hub has an additional part, which contains entries for inter-federation relations. Here, the IdP-hubs of all other federations are stored. Each IdP-hub has an ID, which is its federation ID, and a calculated trust value. This value reflects the trust values between IdP-hubs as shown with an example in Table II, where F1 and F2 are idp-hubs of other federations.

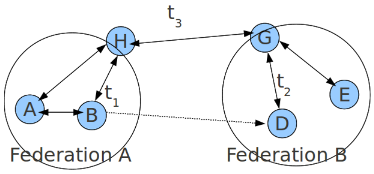


Figure 3. Calculation of inter-federation trust value

Entity-ID	Value	Type
A	0.5	idp
B	0.8	sp
C	0.7	idp
F1	0.3	idp-hub
F2	0.5	idp-hub

Table II
EXAMPLE TRUST TABLE OF AN IDP-HUB, NEEDED FOR INTER-FEDERATION RELATIONS

The trust table of all entities except IdP-hubs can be kept very small because the scope is limited to a single federation. According to a survey of the identity federations listed at the web site of Shibboleth [2] the size of the federations varies between 6 and 992 while the average number of entities is 205. When an SP receives a request from a user, it will first check his IdP in its own trust table. If there is no entry found, it will ask the IdP-hub responsible for its federation. The IdP-hub will check the Id of the destination federation in the part of its trust table dedicated for inter-federation relations. If there is an entry for the IdP-hub of the remote federation it will request the trust value of the respective IdP and compose the final trust value t as follows:

$$t = t_1 * t_2 * t_3$$

where t_1 is the trust value of the link between the IdP-hub and the SP, t_2 is the trust value of the link between the two IdP-hubs and t_3 is the trust value of the link between the remote IdP-hub and the respective IdP. Finally, the IdP-hub will send this composed trust value t back to the requesting SP. Figure 3 illustrate the calculation of interfederation trust value where the entity B is the SP mentioned above, the entity D is the respective IdP and the entity H and G are IdP-hubs of each respective federation.

- **Network protocol**

The SAML Assertion Query and Request Protocol [9] is used to exchange messages between entities. In doing this, HTTP POST Binding [8] is used to encapsulate SAML messages into a HTTP envelope. For the synchronization of the linkstate database and neighbor discovery an appropriate protocol named DYNFED similar to the OSPF protocol is under our development. Since the DYNFED protocol runs on top of the SAML Assertion Query and Request Protocol, we get a 3-layer protocol structure as shown below:

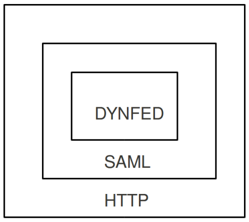


Figure 4. 3-layer protocol structure

V. The Trust model

A. *Trust value calculation*

We treat a federation as a directed graph $G(V, E)$ and define a weight function t: E → R by mapping edges to real-valued weights called trust values. We denote this as $t(v_i, v_j)$, where v_i and v_j are adjacent nodes. We follow the steps below to calculate the trust value between any two entities v_0 and v_n:

1) First we start with pre-defined values for the trust relation between any two neighboring entities v_i and v_j:

$$t(v_i, v_j) = \begin{cases} 1 & \text{if } v_i \text{ can verify } v_j\text{'s certificate with} \\ & \text{the certificate of a common certification} \\ & \text{authority (CA);} \\ 0.5 & \text{if } v_j\text{'s certifcate can be verified with} \\ & \text{the certificate of a trustworthy CA;} \\ 0.1 & \text{if } v_j\text{'s certificate is self-signed;} \end{cases} \tag{1}$$

The procedure is as follows: if a certificate can be verified by a trustworthy CA, then the authenticity of the certificate can be generally trusted. If the certificate can be verified by a common CA, then both entities normally belong to the same organization, and not only the authenticity of the certificate but also the certificate owner has a higher trust value.

2) The trust value of a path $p = (v_0, v_1, ..., v_n)$ between non-adjacent v_0 and v_n is the product of the trust values of its constituent edges:

$$t(p) = \prod_{i=1}^{n} t(v_{i-1}, v_i) * d \quad \text{where } 0 < d \leq 1 \tag{2}$$

Here, we use a constant d to dampen the trust value. The reason for this damping factor is as follows: second-hand evidence can not be considered as reliable as first-hand evidence. Every time an entity passes its trust value to its neighbor, i.e., the first-hand evidence, to the next entity along the path, the degree of this trust value shall be reduced. Obviously, we have $\lim_{n \to \infty} t(p) = 0$.

3) For the effective trust value in case of multiple paths between two entities v_0 and v_n we have:

$$\hat{t}(p) = \begin{cases} max(t(p)), & \text{if there is at least one} \\ & \text{path from } v_0 \text{ to } v_n, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Finally, based on the calculated trust value $\hat{t}(p)$ the node v_0 can make a binary decision with respect to a configurable threshold if node v_n can be trusted or not.

In order to calculate $max(t(p))$ we modify the Dijkstra algorithm [11] to find the path p from source v_0 to any vertex v_n with the largest trust value as shown in algorithm 1:

Algorithm 1 Modified Dijkstra's algorithm

```
function modified_Dijkstra $(G, v_0)$
    //$G(V, E)$ is a directed graph and $v_0$ is the source vertex
    for each vertex $(v_i, t(v_i)) \in V$ //$V$ is the vertex set of the input graph $G$
        $t(v_i) := 0$; //set the trust value of vertex $v_i$ to 0
    end
    $t(v_0) := 1$; //set the trust value of the source vertex to 1
    $S := \emptyset$; //S is an empty set;
    while $V$ is not empty
        take vertex $(v_i, t(v_i))$ from $V$, whereas its trust value $t(v_i)$ is the largest among all vertices in $V$;
        if $t(v_i) = 0$
            break; //all remaining vertices are inaccessible from $v_0$
        $S := S \cup \{(v_i, t(v_i))\}$; //add $v_i$ to $S$
        $V := V \setminus \{(v_i, t(v_i))\}$; //remove $v_i$ from $V$
        for each vertex $v_j \in Adj(v_i)$ //$v_j$ is neighbor of $v_i$
            if $t(v_j) < t(v_i) * t(v_i, v_j) * d$ //$t(v_i, v_j)$ is the trust value of edge $e : v_i \to v_j$
                $t(v_j) := t(v_i) * t(v_i, v_j) * d$;
            else
                do nothing;
        end
    end
    return $S$;
```

With this algorithm we firstly assign a trust value of zero to each vertex v_n of the directed graph G except of the source vertex v_0. The algorithm returns a set S, which contains all vertices, which can be reached from v_0, together with their corresponding trust values. After termination of the algorithm, the trust value $t(v_n)$ is equal to the maximum trust value $\hat{t}(p)$ from v_0 to v_n.

B. Attack resistance

An attack on a public key-based system of an identity federation means that some arbitrary faked target is accepted by the entire system. Two different types of attacks, namely node attack and edge attack are defined by Levien et al. in [16]. A node attack corresponds to stealing the secret keys of the victim and gaining total control of it. An edge attack corresponds to cheating the administrator of another entity to accept the attacker's certificate so that a trust relationship between the attacker's entity and the target entity can be established. An edge attack is easier to achieve than a node attack. If successful, the attacker can create many "bad" entities and deceives administrators of other IdPs or SPs into accepting his certificate and classifying it as trustworthy. Unfortunately, no trust metrics exists that can protect against node attacks efficiently as illustrated by Levien et al. [16]. Furthermore, the trust metric "shortest path" is also unable to protect against edge attacks. Thus we need a new measure to increase the attack resistance of our trust model. Levien discovered that the PageRank algorithm [19] is attack-resistant and can be used to protect against edge attacks [15]. While Google uses the PageRank algorithm to compute ranking of web pages we can use it to determine the reputation of entities by defining the computed PageRank as the reputation of each entity. Both the world wide web and an identity federation can be treated as a directed graph, hence, the PageRank algorithm can be applied to both systems. If the reputation of bad entities created by the attacker is considerably lower than the reputation of good entities, then we can easily detect those bad entities and exclude them from the system. This is why we use PageRank for our dynamic federation.

We give a recap of the PageRank algorithm, which is defined in [19] as follows:

$$R(u) = c * \sum_{v \in B_u} \frac{R(v)}{N_v} + c * E(u) \qquad (4)$$

where $R(u)$ denotes the PageRank of the web page u, B_u is the set of all web pages that point to u and N_v is the number of all outgoing links from the web page v. The factor c is used for normalization, so that the total rank of all web pages is constant. Furthermore, the PageRank of a web page can be treated as the probability that a random surfer lands on that page after lots of clicks. Because of the normalization it holds $\sum_u R(u) = 1$. The second part of equation (4), E is an initial vector over all web pages and represents the distribution of the probability that the surfer gets bored after several clicks and switches to a random page. One of the choices of E presented in [19] is a uniform distribution with $\sum_u E(u) = 0.15$, however, the authors noticed that a uniform distribution of E leads to a problem: some web pages with many related links like copyright warning, disclaimers and mailing list archives receive an overly high ranking. Thus, they introduced the "personalized" PageRank by selecting a distribution that E consists of only one single web page or several trusted web pages. By this means, the respective web pages get the highest PageRank value followed by their immediately linked pages. It means that the "personalized" PageRank prefers the chosen web pages considerably.

Now, we replace the web pages with IdPs and SPs as well as the personalized PageRank with the reputation value. We can then calculate a personalized reputation value for each IdP or SP. Like web pages, the more incoming edges an IdP or SP has, the higher its reputation value is. However, with a uniform distribution of E the reputation calculation according to the PageRank algorithm is not yet attack-resistant. An attacker can just generate many faked entities and interlink them with each other. Because of the uniform distribution of E, each entity regardless of whether the entity is trustworthy or not has the same contribution to increase the reputation of the faked entity. In contrast, by choosing the distribution vector E consisting of only a single entity, i.e., our own entity, we make the reputation calculation resistant against edge-attack for the respective entity. In this way, the entities generated by the attacker do not contribute to increase the reputation value because their probabilities in E are always equal to zero and hence are useless. Now, we can calculate the personalized reputation of arbitrary entity u regarding entity v_i (i.e., the reputation of u as seen by v_i) by defining the initial vector $E(u)$ from equation (4) as follows:

$$c * E(u) = \begin{cases} 1 - c, & \text{if } u = v_i \\ 0, & \text{otherwise} \end{cases} \qquad (5)$$

It means the vector $E(u)$ contains only the entity v_i. The exact value of c must be determined through simulations and experiments. In the original PageRank paper [19] c was set to 0.85.

Therefore each IdP or SP can use equation (4) and (5) to compute the personalized reputation for each other entity in the network. The fewer inbound edges an entity has, the lower its reputation is. Furthermore, the further an entity is located away from v_0, the lower its reputation is. With the personalized reputation it makes no sense for an attacker to create many entities pointing to his entity because its reputation will not get increased by this. Instead, he must create links to trusted entities, i.e., he must fake edges, the more the better. However, because the number of faked edges is proportional to the effort of an attacker, the introduction of reputation can significantly increase the system resistance against edge attacks.

Thus, we can combine the concept of reputation with the concept of trust values and extend equation (3) as follows:

$$\hat{t}(p) = \begin{cases} max(t(p)) * R_{v_0}(v_n), & \text{if there is at least one} \\ & \text{path from } v_0 \text{ to } v_n, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

where $R_{v_0}(v_n)$ is the reputation value of node v_n as seen by v_0.

VI. Conclusions

Approaches for AAI like Shibboleth bring many benefits for the solution of authentication and authorization and are widely-used today. Nevertheless, Shibboleth federations are based on static relationships, which do not scale well and can not be deployed in dynamic situations like mobile networks. With this issue in mind, we have developed a network model to build identity federations and con-federations in a dynamic manner. Similar to a routing protocol, the trust value of the shortest path between two entities is calculated and stored in a trust table. Furthermore, a reputation value is also calculated for each entity for the sake of attack resistance. Based on the trust value an entity is able to decide in real-time if another querying entity is trustworthy without having to know it a priori as such static meta data are not required.

In order to realize and test our network model, we aim to implement our design by extending SAML to bear and send trust values as well as reputation values and define protocols to synchronize the link-state database between entities. In the future, the design and realization of the protocol DYNFED, which was mentioned in Section IV will be completed. An implementation of the network model will be based on the framework of Shibboleth and a trust module for Shibboleth IdPs and SPs will be developed.

References

[1] Shibboleth System. URL `http://shibboleth. internet2. edu/ [2010.06.16]`.

[2] Shibboleth federations. URL `https://spaces. internet2.edu/display/SHIB/ ShibbolethFederations [2010.07.05]`.

[3] M. Ahsant, M. Surridge, T. Leonard, A. Krishna, and O. Mulmo. Dynamic trust federation in grids. *Trust Management*, pages 3–18, 2006.

[4] S. Anderson, J. Bohren, T. Boubez, M. Chanliau, G. Della-Libera, B. Dixon, P. Garg, M. Gudgin, P. Hallam-Baker, M. Hondo, et al. Web services trust language (ws-trust). *Public draft release, Actional Corporation, BEA Systems, Computer Associates International, International Business Machines Corporation, Layer*, 7.

[5] P. Arias Cabarcos, F. Almenárez Mendoza, A. Marín-López, and D. Díaz-Sánchez. Enabling SAML for Dynamic Identity Federation Management. *Wireless and Mobile Networking*, pages 173–184, 2009.

[6] S. Bajaj, G. Della-Libera, B. Dixon, M. Dusche, M. Hondo, M. Hur, C. Kaler, H. Lockhart, H. Maruyama, A. Nadalin, et al. Web Services Federation Language (WS-Federation). *BEA, IBM, Microsoft, RSA Security, Verisign*, 2003.

[7] D. P. Bertsekas and R. G. Gallaher. *Data Networks*. Prentice Hall, 1992.

[8] S. Cantor, F. Hirsch, J. Kemp, R. Philpott, and E. Maler. Bindings for the OASIS Security Assertion Markup Language (SAML) V2. 0, 2005.

[9] S. Cantor, J. Kemp, R. Philpott, and E. Maler. Assertions and protocols for the oasis security assertion markup language (saml) v2. 0, 2005.

[10] S. Cantor, I.J. Moreh, S.R. Philpott, and E. Maler. Metadata for the OASIS Security Assertion Markup Language (SAML) V2. 0, 2005.

[11] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. The MIT press, 2001.

[12] EW Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[13] P. Harding, L. Johansson, and N. Klingenstein. Dynamic Security Assertion Markup Language: Simplifying Single Sign-On. *IEEE Security and Privacy*, 6 (2):83–85, 2008.

[14] C. Hendrik. *Routing Information Protocol, RFC1058*. The Internet Society, 1988. URL `http://tools.ietf.org/html/rfc1058 [2010.02.08]`.

[15] R. Levien. Attack resistant trust metrics. 2003. URL `http://www.levien.com/thesis/compact.pdf [2010.02.09]`.

[16] R. Levien and A. Aiken. Attack-resistant trust metrics for public key certification. In *Proceedings of the 7th USENIX Security Symposium*, pages 229–242, 1998.

[17] D.S. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-peer computing. Technical report, HP Laboratories Palo Alto , HPL-2002-57 (R.1), 2003.

[18] J. Moy. RFC2328: OSPF Version 2. *RFC Editor United States*, 1998.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford InfoLab, 1998.

[20] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *ACM SIGCOMM'01*, 2001. URL `http://pdos.lcs.mit.edu/chord/ [2010.02.08]`.

A Distributed Workflow Platform for Simulation

Toàn Nguyên, Laurentiu Trifan
Project OPALE
INRIA Grenoble Rhône-Alpes
Grenoble, France
tnguyen@inrialpes.fr, trifan@inrialpes.fr

Jean-Antoine-Désidéri
Project OPALE
INRIA Sophia-Antipolis Méditerranée
Sophia-Antipolis, France
Jean-Antoine.Desideri@sophia.inria.fr

***Abstract*—This paper presents an approach to design, implement and deploy a simulation platform based on distributed workflows. It supports the smooth integration of existing software, e.g. Matlab, Scilab, Python, OpenFOAM, Paraview and user-defined programs. The contribution of the paper is a new feature which supports application-level fault-tolerance and exception-handling, i.e., resilience.**

Keywords-workflows; fault-tolerance; resilience; simulation; distributed systems

I. Introduction

Large-scale simulation applications are becoming standard in research laboratories and in the industry [1][2]. Because they involve a large variety of software and terabytes of data, moving around calculations and data files is not a simple process. Further, proprietary software and data often reside in locations from where they cannot be moved. Distributed computing infrastructures are therefore necessary [6][8].

This paper explores the design, implementation and use of a distributed simulation platform. It is based on a distributed workflow system and distributed computing resources. This infrastructure includes heterogeneous hardware and software components. Further, the application codes must interact in a timely, secure and effective manner. Additionally, because the coupling of remote hardware and software components are prone to run-time errors, sophisticated mechanisms are necessary to handle unexpected failures at the infrastructure and system levels. This is also critical for the coupled software that contribute to large-scale simulation applications. Consequently, specific approaches, methods and software tools are required to handle unexpected application behavior.

This paper addresses these issues. Section II is an overview of related work. Section III is a general description of a sample application, infrastructure, systems and application software. Section IV addresses resilience and asymmetric checkpointing issues. Section V gives an overview of the implementation using the YAWL workflow management system [4]. Section VI is a conclusion.

II. Related Work

Simulation is nowadays a prerequisite for product design and scientific breakthroughs in most application areas ranging from pharmacy, meteo, biology to climate modeling, that all require extensive simulations and testing [6][8]. They often need large-scale experiments, including long-lasting runs, tested against petabytes volumes of data on large multi-core supercomputers [10] [11].

In such application environments, various teams usually collaborate on several projects or part of projects. Computerized tools are shared and tightly or loosely coupled. Some codes may be remotely located and non-movable. This requires distributed code and data management facilities. Unfortunately, this is prone to unexpected errors and breakdowns.

Data replication and redundant computations have been proposed to prevent from random hardware and communication failures, as well as deadline-dependent scheduling [9].

Hardware and system level fault-tolerance in specific programming environments is also proposed, e.g. Charm++ [5]. Also, middleware and distributed computing systems usually support mechanisms to handle fault-tolerance. They call upon data provenance [12], data replication, redundant code execution, task replication and job migration, e.g., ProActive [17], VGrADS [15].

However, erratic application behavior needs also to be addressed. This implies evolution of the simulation process in the event of unexpected data values or unexpected control flow. Little has been done in this area. The primary concern of the application designers and users is that of efficiency and performance. Therefore, application erratic behavior is usually handled by re-designing and re-programming pieces of code and adjusting parameter values and bounds. This usually requires the simulations to be stopped and rebuilt [15].

Departing from these solutions, a dynamic approach is presented in the following sections. It supports the evolution of the application behavior using the introduction of new exception handling rules at run-time by the users, based on occurring (and possibly unexpected) data values. The running workflows do not need to be aborted, as new rules can be added at run-time without stopping the executing workflows [13]. At worst, they need to be paused.

This allows on-the-fly management of unexpected events. It allows also a permanent evolution of the applications, supporting their continuous adaptation to the occurrence of unforeseen situations. As new situations arise and new data values appear, new rules can be added to the workflows that will permanently be taken into account in the future. These evolutions are dynamically plugged-in to the workflows, without the need to stop the running applications. The overall application logics are therefore maintained unchanged. This guarantees a continuous adaptation to new situations without the need to redesign the existing workflows. Further, because exception-handling codes are themselves defined by new specific workflows plug-ins, the user interface to the applications remains unchanged [14].

III. APPLICATION TESTCASE

A. *Example Testcase*

An overview of a running tescase is presented here. It deals with the optimization of a car air-conditioning duct. The goal is to optimize the air flow inside the duct, maximizing the throughput and minimizing the air pressure and air speed discrepancies inside the duct. This example is provided by a car manufacturer and involves industry partners, e.g., software vendors, as well as optimization research teams (Figure 1).

The testcase is a dual faceted 2D and 3D example. Each facet involves different software for CAD modeling, e.g. CATIA and STAR-CCM+, numeric computations, e.g., Matlab and Scilab, and flow computations, e.g., Open FOAM and visualization, e.g., ParaView (Figure 1).

The testcase is deployed using the YAWL workflow management system [4]. The goal is to distribute the testcase on various partners' locations where the different software are running (Figure 2). In order to support this distributed computing approach, an open source middleware is used, namely: ProActive [17].

A first prototype was achieved using extensively the virtualization technologies (Figure 3), in particular Oracle VM VirtualBox®, formerly called Sun VirtualBox® [7]. This allowed experiments connecting virtual guest computers running heterogeneous software. These include Linux Fedora Core 12, Windows® 7 and Windows® XP on a range of local workstations and laptops (Figure 2).

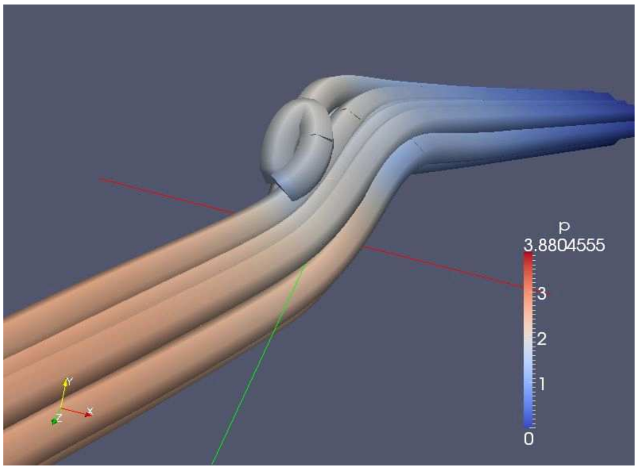


Figure 1. Pressure flow in the air-conditioning duct (ParaView display).

This work is performed for the OMD2 project, an acronym for *Optimisation Multi-Disciplinaire Distribuée*, i.e., Distributed Multi-Discipline Optimization, supported by the French National Research Agency ANR.

B. *Application Workflow*

In order to provide a simple and easy-to-use interface to the computing software, the YAWL workflow management system is used (Figure 2). It supports high-level graphic specifications for application design, deployment, execution and monitoring. It also supports the modeling of business organizations and interactions among heterogeneous software components. Indeed, the example testcase described above involves several codes written in Matlab, OpenFOAM and displayed using ParaView. The 3D testcase facet involves CAD files generated using CATIA and STAR-CCM+, flow calculations using OpenFOAM, Python scripts and visualization with ParaView. Future testcases will also require the use of the Scilab toolbox [16].

Because proprietary software are used, as well as open-source and in-house research codes, a secured network of connected computers is made available to the users, based on the ProActive middleware (Figure 5).

This network is deployed on the various partners' locations throughout France. Web servers accessed through the ssh protocol are used for the proprietary software running on dedicated servers, e.g., CATIA v5 and STAR-CCM+.

A powerful feature of the YAWL workflow system is that composite workflows can be defined hierarchically [4]. They can invoke external software, i.e., pieces of code written in whatever language is used by the users. They are called by custom YAWL services or local shell scripts. Web Services can also be invoked. Although custom services need Java classes to be implemented, all these features are natively supported in YAWL.

YAWL thus provides an abstraction layer that helps users design complex applications that may involve a large number of distributed components (Figure 3). Further, the workflow specifications allow alternative execution paths which may be chosen automatically or manually, depending on data values, as well as parallel branches, conditional branching and loops. Also, multiple instance tasks can execute in parallel for different data values. Combined with the run-time addition of code using the corresponding dynamic selection procedures, as well as new exception handling procedures (see Section IV), a very powerful environment is provided to the users [4].

IV. RESILIENCE

A. *Fault-tolerance*

The fault-tolerance mechanism provided by the underlying middleware copes with job and communication failures. Job failures or time-outs are handled by reassignment of computing resources and re-execution and of the jobs. Communication failures are handled by re-sending appropriate messages. Thus, hardware breakdowns are handled by re-assigning running jobs to other resources, which imply possible data movements to the corresponding resources. This is standard for most middleware [17].

B. Resilience

Resilience is commonly defined as “the ability to bounce back from tragedy” and as “resourcefulness” [18]. It is defined here as the ability for the applications to handle correctly unexpected run-time situations, possibly – but not necessarily – with the help of the users.

Usually, hardware, communication and software failures are handled using hard-coded fault-tolerance software [15]. This is the case for communication software and for middleware that take into account possible computer and network breakdowns at run-time. These mechanisms use for example data and packet replication and duplicate code execution to cope with these situations [5].

However, when unexpected situations occur at run-time, which are due to unexpected data values and application erratic behavior, very few options are offered to the users: ignore them or abort the execution, analyze the errors and later modify and restart the applications.

Optimized approaches can be implemented in such cases trying to reduce the amount of computations to be re-run, or anticipating potential discrepancies by multiplying some critical instances of the same computations. This latter approach can rely on statistical estimations of failures. Another approach for anticipation is to prevent total loss of computations by duplicating the calculations that are running on presumably failing nodes [9].

While these approaches deal with hardware and system failures, they do not cope with application failures. These can originate from:

- Incorrect or incomplete specifications.
- Incorrect or hazardous programming.
- Incorrect anticipation of data behavior, e.g., out-of-bounds data values.
- Incorrect constraint definitions, e.g., approximate boundary conditions.

To cope with this aspect of failures, we introduce an application-level fault management that we call *resilience*. It provides the ability for the applications to survive, i.e., to restart, in spite of their erroneous prevailing state. In such cases, new handling codes can be introduced dynamically by the users in the form of specific new component workflows.

This requires a roll-back to a consistent state that is defined by the users at critical checkpoints.

In order to do this efficiently, a mechanism is implemented to reduce the number of necessary checkpoints. It is based on user-defined rules. Indeed, the application designers and users are the only ones to have the expertise required to define appropriate corrective actions and characterize the critical checkpoints. No automatic mechanisms can be substituted for them, as is the case in hardware and system failures. It is generally not necessary to introduce checkpoints systematically, but only at specific locations of the application processes, e.g., only before parallel branches of the applications. We call this approach *asymmetric checkpoints*. This is described in Section D, below.

C. Exception Handling

The alternative used proposed here to cope with unexpected situation is based on the dynamic selection and exception handling mechanism featured by YAWL [13].

It provides the users with the ability to add at run-time new rules governing the application behavior and new pieces of code that will take care of the new situations.

For example, it allows for the runtime selection of alternative workflows, called worklets, based on the current (and possibly unexpected) data values. The application can therefore evolve over time without being stopped. It can also cope later with the new situations without being altered. This refinement process is therefore lasting over time and the obsolescence of the original workflows reduced.

The new worklets are defined and inserted in the original application workflow using the standard specification approach used by YAWL (Figure 2).

Because it is important that monitoring long-running applications be closely controlled by the users, this dynamic selection and exception handling mechanism also requires a user-defined probing mechanism that provides with the ability to suspend, evolve and restart the code dynamically.

For example, if the output pressure of an air-conditioning pipe is clearly off limits during a simulation run, the user must be able to suspend it as soon as he is aware of that situation. He can then take corrective actions, e.g., suspending the simulation, modifying some parameters or value ranges and restarting the process immediately. These actions can be recorded as new execution rules, stored as additional process description and invoked automatically in the future.

These features are used to implement the applications erratic behavior manager. This one is invoked by the users to restart the applications at the closest checkpoints after corrective actions have been manually performed, if necessary, e.g., modifying boundary conditions for some parameters. Because they have been defined by the users at critical locations in the workflows, the checkpoints can be later chosen automatically among the available asymmetric checkpoints available that are closest to the failure location in the workflow.

D. Asymmetric Checkpoints

Asymmetric checkpoints are defined by the users at critical execution locations in the application workflows. They are used to avoid the systematic insertion of checkpoints at all potential failure points. They are user-defined at specific locations, depending only on the application logic. Clearly, the applications designers and users are the only ones that have the domain expertise necessary to insert appropriately these checkpoints. In contrast with middleware fault-tolerance which can re-submit jobs and resend data packets, no automatic procedure can be implemented here. It is therefore based on a dynamically evolving set of heuristic rules.

This approach significantly reduces the number of necessary checkpoints to better concentrate on only those that have an impact on the applications runs [3].

For example (Figure 4):

- The checkpoints can be chosen by the users among those that follow long-running components and large data transfers.
- Alternatively, those that precede sequences of small components executions.

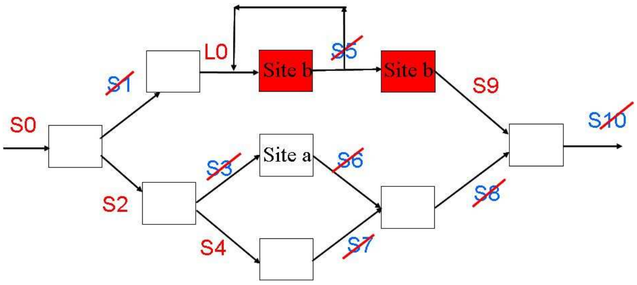


Figure 4. Asymmetric checkpoints example.

The basic rule set on which the asymmetric checkpoints are characterized is the following:

- R1: no output backup for specified join operations.
- R2: only one output backup for fork operations.
- R3: no intermediate result backup for user-specified sequences of operations.
- R4: no backup for user-specified local operations.
- R5: systematic backup for remote inputs.

This rule set can be evolved by the user dynamically, at any time during the application life-time, depending on the specific application requirements. This uses the native rule mechanism in YAWL [13].

V. IMPLEMENTATION

A. Resilience

Resilience is the ability for applications to handle unexpected behavior, e.g., erratic computations, abnormal result values, etc. It is inherent to the applications logic and programming. It is therefore different from systems or hardware errors and failures. The usual fault-tolerance mechanisms are therefore inappropriate here. They only cope with late symptoms, at best.

New mechanisms are therefore required to handle logic discrepancies in the applications, most of which are only discovered incrementally during the applications life-time, whatever projected exhaustive details are included at the application design time.

It is therefore important to provide the users with powerful monitoring features and to complement them with dynamic tools to evolve the applications specifications and behavior according to the future erratic behavior that will be observed during the application life-time.

This is supported here using the YAWL workflow system so-called “dynamic selection and exception handling mechanism” [4]. It supports:

- Application update using dynamically added rules specifying new worklets to be executed, based on data values and constraints.
- The persistence of these new rules to allow applications to handle correctly the future occurrences of the new cases.
- The dynamic extension of these sets of rules.
- The definition of the new worklets to be executed, using the native framework provided by the YAWL specification editor: the new worklets are new component workflows attached to the global composite application workflows [13].
- Worklets can invoke external programs written in any programming language through shell scripts, custom service invocations and Web Services [14].

B. Distributed workflows

The distributed workflows rely on the interface between the YAWL engine and the ProActive middleware (Figure 5). Users provide a specification of the simulation applications using the YAWL Editor. It supports a high-level abstract description of the simulation processes (Figure 2).

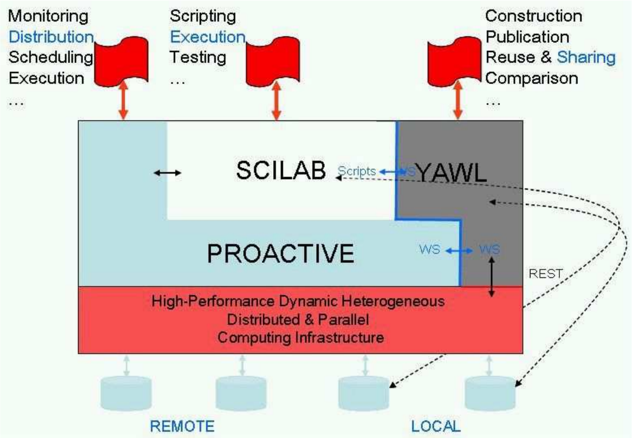


Figure 5. The OMD2 distributed simulation platform.

These processes are decomposed into components which can be other workflows or basic workitems. The basic workitems invoke executable tasks, e.g., shell scripts or so-called “custom services”. These custom services are specific execution units that call user-defined YAWL services. They support interactions with external and remote codes. In this particular platform, the remote external services are invoked through the ProActive middleware interface (Figure 6).

This interface delegates the distributed execution of the remote tasks to the ProActive middleware [17]. The middleware is in charge of the distributed resources allocation to the individual jobs, their scheduling, and the coordinated execution and result gathering of the individual tasks composing the jobs. The scheduler default policy is “best-effort”. However, users can implement their own policy, if desired. The middleware also takes in charge the fault-tolerance related to hardware, communications and system failures. The resilience, i.e., the application-level fault-tolerance is handled using the rules described in the previous sections.

The remote executions invoke the middleware functionalities through ProActive's Java API. The various modules invoked are the ProActive Scheduler, the Jobs definition module and the Tasks which compose the jobs. The jobs are allocated to the distributed computing resources based upon the scheduler policy. The tasks are dispatched based on the job scheduling and resource allocation. They invoke Java executables, possibly wrapping code written in other programming languages, e.g., Matlab, Scilab, Python, or calling other software, e.g., CATIA v5, STAR-CCM+, ParaView, etc.

Optionally, the workflow can invoke local tasks using shell scripts and remote tasks using Web Services. These options are standard in YAWL [4]. Calling the ProActive middleware is however necessary to run tasks on large multi-core clusters. ProActive is here in charge of the scheduling and resource allocation in these highly parallel environments, which YAWL does not support natively.

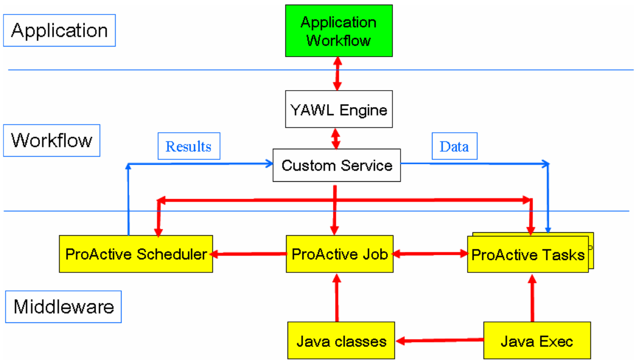


Figure 6. The YAWL workflow / ProActive middleware interface.

VI. CONCLUSION

The requirements for large-scale simulations make it necessary to deploy various software components on heterogeneous distributed computing infrastructures. These environments are often required to be distributed among a number of project partners for administrative and organizational purposes.

This paper presents an experiment for deploying a distributed simulation platform. It uses a network of high-performance computers connected by a middleware layer. Users interact dynamically with the applications using a distributed workflow system. It allows them to define, deploy and control the application executions.

A significant bonus of this approach is that besides fault-tolerance provided by the middleware, which handles communication, hardware and system failures, the users can define and handle the application failures at the workflow specification level.

This means that a new abstraction layer is introduced to cope with the application errors at run-time. Indeed, these errors do not necessarily result from programming and design errors. They may also result from unforeseen situations, data values and boundary conditions that could not be envisaged at first. This is often the case in simulations due to the experimental nature of the applications, e.g., discovering the behavior of the system being simulated, like unusual flight dynamics: characterization of the stall behavior of an aircraft for various load and balance profiles [2].

This provides support to resilience using an asymmetric checkpoints mechanism. This feature allows for efficient handling mechanisms to restart only those parts of an application that are characterized by the users as critical for overcoming erratic behavior.

Further, this approach can evolve dynamically, i.e., when applications are running. This uses the native dynamic selection and exception handling mechanism in the YAWL workflow system [4]. Should unexpected situations occur, it allows for new rules and new exception handlers to be plugged-in at run-time.

New testcases are currently being designed that involve large-scale (1000 CPU hours) simulations, e.g., car aerodynamics, running on a network of multi-core clusters.

ACKNOWLEDGMENT

This work is supported by the OMD2 project (*Optimisation Multi Disciplinaire Distribuée)* of the French National Research Agency ANR (*Agence Nationale de la Recherche*), grant ANR-08-COSI-007, program COSINUS (*Conception et Simulation*). The authors wish to thank all their partners for providing the testcases and for many fruitful discussions (http://www-opale.inrialpes.fr).

REFERENCES

[1] Y.Simmhan, R. Barga, C. van Ingen, E. Lazowska and A. Szalay "Building the Trident Scientific Workflow Workbench for Data Management in the Cloud". In proceedings of the *3rd Intl. Conf. on Advanced Engineering Computing and Applications in Science*. ADVCOMP'2009. Sliema (Malta). October 2009. pp 132-138.

[2] A. Abbas, "High Computing Power: A radical Change in Aircraft Design Process", In proceedings of the *2nd China-EU Workshop on Multi-Physics and RTD Collaboration in Aeronautics*. Harbin (China) April 2009.

[3] T. Nguyên and J-A Désidéri, "Dynamic Resilient Workflows for Collaborative Design", In proceedings of the 6th *Intl. Conf. on Cooperative Design, Visualization and Engineering*. Luxemburg. September 2009. Springer-Verlag. *LNCS 5738*, pp. 341–350 (2009)

[4] A.H.M ter Hofstede, W. Van der Aalst, M. Adams and N. Russell, "Modern Business Process Automation: YAWL and its support environment", *Springer* (2010).

[5] D. Mogilevsky,G.A. Koenig and Y. Yurcik, "Byzantine anomaly testing in Charm++: providing fault-tolerance and survivability for Charm++ empowered clusters", In proceedings of the 6th *IEEE Intl. Symp. On Cluster Computing and Grid Workshops*. CCGRIDW'06. Singapore. May 2006. pp146-154.

[6] E. Deelman and Y. Gil., "Managing Large-Scale Scientific Workflows in Distributed Environments: Experiences and Challenges", In proceedings of the 2nd IEEE Intl. Conf. on e-Science and the Grid. Amsterdam (NL). December 2006. pp 26-32.

[7] Oracle Corp. "Oracle VM VirtualBox, User Manual", version 3.2.0, May 2010. Also: http://www.virtualbox.org.

[8] M. Ghanem, N. Azam, M. Boniface and J. Ferris, "Grid-enabled workflows for industrial product design", In proceedings of the 2nd Intl. Conf. on e-Science and Grid Computing. Amsterdam (NL). December 2006. pp 88-92.

[9] G. Kandaswamy, A. Mandal and D.A. Reed, “Fault-tolerant and recovery of scientific workflows on computational grids”, In proceedings of the 8th Intl. Symp. On Cluster Computing and the Grid. 2008. pp 264-272.

[10] H. Simon. “Future directions in High-Performance Computing 2009-2018”. Lecture given at the ParCFD 2009 Conference. Moffett Field (Ca). May 2009.

[11] J. Wang, I. Altintas, C. Berkley, L. Gilbert and M.B. Jones, “A high-level distributed execution framework for scientific workflows”, In proceedings of the 4th *IEEE Intl. Conf. on eScience*. Indianapolis (In). December 2008. pp 156-164.

[12] D. Crawl and I. Altintas, “A Provenance-Based Fault Tolerance Mechanism for Scientific Workflows”, In proceedings of the 2nd *Intl. Provenance and Annotation Workshop*. IPAW 2008. Salt Lake City (UT). June 2008. Springer. LNCS 5272. pp 152-159.

[13] M. Adams, “Facilitating Dynamic Flexibility and Exception Handling for Workflows”, PhD Thesis, Queensland University of Technology, Brisbane (Aus.), 2007.

[14] M. Adams and L. Aldred, “The worklet custom service for YAWL, Installation and User Manual”, Beta-8 Release, Technical Report, Faculty of Information Technology, Queensland University of Technology, Brisbane (Aus.), October 2006.

[15] L. Ramakrishnan et al., “VGrADS: Enabling e-Science workflows on grids and clouds with fault tolerance”, Proc. ACM SC’09 Conf. Portland (Or.), November 2009.

[16] M. Baudin, “Introduction to Scilab”, Consortium Scilab. January 2010. Also: http://wiki.scilab.org/

[17] F. Baude et al., “An efficient framework for running applications on clusters, grids and clouds”, in “Cloud Computing: principles, systems and applications”, Springer 2010.

[18] http://edition.cnn.com/2009/TRAVEL/01/20/mumbai.overview/ last accessed: 07/07/2010.

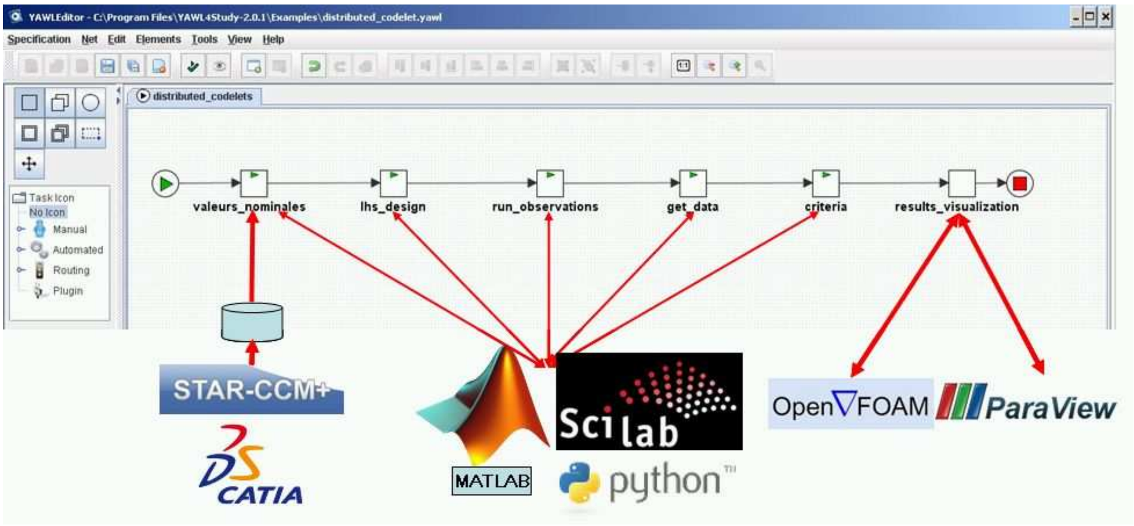


Figure 2. The YAWL workflow interface for the 2D testcase.

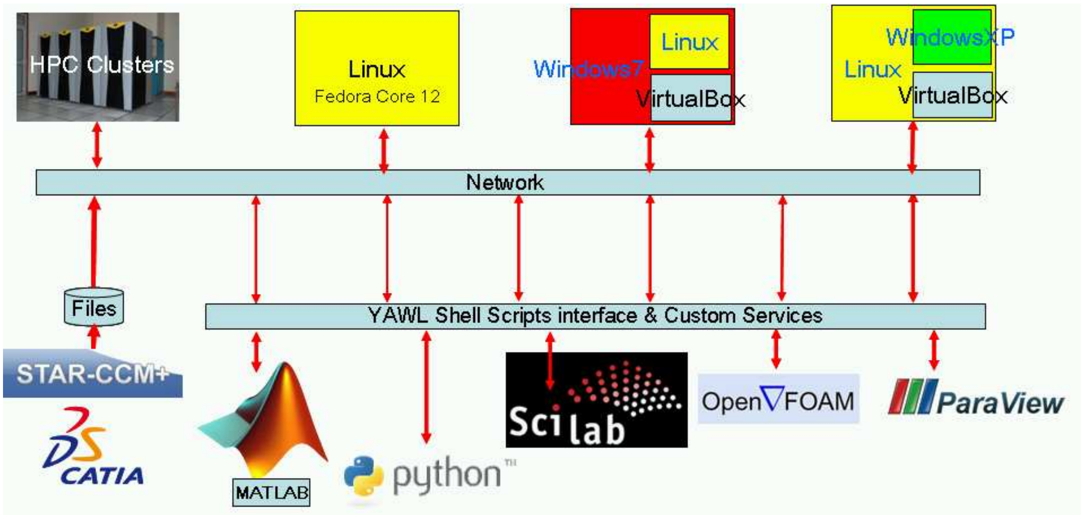


Figure 3. The virtualized distributed infrastructure.

A Web Interface Toolkit to Manage the AMGA Metadata Server Within the EGEE Standard Deployment

Víctor Méndez Muñoz, Gabriel Amorós Vicente, Carlos Escobar Ibañez, Mohammed Kaci, Javier Nadal Durà
Grid and e-Science Group.
Instituto de Física Corpuscular (IFIC)
- mixed institute of Consejo Superior de Investigaciones Científicas (CSIC) and Universitat de València (UV) -
Apt. Correus 22085, E-46071 , València, Spain
Corresponding author e-mail: vmendez@ific.uv.es, Other e-mails: {amoros,carlos.escobar,kaci,jvnadal}@ific.uv.es

***Abstract*—This paper describes a Web Graphical User Interface (GUI) toolkit for the access of the scientists to the official Enabling Grid for E-sciencE (EGEE) metadata server (AMGA). Such toolkit is composed of the amgaNavigator program and a library of PHP4 API to the AMGA server. The amgaNavigator is a high-level Grid interface designed for the current EGEE-III stable operating system (Scientific Linux 4) and the public package distribution. The contribution to the Computer Science is a toolkit, which integrates all the necessary deployment stacked on the EGEE standard platform, keeping the platform homogeneity with the rest of the infrastructure services. For the users, the amgaNavigator offers an exploration of the metadata schema and entries, with advanced searches in the catalog, avoiding the end-user to handle the AMGA SQL syntax.**

***Keywords*-Grid Computing; Information Interfaces; Software Engineering; Software Architecture; Reusable Software**

I. Introduction

The metadata service allows transparent access to information stored in distributed resources. Any metadata service for scientific purpose of different potential end-users and applications involving groups of many research centres, must integrates them on a standard and collaborative information system. For this reason, the gLite, which is the EGEE middleware [1], has adopted the AMGA as the metadata service [2]. The integration in a Grid infrastructure allows the access to the metadata associated to the Virtual Organization (VO), which the user belongs to. The permissions and flexibility of the account management that offers the VO context, is translated to AMGA smoothly. The issues of security, privacy and other related to AMGA, are explained in other works [3,4]. One clear example of AMGA use is for e-Health where typically hospitals or medical physics research centres, have large amount of information that, by legal issues, are not allowed to be exported outside their buildings. However, very often, the studies or the statistics needed by them, must include data stored in several of these centres. Thus, AMGA offers the possibility to collect metadata about this data and facilitate the search and the access to the data keeping on it the rights and permissions. A concrete example is a scientist involved in a study collaboration with several hospitals, to which is necesary the access to specific information with several keywords. He can use AMGA searching with the keywords and, since he has the correct permissions, AMGA acts as a portal to bring him the data that is returned by the documents found with the search, and only gives access to the data related to the study.

In this scientific context, we have identified some metadata services requirements in our research institute. In addition to the user metadata requirements, we have considered our complex Grid infrastructure context. Since we have deployed two Grid infrastructures in our institute, the platform and middleware standardized deployment is a key issue for efficient administration. The first of these infrastructures is a Tier-2 / Tier-3 of the Large Hadron Collider (LHC) Europe Southwest regional Grid infrastructure [5], supporting Grid services on LHC Computing Grid (LCG) basis, mainly for the ATLAS experiment. The other infrastructure is a Grid-CSIC [6] site running scientific applications of all the CSIC institutes. The Grid-CSIC deployment is keeping homogeneity with the EGEE platform, the package distribution and the middleware standards. An additional reason for a standardized deployment, is the calling to integrate the Grid-CSIC in the future NGI/EGI, which starts to operate following and enhancing the EGEE-III in spring of 2010 [7].

The use of AMGA will provide the metadata features to the Medical Physics Grid environment of our researchers and collaborators [8,9], and potentially other scientific applications with metadata requirements. The amgaNavigator toolkit is designed to improve the metadata accessibility by the user point of view, integrating and connecting other Grid services of our complex Grid infrastructure, and providing a technical solution for the standardized deployment of all the services involved, not only the middleware but also the platform dependent components like the databases or the web servers.

In the access to the AMGA service, instead of the native command line client, our end-users need a GUI. The available AMGA GUIs does not cover our special requirements. For deployment reasons we need a Web client

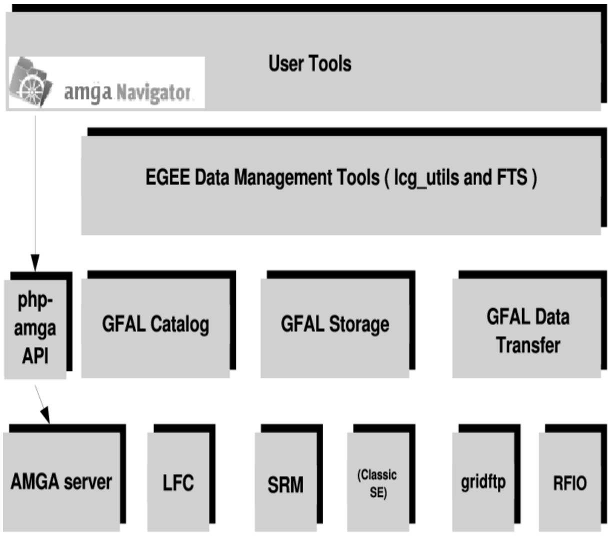


Figure 1. The amgaNavigator framework

accessible for our contributors over the world, without any client installation and support, just the Web browser. For maintenance reasons, the architecture may use the stable EGEE operating system, nowadays the Scientific Linux 4 and the corresponding packages distribution. We have developed a down-porting of AMGA API on PHP5 [10] to the public EGEE stable release on PHP4, and we have built amgaNavigator upon this API following the EGEE standards. The rest of this paper is a related work Section of the catalog components and the metadata user interface. Following, it is described a design Section of the amgaNavigator, which also explains the motivations of this application. We continue with the amgaNavigator detailed functionalities description, and finally conclusions and future works.

II. Related Work

The EGEE architecture is partially web services oriented, mostly in the job scheduling services. Moreover, part of the architecture is component-based, as a result of historical evolution of the gLite middleware. Such components are accessible through wrappers supporting the service oriented functionalities. In Figure 1 it is shown the AMGA service as part of the EGEE architecture. It is a basic grid service used by Grid File Access Library (GFAL) for the catalog management. GFAL is an abstraction of the storage, catalog and transfer specific services. GFAL is used by lcg utilities and high-level transfer services, which furthermore of the high-level functionality it support the VOs context and permissions. The AMGA service is also directly accessible by end-users or applications [11], without mediation of the stacked services.

When metadata access is deployed, two services give the main catalog management functionalities: the LHC File Catalog (LFC), or other catalog services like FiReMan [12], and the AMGA metadata service. AMGA is a metadata catalog organized as a filesystem, where we can find directories with schemas defined by different attributes for each directory. For any Logical File Name (LFN) we will find an entry in the AMGA server, which gives values for the schema attributes. Such values of the attributes are the metadata associated to the file. The AMGA relates the LFN with the Grid Unique Identifier (GUID), but does not provide any information about physical location of the file replicas, which should be supplied by the catalog service.

The AMGA client is a command line shell. This is a rude access to the service, specially when end-users have to interact with the AMGA for common operations. Therefore, some research groups have developed AMGA GUIs for different purposes.

Amga Browser [13] is a Python client for generic access to AMGA server. Amga Browser allows a graphical exploration of the metadata schema, and a command line launcher and text results screen. The commands are on AMGA SQL like, so the end-user must have some knowledge of such syntax.

LHCb Book-keeping Database Browser [13], is a very specific AMGA utility for browsing of the LHCb logging and booking system. It is a Java and Python client integrated in the Ganga Grid UI.

The INFN team have developed the AMGA server and also some Web GUIs integrated on applications designed for specific environments. The first one is the INFN-Catania Web Front-end, a GUI to access all the metadata related to gMOD, the Grid Movie On Demand service of Genius Portal [14]. The AMGA WI, a metadata Web Interface of GILDA [15] project, developed based on P-GRADE [16] with Java technologies using the AMGA Java API [17]. Another example is gLibrary [10] the Grid digital assets management software based on AMGA PHP5 API.

III. Design

There are some reasons that motivate the design and implementation of the new AMGA UI in our complex Grid infrastructure.

- We need an AMGA Web GUI, for an affordable deployment and maintenance of the client side.
- The AMGA Web GUI must have an available open-source distribution to support the application modifications and integration on our scientific platforms, without third-party provider dependencies.
- The AMGA Web GUI server side should run over the stable Scientific Linux, nowadays SL4, and the package software dependencies should be the stable package distribution, which includes PHP4 and MySQL4, for homogeneous administration.
- The AMGA Web GUI software architecture should be reusable for future integration of other services.
- The AMGA Web GUI should be designed to fulfil the end-user requirements.

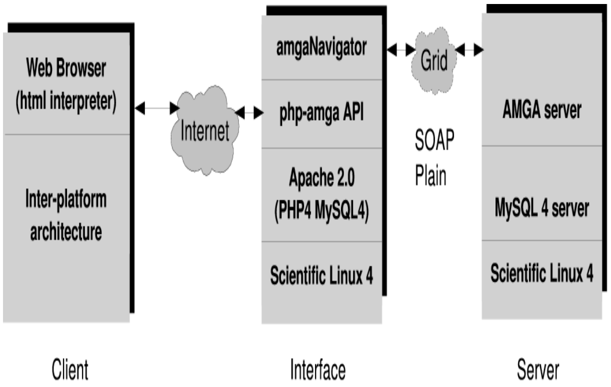


Figure 2. The amgaNavigator client/server

Figure 1 shows the architecture framework of the application. We can see that amgaNavigator offers a high-level middleware service on the user tools layer. The amgaNavigator uses amga-php API for direct access to the AMGA server, regardless of the EGEE stack architecture (GFAL, LCG utilities and file services). On such architecture framework, the amgaNavigator may access other data management services, to integrate or launch user operations, for example, catalog services for replica location and selection or VOMS for the X509 certificate authorization.

In Figure 2, we find the amgaNavigator client/server schema. The client environment can be a platform where a Web browser is installed. The Web browser receives from the amgaNavigator the html tags and embedded information and sends the data forms. In the server side the architecture is stacked as shown in Figure 2. The Apache 2.0 server is configured with PHP4 and MySQL4 modules. PHP provides dynamic Web execution on Apache server. The amgaNavigator uses the php-amga API access to the AMGA server, with MySQL database back-end. The AMGA server offers other back-ends like PostgreSQL. The php-amga communication with AMGA server is possible with SOAP or connection oriented using plain information on TCP sockets. The different servers involved with amgaNavigator: MySQL, AMGA and Apache, can be deployed on a local or remote host.

The design of amgaNavigator has been raised with reusable software techniques in the design of the API library to connect the AMGA server. We need an application that uses the distribution of the stable release EGEE platform. Thus, starting from the official amga-php API, nowadays on PHP5, we have ported to PHP4 and we have included some functionalities, mainly to parse the AMGA strings to fulfil amga-php syntax and to retrieve results from the API. The updated amga-php API on PHP4 would be valid for next PHP releases, due to descent compatibility feature of PHP project. This PHP class comes with License for EGEE Middleware, which basically is an opensource with some restrictions for distribution out of the scientific and academic environments.

We also apply reusable engineering of PHP File Navigator (PFN) [18], with GPL distribution, to reduce the implementation and debugging time. Therefore, the obtained amgaNavigator is also GPL distributed, which increases the possible collaborative work comparing with proprietary licensing restrictions. We have used the CSS Styles, dynamic language support (PFN vars class), configuring functionalities (PFN conf class), icon and imaging management (PFN imx class) and we did re-engineering with some parts of the PHP presentation templates.

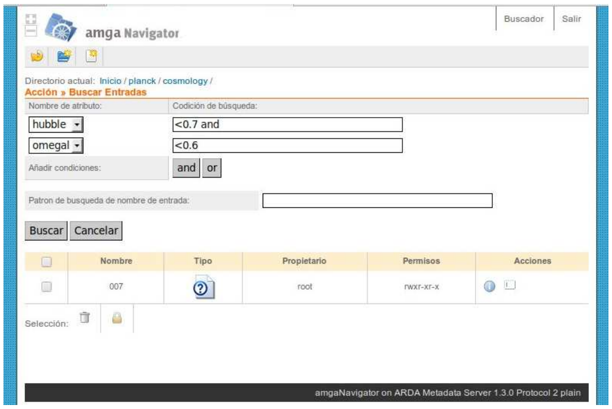
IV. Functionalities

We have considered the common AMGA user operations and we have implemented them in the amgaNavigator:

- Control access
- Browse the AMGA directories and entries structure
- Create directories and assign attributes to them
- View and modify directories attributes
- Delete directories or their attributes
- Create entries and assign attribute values to them
- View and modify the attributes entries values
- Delete entries
- Advanced search by multiple conditions
- View and modify the file permissions
- Multiple delete operations
- Multiple permissions assignment operations.

AmgaNavigator offers dynamic language support. The main components of the web homepage may be summarize as following. The header has the main menu, wich has some icons to update information of actual directory browsing, create directory form and create entry form. The top right main menu offers links to advanced search and exit. On the body of the web page, the user can explore into directories getting their contents. Each line of the list is related to a directory or entry information: name, type, owner, permissions, and associated actions for the individual entry or directory (edit, permissions, delete). The list has check boxes for multiple operations, launched on the bottom of the list. The possible multiple actions are delete directories or entries and set permissions.

The rest of the web pages, reports and forms, are composed with the main components and other specific functionalities components as well. An example of advanced search is shown in Figure 3. Regardless of the mentioned main graphical components, the specific components allow the user to select any attribute from a pull-down list to define the searching conditions. Figure 3 presents the result of the attributes condition search for *hubble<0.7 and omega<0.6* in the directory */Inicio/planck/cosmology*. If more conditions want to be introduced, then it has to be clicked the *and/or* buttons. Below those buttons we can see that no pattern has been introduced as condition of the entrie names. Only file *007* satisfied all the search conditions.


amga Navigator
Buscador
Salir
and
or
Buscar
Cancelar
Selección:
amgaNavigator on ARDA Metadata Server 1.3.0 Protocol 2 plain

Directorio actual: Inicio / planck / cosmology /
Acción » Buscar Entradas

Nombre de atributo:	Codición de búsqueda:
hubble	<0.7 and
omegal	<0.6
Añadir condiciones:	and or

Patron de busqueda de nombre de entrada:

□	Nombre	Tipo	Propietario	Permisos	Acciones
□	007	?	root	rwxr-xr-x	

Figure 3. Advanced search screen shot example

The amgaNavigator attends user specifications on metadata common operations, with advanced graphical mechanisms not available in the AMGA server, like fast operations on repetitive task when is needed, for example, adding entries. Other improvements on the accessibility to the native service are the multiple operations with single step for selected items; the modify operations integrated with the corresponding display values; or the visual icon information and links to browse. The users have on a single web page all the necessary information for any operation of their common metadata requirements, with improved usability over AMGA server.

V. Conclusions and Future Work

We have developed a browser based on probed code components. It has been developed with an easy to maintain criteria, following the POSIX code presentation and the PHP object oriented scalable code structure. By the user point of view amgaNavigator offers an interface integrated with the EGEE data management to improve the accessibility of the AMGA server, providing additional functionalities. All the necessary deployment is also possible following the EGEE standards, which is an important computer administration issue. For this purpose we have developed a down-porting of the amga-php API, to the stable PHP4 package on SL4 release. In this manner we keep more homogeneity within the complex Grid infrastructure.

The software architecture is designed to integrate future requirements of other EGEE data services. Additional future work is the implementation of advanced user features, like schema copies, or functions not contemplated on AMGA server, like the recursive search. When our Medical Physics Grid environment will reach a production state, a critical mass of users will provide the feedback of amgaNavigator accessibility and aditional services integration requirements.

Acknowledgements

We are greatly indebted to the founding agency Spanish National Research Council-CSIC for their support from project Grid-CSIC.

References

[1] M. Lamanna, M.-E. Begin, Z. Sekera, and S. Collings, "Egee middleware architecture," CERN (EGEE-DJRA1.1-594698-v1.0), Tech. Rep., 2005.

[2] M. Schulz, "glite data management and information system," in *EGEE User Forum*. OGF20/EGEE, 2007.

[3] J. Montagnat, D. Jouvenot, C. Pera, A. Frohner, P. Kunszt, B. Koblitz, N. Santos, and C. Loomis, "Bridging clinical information systems and grid middleware: a medical data manager," *Studies in health technology and informatics*, vol. 120, pp. 14–24, 2006.

[4] J. Montagnat, A. Frohner, D. Jouvenot, C. Pera, P. Kunszt, B. Koblitz, N. Santos, C. Loomis, R. Texier, D. Lingrand, P. Guio, R. D. Rocha, A. S. de Almeida, and Z. Farkas, "A secure grid medical data manager inteface to glite middleware," *Journal of Grid Computing*, vol. 6, pp. 45–59, 2007.

[5] S. G. de la Hoz, L. March, E. Ros, J. Sánchez, G. Amorós, F. Fassi, A. Fernández, M. Kaci, and A. Lamas, "Analysis facility infrastructure (tier-3) for atlas experiment," *The European Physical Journal C*, vol. 54, pp. 691–697, 2008.

[6] Grid-CSIC-Project, "Grid infrastructure for advanced research at the spanish national research council (grid-csic)," 2010, http://www.grid.csic.es/.

[7] WP5, "Egi blueprint: Design study," CERN, Tech. Rep., 2008, http://web.eu-egi.eu/fileadmin/public/EGI_DS_D5_3_V300b.pdf.

[8] G. Amorós, F. Albiol, J. Ors, A. Fernández, S. González, M. Kaci, A. Lamas, L. March, E. Oliver, J. Salt, J. Sánchez, M. Villaplana, and R. Vives, "Scientific applications running at IFIC using the grid technologies within the s-science framework," in *Third International Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP)*. IEEE, 2009.

[9] Partner-Project, "Particle training network for european radiotherapy," 2010, http://cern.ch/partner.

[10] A. Calanducci, C. Cherubino, L. N. Ciuffo, and D. Scardaci, "glibrary: Digital asset management system for the grid," in *Conference on Hypermedia And Grid Systems*. IEEE, 2007.

[11] N. Santos and B. Koblitz, "Distributed metadata with the amga metadata catalog," in *Workshop on Next-Generation Distributed Data Management (HPDC-15)*. ACM, 2006.

[12] C. Munro, B. Koblitz, N. Santos, and A. Khan, "Performance comparison of the lcg2 and glite file catalogues," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 559, no. 1, pp. 48 – 52, 2006.

[13] D.Piparo, "Two graphical browsers for the amga metadata cataloge," in *Nuclear Physics B, Proceedings Supplements*. Elsevier B.V., 2007, pp. 311–313.

[14] Genius-Portal, "Grid enabled web environment for site independent user job submission," 2009, https://glite-tutor.ct.infn.it.

[15] GILDA-Portal, "Grid INFN laboratory for dissemination activities," 2010, https://gilda.ct.infn.it/testbed.html.

[16] P-Grade-Portal, "Parallel grid run-time and application development environment," 2010, http://www.p-grade.hu.

[17] S. Scifo and V. Milazzo, "Amga wi - amga web interface," in *Conference on Hypermedia And Grid Systems*. IEEE, 2007.

[18] PFN-Project, "Php file navigator," 2010, http://pfn.sourceforge.net.

Using Simulated Annealing to Improve Reliability of Grid Computing Systems

Ehsan Gholami
Islamic Azad University Shoushtar
Branch, Shoushtar, Iran
eh.gholami@gmail.com

Amir Masoud Rahmani
Islamic Azad University Science
and Research Branch, Tehran, Iran
rahmani@srbiau.ac.ir

Reza Farshidi
Islamic Azad University Dezful
Branch, Dezful, Iran
farshidi@iaud.ac.ir

***Abstract*—Grid computing system has been suggested as an effective technology in distributed resource coupling for applications that require large-scale resource sharing, where processors and communication have significantly influence on grid computing system reliability. A grid computing system contains a series of machines, resources, programs and communication network. Each machine includes a series of resources and programs. In designing a grid computing system, a topology is specified by distributing the resources and programs in grid computing system machines and defining communication network between the machines, which have significant effect on grid computing system reliability. Finding such a topology that guarantees the maximum reliability is difficult and considered as an NP-Hard problem. In this work, a novel method is proposed for designing and improving the grid computing system reliability based on Simulated Annealing. The proposed method has this ability to obtain a new topology with more reliability in comparison to the initial given grid computing system.**

Keywords-topology; reliability; Grid computing system; simulated annealing.

I. INTRODUCTION

In the recent years, Grid Computing Systems (GCS) have been created as a way to increase the capacity of computing resources like processing and memory resources by integrating them and use computing resources, which are distributed geographically [1]. In GCS, the integration of shared computing resources, e.g., computer, software, data, and peripheral equipments are achieved through a communication network [1][2].

GCS can be built in various sizes, ranging from just a few machines in a department to groups of machines organized as a hierarchy spanning the world. Machines that participate in the GCS may include systems from multiple departments with the same organization. Such a grid is also referred as an intragrid [1]. In this paper, we focus on reliability of this type of grid and propose a new algorithm to design and improve the reliability of intragrid that guarantees the maximum reliability.

A GCS contains a series of machines, resources, programs and a communication networks. Each machine in GCSs can be a computer, data storage device or other devices. Every machine includes a series of programs and resources such as software, data, etc. These programs can start their execution from this machine. Machines are connected via the communication network. In a GCS, each program completes its own execution by connecting to some grid system machines via communication network and exchanging the information with their resources.

The reliability of a program on a GCS is the probability that one program may exchange information with other machines successfully, and completes its own execution. Likewise, the reliability of a grid computing system is the probability that all programs on a grid computing system are executed successfully [2][3]. This reliability can be calculated by determining the reliability of each component using Monte Carlo [14] or any other method such as Grid system reliability [2] and Markov model [3]. Therefore, the way that programs access to required resources in remote machines and also the circumstances of distributing resources and programs between machines have significant effect on GCS reliability. To define the GCS topology, communication network between machines and distributing resources and programs in GCS machines must be defined in a way that guarantees maximum reliability against failures in GCS components [2][14]. Many different topologies can be defined for a GCS with specific number machines, programs and resources. Finding the best GCS topology from the viewpoint of reliability between these topologies is an NP-Hard problem. Since there is not any standard algorithm, which guarantees an optimized reliability of GCS, in this paper, Simulated Annealing as a meta-heuristic method is presented for designing GCS topology and improving reliability of GCS. Furthermore, other methods such as Genetic algorithms do not offer any guarantee of global convergence to an optimal point to solve such problems [6]. But the proposed method has better results for designing GCS topology in shorter run time.

The rest of this paper is organized as below. Section II describes the simulated annealing. Section III defines the designing of GCS topology. Sections IV, V and VI develop the algorithm for improving reliability of GCS. In Section VII, experimental results are illustrated. Section VIII concludes the paper.

II. SIMULATED ANNEALING

Simulated Annealing (SA) is one of the probabilistic algorithms presented by Kirkpatrick in 1987 to solve optimizing problems with large search span of solutions [10]. Basically, SA is a random-search technique, which exploits an analogy between the way in which a metal cools and freezes into a minimum energy crystalline structure (the annealing process) and the search for a minimum in a more general system; it forms the basis of an optimization technique for combinatorial problems [16]. Search span of a solution for a problem is called *S* and each solution in this search span is called *s,* where $s \in S$. The SA algorithm begins from an initial solution s_0 as current solution and gradually obtains the optimized solution by passing from a solution to another solution in search span and choosing a neighbor solution of current solution. The solution such as s_1 from search span is chosen as neighbor solution of the current solution. Current solution is replaced with neighbor solution or remains unchanged by a probability. This process iterates until finding an optimized solution or reaching to maximum number of iteration in SA.

In this algorithm, a better solution for the problem has a better cost. Accepting the neighbor solution as current solution in next iteration is based on a strategy. According to this strategy if the cost of neighbor solution is better than current solution it will be accepted as the current solution in next algorithm iteration and if the cost of neighbor solution is worse, it will be accepted by a probability, which will be mentioned later. Therefore, by this manner SA can leave local optimized solution. In SA, temperature parameter *T* is defined to obtain probability of acceptance of neighbor solution. When the algorithm starts, initial value of parameter *T* is defined such that most neighbor solutions in algorithm iterations are accepted as alternative for current solution. Parameter *T* is decreased gradually by iterations of SA algorithm such that it reaches to zero before the number of iterations of algorithm reaches to last (maximum) iteration. Therefore, by increasing the number of iterations, parameter *T* is decreased and the probability of accepting neighbor solution is decreased. So, in last iteration of algorithm, only the neighbor solution which has better cost than current solution is replaced as current solution in next algorithm iteration. In this algorithm, decreasing the parameter *T* in each iteration, obtaining neighbor solution, determining initial temperature T_0 and maximum number of iterations of algorithm *(MAX)* have significant effect on obtaining best or optimized solution. SA pseudo-code is shown in Fig. 1.

III. PROBLEM DEFINITION

In this paper, GCS topology is designed in two separate steps: first step is distributing the GCS resources and programs between GCS machines. The second step is defining a communication network for these machines. To design such a GCS, one of the problems is the definition of its topology to guarantee a maximum reliability for the GCS. In this work, we tackle the following version of this problem: given an user-defined network topology, find a more reliable alternative GCS in real system. Therefore, reaching to a GCS topology with high reliability is possible by changing the results of each step or both of them and combining them to obtain a topology with a more reliability. Searching for such a topology is an NP-hard problem, because there are many different topologies during implementation of first and second steps. Solving these problems with conventional algorithms is so difficult and time consuming. In this work, an intra GCS is assumed as a graph *G(V, E)* with *n* nodes V={v_1,v_2, ... ,v_n} and *m* edges E={e_1,e_2, …. ,e_m}, where GCS machines are its nodes and communication network links are its edges.

```
$t = T_0$
$Initial\ solution = S_0$
$Best\ solution = S_0$
$OF_0 = Objective\ fumction(S_0)$
$OF_{Best\ solution} = OF_0$
$for\ i = 1\ to\ Max\ do$
  { $S_1 = Generate\ neighbor(S_0)$
    $OF_1 = Objective\ function(S_1)$
    $If\ OF_1 \geq OF_0\ then$
      { $S_0 = S_1$
        $OF_0 = OF_1$
        $If\ OF_1 \geq OF_{Best\ solution}\ then$
          { $Best\ solution = S_1$
            $OF_{Best\ solution} = OF_1$ }
      }
    $else$ { $Accept = e^{-(\frac{OF_0 - OF_1}{t})}$
             $r = Random(0,1)$
             $if\ \ r > Accept\ \ then$
             { $S_0 = S_1$
               $OF_0 = OF_1$ }
    }
    $t = decrease(t)$
  } // $End for$
$Return\ (Best\ solution)$
```

Figure 1. SA Algorithm pseudo-code

Reliability of each machine depends on number of its resources and programs and reliability of each link depends on programs and machines that transfer their data through this link [2][14]. Also, in this work reliability of GCS is calculated based on Monte Carlo simulation according to the reliability of machines and links [14].

According to SA, algorithm must begin from an initial solution. In this work, the SA algorithm begins from initial topology as initial solution and improves the reliability of GCS, too. The initial topology can be defined by user or randomly. Implementing SA algorithm to find a GCS topology and improving its reliability include some parts, which are considerable for obtaining best solution. These parts are mentioned in next sections.

IV. NEIGHBORHOOD STRUCTURE

In designing GCS, resources and programs in GCS machines are distributed by the designer before implementing and creating GCS, such that the designer denotes how the software, data storage devices and other devices are distributed in GCS machines and also denotes machines, from which programs start their execution. Thus, the proposed method can be used in designing real intragrid system to achieve at more reliability. Obtaining a neighbor topology from current topology of GCS has significant effect on faster and better convergence of SA to global optimum topology from the viewpoint of GCS reliability. In this work, two methods are proposed to obtain a neighbor topology from current topology in GCS:

A. Obtain Neighbor Topology by Exchanging the Resources and Programs of Machines

In this method, at first, two machines are selected randomly from current topology of GCS. The neighbor topology can be obtained by changing all resources and programs or part of them in these two selected machines of current topology. Note that the resources and programs of other machines remain unchangeable. For example if *i, j* machines are selected in current topology the neighbor topology is obtained by exchanging resources and programs of these two machines. It has been shown in Fig. 2. The reason of obtaining neighbor in such manner is that in GCS, distributing resources and programs has direct effect on machine reliability and amount of data exchanging between machines, which consequently have affects on reliability of machines and GCS [2][3][11]. Therefore, exchanging the resources and programs might decrease the workload and amount of data exchanging, which consequently increases reliability of GCS.

Node j	
Program	P_3, P_4
Resource	R_1, R_2, R_5
Neighbors	$N1, N2, N5$

Node i	
Program	P_1, P_2
Resource	R_1, R_3, R_4
Neighbors	$N3, N4$

(a)

Node j	
Program	P_1, P_2
Resource	R_1, R_3, R_4
Neighbors	$N1, N2, N5$

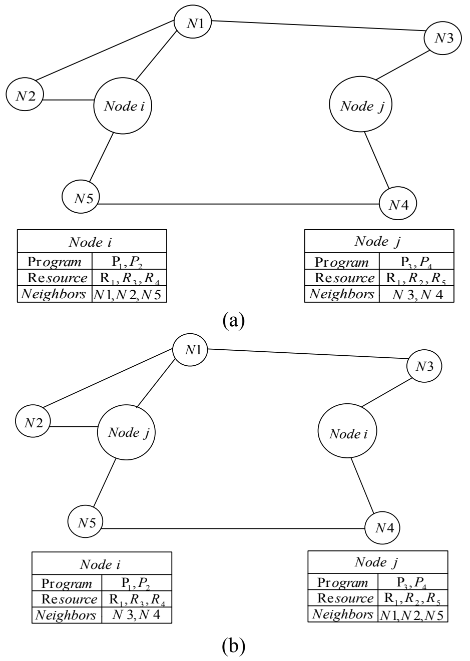
Node i	
Program	P_3, P_4
Resource	R_1, R_2, R_5
Neighbors	$N3, N4$

(b)

Figure 2. Neighbor topology: (a) Machines specifications in current topology, (b) Machines specification in neighbor topology.

B. Obtain Neighbor Topology by Exchanging Links

In this method, similarly to the previous method, two machines are selected from current topology of GCS. Neighbor topology is a topology in which resources and programs of machines remain immutable and all or some of connected links of two selected machines are exchanged. Also, it is controlled to be just one direct link between two machines and each machine has not any link to itself (loop). For example if *i, j* machines are selected for exchanging the data, the resulted neighbor topology is as shown in Fig. 3. The reason of obtaining neighbor topology in this manner is that GCS programs access to required resources through links in communication network [3][12]. Therefore, it is possible to decrease the amount of data exchanging through links that consequently can increase the reliability of the links and GCS.



Node i	
Program	P_1, P_2
Resource	R_1, R_3, R_4
Neighbors	$N1, N2, N5$

Node j	
Program	P_3, P_4
Resource	R_1, R_2, R_5
Neighbors	$N3, N4$

Node i	
Program	P_1, P_2
Resource	R_1, R_3, R_4
Neighbors	$N3, N4$

Node j	
Program	P_3, P_4
Resource	R_1, R_2, R_5
Neighbors	$N1, N2, N5$

Figure 3. Neighbor topology: (a) Communication network in current topology, (b) communication network in neighbor topology.

ISBN: 978-1-61208-101-4

In spite of original SA algorithms, which obtain a neighborhood structure as a fix structure [8][9][14], a different strategy is used in proposed method. In the proposed SA algorithm, in each step the number of programs and resources or links that must be exchanged are variable to obtain neighbor topology. It means that in the first step all resources and programs or links between two selected machines are exchanged and by approaching to the maximum number of SA iterations, in both A and B methods the number of exchanges in neighborhood structure is decreased such that in last iteration of algorithm the number of exchanges from current topology to neighbor topology is restricted to just one exchange. Therefore, with this strategy the speed of convergence to the global optimum topology is increased. In the proposed algorithm the percentage of exchanges in k-th iteration is calculated in (1).

$$(\mathit{Number\ of\ changes})\% = \frac{T_k}{T_0} \qquad (1)$$

where T_0, T_k are initial temperature and temperature of k-th iteration respectively. According to this equation the percentage of exchanges is decreased as k is increased.

V. ACCEPTANCE RULE OF NEIGHBOR TOPOLOGY

After obtaining a neighbor topology its reliability is calculated. If reliability of neighbor topology is better than current topology, it will be selected as the new current topology. But if this reliability is worse than reliability of current topology, it will be selected as new current topology according to a probability. This probability is calculated in (2).

$$P(Accept) = \begin{cases} 1 & \mathit{if} \quad \Delta \le 0 \\ e^{-\Delta / T_k} & \mathit{if} \quad \Delta > 0 \end{cases} \qquad (2)$$

where Δ is equal to the difference between reliability of current and neighbor topology and T_k is stand for temperature parameter in k-th iteration. Since the temperature parameter T_k in SA algorithm is decreased as iteration increases, so according to (2), decreasing T_k results in the reduction of acceptance probability.

Therefore, in this rule when the reliability of neighbor topology is worse than the current topology a number is selected between zero and one randomly that is named *r*. If *r* is more than acceptance probability, the neighbor topology is selected as new current topology.

VI. SCHEDULING RULE TO DECREASE THE TEMPERATURE PARAMETER AND INITIALING THE PARAMETER

In SA algorithm, scheduling the reduction of temperature parameter during algorithm iteration has significant effect on the way the algorithm operates. Scheduling rule for decreasing the temperature parameter *T* must be in a manner that reaches to zero before the last iteration of SA algorithm. In this paper, the applied method to decrease the temperature is based on geometrical progression, which is defined as (3).

$$T_{k+1} = \alpha T_k \qquad (3)$$

where α=0.95. Reduction of temperature parameter *T* as (3) is used in SA algorithm usually. In (3), α accepts different values. In our work, the different values for α are selected experimentally and shows this reality that $\alpha = 0.95$ results in better solution. In fact, α determines the speed of reduction for temperature parameter *T*.

```
$t = T_0$
$Initial\ topology = S_0$
$Best\ topogy = S_0$
$GR_0 = reliability\ (S_0)$
$GR_{Best\ topology} = GR_0$
$for\ i = 1\ to\ Max\ do$
  {   $NC = \frac{t}{T_0}$
        $S_1 = Generate\ neighbor(S_0, NC)$
      $GR_1 = reliabilty\ (S_1)$
      $If\ \ GR_1 \ge GR_0\ then$
         {   $S_0 = S_1$
            $GR_0 = GR_1$
            $If\ \ GR_1 \ge GR_{Best\ topology}\ \ then$
               { $Best\ topology = S_1$
                  $GR_{Best\ topology} = GR_1$   }
         }
$else$  {   $Accept = e^{-(\frac{GR_0 - GR_1}{t})}$
            $r = Random(0,1)$
            $if\ \ r > Accept\ \ then$
               {   $S_0 = S_1$
                  $GR_0 = GR_1$ }
         }
      $t = 0.95 * t$
  } $//for$
$Return\ (Best\ topology)$
```

Figure 4. SA pseudo-code for improve GCS reliability

TABLE I. Program Specifications

Program	Necessary Resources(Exchanged information (KB))
P1	R3(100),R4(300),R6(200),R7(400)
P2	R1(250),R2(200),R5(300)
P3	R2(100),R3(100),R4(200),R5(300),R7(300)
P4	R1(200),R2(100)

The parameters, iteration number (*max*) and initial temperature (T_0) are initialized by trial and error method. The value of *max* must be chosen in such manner that finds the global optimized topology with high probability before reaching to maximum number of iterations. Also, selection of parameter T_0 is in such a way that reaches to zero before reaching to the maximum number of iterations. Proposed algorithm for improving reliability of GCS has been shown as pseudo-code in Fig. 4 according to above explanations.

VII. EXPERIMENTAL RESULTS

In this section, the proposed algorithm is run to design a sample GCS with below characteristics and improve its reliability. To simulate the algorithm GridSim software is used. In this simulation initial GCS consisted of six machines, seven types of resources and four different programs. The failure rate and data transfer rate of each link of communication network are assumed to be 0.003 failures per second and 150 kbps respectively. The specifications of programs, which include the types of required resources and amount of data that must be exchanged to them, are assumed as shown in Table I. The corresponding graph of the initial GCS has been shown in Fig. 5. The specifications of machines in initial GCS that contain resources, programs and failure rate have been shown in Table II. The proposed algorithm was run with 0.5 and 100 for T_0 and *max* respectively. The GCS topology resulted from executing the proposed algorithm by using method B to obtain neighbor topology is shown in Fig. 6. (in this method specification of machines are immutable). Table III illustrates the specifications of obtained GCS topology using method A to obtain neighbor topology (in this method communication network is immutable).

TABLE II. Machines Specification of Initial GCS

Machine	Resource	Programs	Failure rate (Failure / sec)
M1	R3,R4	P1,P2	0.001
M2	R1,R2,R5	P1	0.002
M3	R2,R7	-	0.003
M4	R2,R4,R7	P4	0.004
M5	R3,R4	P2,P3	0.005
M6	R1,R2,R5	P3,P4	0.006

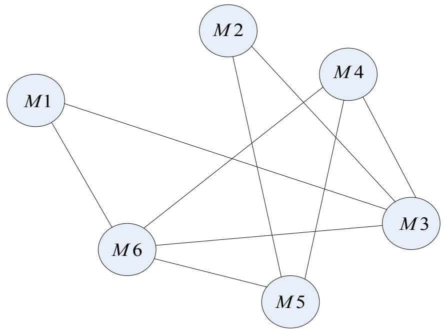


Figure 5. Initial GCS Communication Network

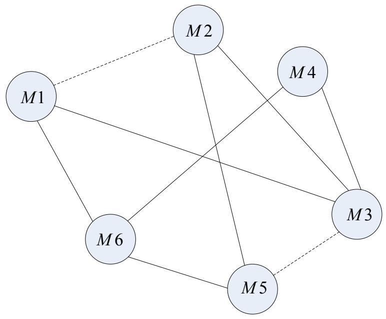


Figure 6. Resulted GCS topology by neighborhood structure method B

According to Fig. 6, in resulted GCS topology which uses method B to obtain neighbor topology, the communication links that are shown with dotted lines have been changed rather than original communication network in Fig. 5. According to Table III, in resulted GCS topology, which applies method A to obtain neighbor topology, the resources or programs that are shown with gray color have been changed rather than original specification of machines in Table II.

In this paper, the method proposed in [14] was used to estimate the reliability of GCS. Therefore, the reliability of initial GCS topology with specification of machines in Table II and communication network in Fig. 5 is 0.91324. The reliability of resulted GCS topology from executing proposed algorithm and methods A and B to obtain neighbor topology is 0.94347 and 0.96512. According to results, the reliability of obtained GCS by using proposed algorithm and neighborhood structures A or B is improved rather than initial GCS.

TABLE III. Machines Specification of Resulted GCS Topology by Neighborhood Structure Method A

Machine	Resource	Programs	Failure rate (Failure / sec)
M1	R3,R5	P1,P2	0.001
M2	R2,R5,R7	P1	0.002
M3	R2,R4	-	0.003
M4	R2,R5,R7	P3	0.004
M5	R2,R4	P2,P4	0.005
M6	R1,R2	P3,P4	0.006

TABLE IV. Results of SA by Different Max Iterations

Max-Iteration	1	2	3	4	5	Best	Average
1000	0.94061	0.93832	0.93154	0.94928	0.94867	0.94928	0.94166
10000	0.94402	0.96653	0.95436	0.97624	0.95931	0.97624	0.96009

It is clear from the obtained results that the proposed algorithm improves GCS reliability by using each one of neighborhood structure.

Since choosing initial values of parameters *max* iterations and T_0 is important in proposed algorithm; so to better evaluation of proposed algorithm, it was run with two different values of *max* iterations for five times. The results of reliability of obtained GCS topology are shown in Table IV. According to the results of Table IV, in each execution of proposed algorithm the resulted reliability of obtained GCS topology has been improved. Therefore, the percentage of reliability improvement for max=1000 is 2 to 4 percent and 3.5 to 7 percent for max= 10000, which shows this reality that changing the initial parameters can change the percentage of reliability improvement. Also it is clear from the obtained results that the proposed algorithm is convergent to a GCS topology with better reliability but improved percentage is different.

VIII. CONCLUSION AND FUTURE WORK

Since designing and obtaining the reliability of a GCS is an NP-Hard problem, so in this work SA has been applied to obtain a GCS topology and improve its reliability. The obtained results of running the proposed algorithm show that SA has this capability to be applied for designing a GCS topology and improving its reliability. It should be noted that, proposed algorithm can be improved efficiently by exploiting other techniques for reduction of parameter *T* or percentage of exchanges in obtained neighbor structure. This algorithm also can be used to obtain a GCS topology, which considers other factors like decreasing the congestion, rapid performing of programs, etc. Using combination of two neighbor structures together in proposed algorithm may improve proposed algorithm to represent better performance. This algorithm is used as a tool for initial designing of GCS to improve its reliability. In future work, we will focus on applying this method for real GCS in which the topology has been designed before. So, in order to improve the reliability and performance of GCS, the proposed method is used wherein resources are changeable such as databases or software.

REFERENCES

[1] I. Foster, C. Kesselman, and S. Tuecke, “The anatomy of the grid: Enabling scalable virtual organizations,” International Journal of High Performance Computing Applications, vol. 15, pp. 200-222, 2001.

[2] Y. S. Dai, M. Xie, and K. L. Poh, “Reliability analysis of grid computing systems,” Proc. IEEE Pacific Rim Int'l Symp. Dependable Computing (PRDC'02), pp. 97–104, 2002.

[3] Y. S. Dai, M. Xie, and K. L. Poh, “Markov renewal models for correlated software failures of multiple types,” IEEE Trans. Rel., vol. 54, no. 1, pp. 100–106, 2005.

[4] M. Daoud and Q. H. Mahmoud, “Monte Carlo simulation-based algorithms for estimating the reliability of mobile agent-based systems,” Journal of Network and Computer Applications, pp. 19–31, 2008.

[5] G. S. Fishman, “A comparison of four Monte Carlo method for estimating the probability of set conectedness,” IEEE Transactions on Reliabilty, vol. R-35, pp. 145-155, 1986.

[6] F. Altiparkmak, B. Dengiz, and A. E. Smith, “Reliability optimization of computer communication networks using genetic algorithms,” IEEE international conference on systems. San Diego, California, USA: Man and Cybernetics, pp. 4676–81, Oct. 1998.

[7] C. C. Hsieh and Y. C. Hsieh, “Reliability and cost optimization in distributed computing systems,” Computer Operation Research, vol. 30, pp. 1103–19, 2003.

[8] S. Anily and A. Federgrune, “Simulated annealing methods with general acceptance probabilities,” Journal of Applied Probability, vol. 24, pp. 657-667, 1987.

[9] D. Connoly, “General purpose simulated annealing,” European Journal of Operational Research, vol. 43, pp. 495-505, 1992.

[10] S. Kirkpatric, C. D. Gellat Jr, and M.P. Vecchi, “ Optimization by simulated annealing,” Science, vol. 220, pp. 671-681, 1987.

[11] A. Kumar, S. Rai, and D. P. Agarwal, “On computer communication network reliability under program execution constraints,” IEEE Journal of Selected Area in Communication, vol. 6, pp. 1393-1400, Oct. 1988.

[12] D. J. Chen and T. H. Huang, “Reliability analysis of distributed systems based on a fast reliability algorithm,” IEEE Transaction on Parallel and Distributed System, vol. 3, pp. 139-154, March 1992.

[13] B. Shirazi, M. Wang, and G. Pathak, “Analysis and Evaluation of Heuristic Methods for Static Task Scheduling,” Journal of Parallel and Distributed Computing, vol. 10, pp. 222-232, 1990.

[14] E. Gholami, A. M. Rahmani, and A. Habibizad Navin, “Using Monte Carlo Simulation in Grid Computing Systems for Reliability Estimation,” The Eight International Conference on Networks, Cancun, Mexico, pp. 380-384, March 2009.

[15] L. Ingber, “Simulated annealing: practice versus theory,” Mathl. Comput. Modelling, vol. 18, no. 11, pp. 29-57, 1993.

[16] H. R Anderson and J. P. McGeehan, “Optimizing microcell base station locations using simulated annealing techniques,” in 44th IEEE Vehicular Technology Conference, pp. 858–862, IEEE, 1994.

[17] S. Z. Selim and K. Alsultan, “A Simulated Annealing Algorithm for the Clustering Problem,” Pattern Recognition, vol. 24, no. 10, pp. 1003-1008, 1991.

[18] M. Gerla and L. Kleinrock, “On the topological design of Distributed Computer Networks,” IEEE Transactions on Communications, vol. 25, no. 1, pp. 55-67, 1977.

Optimal Selection of Sampling Rate in Multiple H_2 Control Loops

Antonio Sala
AI2 Institute
Technical University of Valencia
Valencia, Spain
asala@isa.upv.es

Carlos Ariño, Julio Romero, Roberto Sanchis
ESID Department
Universitat Jaume I,
Castellón de la Plana, Spain
arino,romeroj,rsanchis@esid.uji.es

***Abstract*—In this paper, the scheduling of the sampling frequencies of a set of independent controllers that share a limited resource is addressed. Bandwidth or CPU time limitations are assumed to be translated to constraints on the sum of the sampling frequencies. For the individual loops, H_2 sampled-data controllers are proposed, whose performance indexes can be calculated for the different sampling frequencies. A weighted sum of the individual loop performance conforms a global cost index. The problem is then posed as an optimization one, and some sensible simplifying alternatives are proposed, based on a grid of frequency points, that allow to solve it with Linear Programming (and hence with a low computing cost).**

***Keywords*-network control; optimal sampling frequency; network resource sharing**

I. Introduction

In digitally controlled systems, limitations on the frequency of the control computations are frequent. They may arise from multiple tasks running on the same processor so that a higher frequency of the control tasks would saturate the CPU load; they may also arise when a communication network between process and controllers is present and it has a limited bandwidth to be shared between multiple controllers, Programmable Logic Controllers and other information-processing elements. Apart from bandwidth limitations, increasing the network or computer load also gives rise to increased delays and sampling jitter, which might as well result in a performance loss in the tasks requiring the limited resource.

In most control literature, criteria for selection of sampling time do not usually consider the underlying resource limitation. Basically, the desired settling time, performance attenuation level, etc. result in a recommended sampling period, in most cases with practical “rules of thumb” (see [1] or [2]); it is left to the underlying real-time scheduler to achieve such a period with a reduced jitter, by dedicating to the task whichever computing/network resources are necessary.

The so-called co-design research line [3] tries to consider the design of both the control system and the communication and multi-tasking structures as a joint problem. Basically, the idea is combining restrictions on the sampling period arising from schedulability issues and bandwidth limitation (computation and transmission cost) and sampling-period dependent controller performance measures in order to solve a joint optimization whose results are the scheduler sampling periods and the controllers to be applied.

For instance Branicky et al. [3] proposed an optimality-based choice of sampling period for a multiple-loop control over a network based on a performance measure for each loop and some schedulability constraints. In [4] the objective was stability robustness, although first-order systems were only considered. Integral of Absolute Error as a function of sampling period was considered in [5]. Interestingly, Cervin et al. [6] discuss a generic approach in which the sampled-data cost of a controller is evaluated.

This paper roots on the last of the above cited works, formalizing the ideas to sampled-data output-feedback $\mathcal{H}_2$ control, and proposing a linear programming approximation on a finite grid of sampling frequency points.

The structure of the paper is as follows. Next section discusses some preliminary ideas and states the problem to be solved. Section III reviews sampled-data $\mathcal{H}_2$ control. Off-line (fixed rate) scheduling is discussed on Section IV. An example section and some conclusions are also provided.

II. Preliminaries and problem statement

Consider a network that is shared by several control loops (controllers, sensors and actuators).

The network resources used by each control loop and the achieved performance depend on the sampling frequency of that loop and the controller designed for it. Hence, each loop will be characterized by:

- Its sampling frequency f_i
- A controller scheduling policy $C(f_i)$
- A theoretical performance measure with that controller $J_i(f_i)$

Therefore, due to the overall bandwidth limitations, if the performance of one loop needs to be improved by increasing its sampling frequency, other loops must reduce their frequency and, hence, their performance. The problem to be discussed in this paper is how to apportion the limited resource between loops while trying to maximize the overall performance by minimizing a global cost index (composed

ISBN: 978-1-61208-101-4

of the weighted sum of individual indices), such as:

$$J(f_1, \ldots, f_r) = \sum_{i=1}^{r} h_i J_i(f_i)^p \tag{1}$$

where h_i are weights allowing the designer to emphasize the need of more accurate control of some processes. As commented in the introduction, this idea has been previously explored in literature. The result is an optimal sampling frequency distribution given the network constraints.

This paper has chosen the $\mathcal{H}_2$ performance measure as cost index (minimum-variance controller when subject to white-noise inputs). Indeed, if $\mathcal{H}_2$ sampled-data optimal controllers are used in the loops, the optimal performance of each loop can be calculated as a function of its sampling period via well-known sampled-data $\mathcal{H}_2$ formulae.

III. Sampled-data optimal control

The main issues in $\mathcal{H}_2$ sampled-data optimal control are reviewed next.

Consider a linear time-invariant continuous-time process given by:

$$\dot{x} = A_1^c x + B^c u + G_1^c v \tag{2}$$
$$\dot{\psi} = A_2^c \psi + G_2^c w \tag{3}$$
$$z = Cx + Du \tag{4}$$
$$y = C_2 x + C_3 \psi + D_2 u \tag{5}$$

so that it has a transfer function representation given by:

$$z = G_{11}(s)v + G_{13}(s)u \tag{6}$$
$$y = G_{21}(s)v + G_{22}(s)w + G_{23}(s)u \tag{7}$$

where z denotes the variables to be controlled, u are the manipulated inputs, y are the measurements and v, w are assumed to be white-noise disturbances to be denoted as process noise and measurement noise, respectively, with unit variance (all variance information is included in matrices G_1^c ad G_2^c). The state variables x are denoted as process state, whereas the state variables ψ are states of the measurement noise generator subsystem, assumed to evolve uninfluenced by x.

Given a sampling period T, a sampled-data controller will be designed so that its input will be the sequence of sampled outputs y_k and its output will be a sequence of control actions u_k to be fed to the continuous-time plant via a zero-order hold.

The control objective is to obtain the controller that minimizes the variance of z, $tr(E(zz^T))$ for a given, fixed, sampling period T. Such a problem is denoted in literature as the $\mathcal{H}_2$ sampled-data optimal control problem. It was shown in [7], [8] that such a problem can be cast as a pure discrete-time $\mathcal{H}_2$ optimal control problem for the discretized model given by:

$$x_{k+1} = A_1 x_k + B u_k + G_1 v_k \tag{8}$$
$$\psi_{k+1} = A_2 \psi_k + G_2 w_k \tag{9}$$
$$z_k = C_1 x_k + D_1 u_k \tag{10}$$
$$y_k = C_2 x_k + C_3 \psi_k + D_2 u_k \tag{11}$$

where the above discrete-time matrices are given by:

$$A_1 = e^{A_1^c T}, A_2 = e^{A_2^c T} \tag{12}$$
$$B = \int_0^T e^{As} B ds \tag{13}$$

and G_1, G_2, C_1 and D_1 are any matrices satisfying:

$$G_i G_i^T = \int_0^T e^{A_i^c s} G_i^c (G_i^c)^T e^{(A_i^c)^T s} ds \tag{14}$$
$$(C_1 D_1)^T (C_1 D_1) = \int_0^T e^{\underline{\mathrm{A}}^T s} (CD)^T (CD) e^{\underline{\mathrm{A}} s} ds \tag{15}$$

where:

$$\underline{\mathrm{A}} = \begin{pmatrix} A & B \\ 0 & 0 \end{pmatrix} \tag{16}$$

and all of the above matrices can be obtained from matrix exponential formulae without the need of actually carrying out any integration [9].

The obtained variance approaches that of the continuous-time $\mathcal{H}_2$ controller when the sampling period tends to zero [10]. Indeed, as a piecewise-constant control is a valid possibility for the optimal control action $u(t)$, the continuous-time solution of the above minimum-variance problem will be equal or better than any sampled-data optimal solution. The sampled-data system will have a closed-loop $\mathcal{H}_2$ norm given by that of the above discrete system plus a factor given by [10]:

$$\frac{1}{T} \int_0^T \int_0^{T-s} trace(C_2 e^{A_i^c \tau} G_i^c (G_i^c)^T e^{(A_i^c)^T \tau} C_2 d\tau ds \tag{17}$$

It is well known that the optimal controllers have the form of a Kalman filter observer plus a static state feedback, and that optimal control weights in classical linear quadratic regulator setups can be translated to the above $\mathcal{H}_2$ problems by a suitable choice of C and D.

If the measurement-noise dynamics G_{22} is sufficiently fast, the measurement noise will appear as a constant-variance stationary process when sampled at all except very small sampling periods; in that case, the dynamics of the states ψ can be eliminated in practice and $C_3\psi$ replaced by a discrete stochastic process with a constant variance equal to the stationary variance of the continuous one. This yields the classical “measurement noise variance” in discrete stochastic process models.

IV. Resource scheduling

Basically, the computing and network resources required by a control task will be proportional to the sampling frequency. Hence, the objective is achieving maximum performance (in terms to be later detailed) given a finite "bandwidth" bound, β, set up from computer and network load analysis.

The objective of this paper is proposing an scheduling methodology that allows an efficient use of the assigned control bandwidth by devoting more resources to processes that need a better control and, hence, must be operated at a higher sampling rate. The ideas in the previous section allow to easily obtain the optimal performance as a function of the sampling rate of a particular process. Considering now r independent control loops, which should share a computer or a network, we will denote as $\gamma_i(f)$ the optimal disturbance-rejection $\mathcal{H}_2$ performance obtained for process i by a controller operating at frequency f (i.e., with sampling period $T = 1/f$).

Taking into account all loops simultaneously, an overall performance measure may be defined as:

$$J(f_1, \ldots, f_r) = \sum_{i=1}^{r} h_i \gamma_i(f_i)^p \tag{18}$$

where h_i are weights allowing the designer to emphasize the need of more accurate control of some processes. The selection of these weights should depend on the disturbances acting on each loop, and on the economic cost derived from the resulting loop error. The higher the disturbance and the cost, the higher the weight.

The needed resources as a function of controller frequency may be expressed as:

$$R(f_1, \ldots, f_r) = \sum_{i=1}^{r} d_i f_i \tag{19}$$

for some given constants d_i to be determined based on processor load and number of bits transmitted by each control task (transmission time plus computation time). On the sequel, the vector of frequencies for each control loop will be denoted as F.

The goal of the bandwidth scheduling will be to obtain the sampling frequencies f_i for each of the control loops taking into account the performance and resource measures defined above.

Then, some scheduling problems of interest may be conceived:

- Given a resource constraint
$$R(F) \leq \beta \tag{20}$$
obtain the lowest $J(F)$.
- Given a performance objective J_0, obtain the lowest level of resources needed to achieve it: minimize $R(F)$ constrained to $J(F) < J_0$.
- variations of the above problems including some performance requirements for individual loops $\gamma_i(f_i) \leq J_{0,i}$ or multi-criteria settings (obtaining, for instance, a Pareto front on performance vs. available bandwidth).

A. Alternatives for the optimization problems

Depending on the shape of γ_i, the allowed values for the decision variables F, the value of the exponent parameter p in (18) and the chosen problem formulation from the options above, the required optimization algorithm will be different.

Several interesting options are discussed below:

1) Discrete optimization over a finite set of alternatives: Set up two or three performance levels for each process, say: high-frequency, normal-frequency, low-frequency sampling. Then, the problem gets transformed to a choice between a finite set of decision variables and it can be explored by brute force if the number of controlled loops is small.

2) Linear (approximate) optimization: Set up a dense enough grid of points f_j^*. Then, for each individual $\gamma_i(\cdot)^p$ function, compute the linear interpolation between the available frequency points, giving rise to a piecewise-linear $\gamma_i^*(\cdot)$ interpolation function. Subsequently, determine an interval of interest $[f_i^-, f_i^+]$ where individual performances $\gamma_i^*(\cdot)$ are convex functions. Doing this for all the controlled loops, the controller cost will then be the sum of univariate convex functions and, hence, a convex piecewise-affine function.

It is well known that piecewise-affine functions can be optimized via Linear Programming. Let us describe the basic idea below:

Denote as f_k^*, $k = 0, \ldots, \bar{k}$ the grid points in the above interval $[f_i^- = f_0^*, f_1^*, \ldots, f_i^+ = f_{\bar{k}}^*]$.

In that interval, $\gamma_i^*(f)$ may be approximately expressed as the piecewise-linear interpolation between grid points, to be denoted as $\bar{\gamma}_i^*(f) \approx \gamma_i^*(f)$. Conveniently, such linear interpolation can be rewritten as a linear-programming optimization:

$$\bar{\gamma}_i^*(f) = \gamma_i(f_i^-) + \min_{\epsilon_k} \sum_{k=0}^{\bar{k}} \alpha_k \epsilon_k \tag{21}$$

subject to the linear constraints

$$\epsilon_k \leq f_k^* - f_{k-1}^*, \quad \sum_{k=0}^{\bar{k}} \epsilon_k = f - f_i^- \tag{22}$$

and where α_k are the piecewise slopes, defined as

$$\alpha_k = \frac{\gamma_i(f_{k+1}^*) - \gamma_i(f_k^*)}{f_{k+1}^* - f_k^*}$$

that fulfils the condition $\alpha_{k+1} \geq \alpha_k$ due to the assumed convexity of $\gamma_i^*(\cdot)$ in the given interval.

Carrying out a similar derivation for each of the loop performances (choosing a gridding with $\bar{k}_i$ intervals for each

i), the overall cost can be expressed as:

$$J(F) = \min_{\epsilon_{i,k}} \sum_{i=1}^{r} h_i \left(\gamma_i(f_i^-) + \sum_{k=0}^{\bar{k}_i} \alpha_{i,k}\epsilon_{i,k} \right) \qquad (23)$$

subject to $\epsilon_{i,k} \leq f_{i,k}^* - f_{i,k-1}^*$, $\sum_{k=0}^{\bar{k}_i} \epsilon_{i,k} = f_i - f_i^-$. Hence, the optimization problem to be solved consists of the above problem constrained to the additional condition $\sum d_i f_i \leq \beta$. Such problem is a linear programming one that can be efficiently solved.

Remark: An interesting particular case results when the weights $d_i = 1$, and the grid points f_j^* are uniformly spaced so the distance between the grid points is an exact divisor of the bound β. In that case, as LP optimal solutions always lie at slope changes or constraint bounds, the result of the LP optimization will always lie at one of the grid points, i.e., the optimum frequencies always belong to a predefined set. This would be especially useful for an on-line scheduling, where the switching between a finite set of controllers could be a simple solution. This issue will be studied in future works.

3) Generic nonlinear optimization.: If a non-linear (polynomial, spline, etc.) interpolation were chosen to approximate $\gamma_i(\cdot)$, the scheduling problem would be a nonlinear optimization problem. That would also be the case if the intervals $[f_i^-, f_i^+]$ were too small for the particular application. If the number of simultaneous loops were small, a subdivision of the interpolation table in a finite number piecewise convex (or concave) regions would allow for solving a linear programming problem for each of such regions and computing the global minimum as the minimum of the local optimizers (details omitted for brevity).

V. Examples

Considerer the simple case of controlling two identical SISO systems whose disturbance inputs might, however, be not identical. Under limited resources, the effort should concentrate on the process subject to larger disturbances, which must be known *a priori* and cast into the optimization index in the off-line scheduling case.

Each of the systems can be represented by the state space model (24) where x are the state variables, y is the output, v are the white noise variables and z represents the weighted controlled variables.

$$\begin{aligned} \dot{x} &= Ax + Bu + Gv \\ y &= Cx \\ z &= C_z x + D_z u \end{aligned} \qquad (24)$$

where

$$A = \begin{pmatrix} -20 & -12.5 \\ 8 & 0 \end{pmatrix}, \; B = \begin{pmatrix} 16 \\ 0 \end{pmatrix} \qquad (25)$$

$$G = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad (26)$$

$$C = \begin{pmatrix} 0 & 15.625 \end{pmatrix}, \; C_z = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \; D_z = \begin{pmatrix} 0 \\ 0 \\ 0.1 \end{pmatrix} \qquad (27)$$

In order to select the discrete dynamic controllers for both systems, we obtain the sampled models at different frequencies f following the procedure commented in Section II. Once converted we obtain for each frequency the $\mathcal{H}_2$ controller and the minimum $\mathcal{H}_2$-gain bound for disturbance rejection. These values are presented in Figure 1. As it can be seen at low frequencies the closed-loop $\mathcal{H}_2$ norm approaches the open loop value (5.867), as it was expected, and at high frequencies $f \simeq 1KHz$ it approaches the norm of the continuous-time $\mathcal{H}_2$ optimal controller, whose value is 1.576.

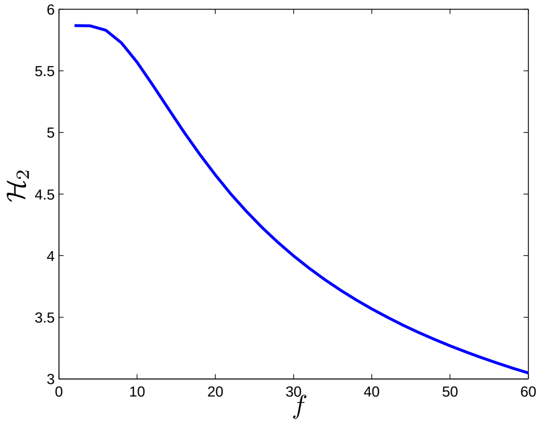


Figure 1. Minimum $\mathcal{H}_2$ rejection at frequency f

The selection of the frequencies for each controller (f_1 and f_2) will be done taking into account the resource constraint (28) and the performance index (29).

$$f_1 + f_2 \leq K \qquad (28)$$

$$J = h_1\gamma_1^2(f_1) + h_2\gamma_2^2(f_2) \qquad (29)$$

where the chosen values have been $h_1 = 4$, $h_2 = 1$ indicating that the expected value of the disturbance inputs for process 1 is double than that for process 2. The frequency limit has been set to $K = 60$.

As it can be seen in Figure 1, the cost function J will not be convex at low frequency values, but it will be convex for frequencies larger or equal than 10 Hz. However, the part of the plot on the left of the inflection point corresponds to almost open-loop behaviour for very low sampling rates, which are not relevant for the tested cases. They would only be relevant for a setup in which one of the disturbances is extremely larger than the other one, almost requiring controlling one of the processes at the maximum frequency

ISBN: 978-1-61208-101-4

and the other one in open-loop. This has not been the case for the simulations in this paper's examples.

In order to solve a convex problem we approximate the cost function J following the methodology in Section IV-A2 at the convex range of J as

$$\gamma_1^{*2} \quad = \quad \gamma_1^2(f_1^-) + \min_{\epsilon_{1,k}} \sum_k \alpha_k \epsilon_{1,k} \tag{30}$$

$$\gamma_2^{*2} \quad = \quad \gamma_1^2(f_2^-) + \min_{\epsilon_{2,k}} \sum_k \beta_k \epsilon_{2,k} \tag{31}$$

Note that, in this case $\alpha_k = \beta_k$ because the systems have the same dynamic model. The candidate sampling frequencies vectors F_1 and F_2 are taken also identical for both systems, uniformly distributed from 12 to $60Hz$, computing approximation points every 2 Hz (i.e., $f_1^- = f_2^- = 12$, $f_1^+ = f_2^+ = 60$, $f_{i,k}^* = 12 + 2k$). Then the optimization problem can be approached by the linear programming problem procedure presented in the referred section.

As a result, we obtain the optimal $\mathcal{H}_2$ norm bound at frequencies $f_1 = 42$ and $f_2 = 18$. The controllers' state space gains (K_i) and Kalman filter gains (L_i) that minimize J^* are presented in (32) and (33) below, respectively.

$$K_1 = (-4.228 \ \ -25.16), \quad L_1 = (-0.014 \ 0.0622)^T \tag{32}$$

with the sampling time $T_1 = 1/42$.

$$K_2 = (-1.839 \ \ -6.4171), \quad L_2 = (-0.0275 \ 0.0548)^T \tag{33}$$

with the sampling time $T_2 = 1/18$.

A contour plot of the cost function $J(f_1, f_2)$ is represented at Figure 2 with its constraints.

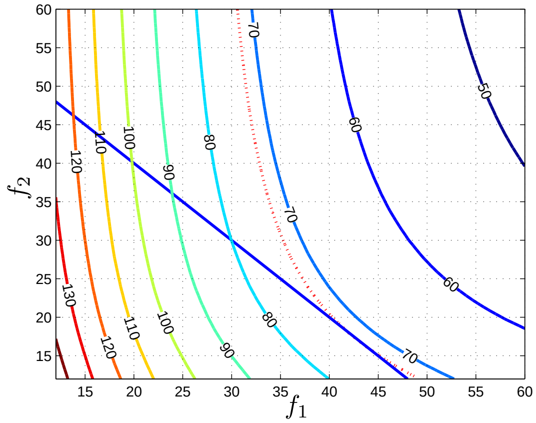


Figure 2. Contour plot of the cost function J and the constraint $f_1 + f_2 \leq 60$.

VI. Conclusion and Future Work

This paper has presented an optimal scheduling of a set of independent $\mathcal{H}_2$ sampled-data controllers operating on a shared resource, which gives rise to sampling frequency constraints. The problem has been posed as an optimization one and some sensible simplifying alternatives have been proposed, based on a grid of frequency points, that allow to solve it with Linear Programming, and hence, with a low computing cost. Some examples have illustrated the approach. As a future work, the online scheduling of the sampling rates will be studied. The idea will be to adapt the scheduling to changes in the available resources or in the process disturbances. The simplified Linear Programming based optimization presented in this paper will be a key point in that work.

Acknowledgment

The authors are grateful to the financial support of Spanish Ministry of Education grants DPI2008-06731-C02-01, DPI2008-06731-C02-02 and Generalitat Valenciana grant PROMETEO/2008/088.

References

[1] K. Astrom and B. Wittenmark, *Computer-controlled systems: theory and design*. Prentice Hall New York, 1996.

[2] P. Albertos, A. Sala, and M. Chadli, "Multivariable control systems—an engineering approach," *Automatica*, vol. 41, no. 9, pp. 1665–1666, 2005.

[3] M. Branicky, S. Phillips, and W. Zhang, "Scheduling and Feedback Co-Design for Networked Control Systems (I)," in *IEEE Conference on Decision and Control*, vol. 2. Citeseer, 2002, pp. 1211–1217.

[4] L. Palopoli, C. Pinello, A. Vincentelli, L. Elghaoui, and A. Bicchi, "Synthesis of robust control systems under resource constraints," *Lecture Notes in Computer Science*, pp. 337–350, 2002.

[5] C. Peng, D. Yue, Z. Gu, and F. Xia, "Sampling period scheduling of networked control systems with multiple-control loops," *Mathematics and Computers in Simulation*, vol. 79, no. 5, pp. 1502–1511, 2009.

[6] A. Cervin, M. Velasco, P. Marti, and A. Camacho, "Optimal on-line sampling period assignment," Automatic Control Department, Technical University of Catalonia, Tech. Rep. ESAII-RR-09-04, December 2009, available as http://paginespersonals.upcnet.es/~pmc16/0910RR04.pdf.

[7] B. Bamieh and J. Pearson Jr, "A general framework for linear periodic systems with applications to H∞ sampled-data control," *IEEE Transactions on Automatic Control*, vol. 37, no. 4, pp. 418–435, 1992.

[8] P. Khargonekar and N. Sivashankar, "H2 optimal control for sampled-data systems," *Systems & Control Letters*, vol. 17, no. 6, pp. 425–436, 1991.

[9] V. Loan, "Computing integrals involving the matrix exponential," *IEEE Transactions on Automatic Control*, vol. AC-23, no. 3, pp. 395–404, 1978.

[10] H. Trentelman and A. Stoorvogel, "Sampled-data and discrete-time H-2 optimal control," *SIAM Journal on Control and Optimization*, vol. 33, no. 3, pp. 834–862, 1995.

ISBN: 978-1-61208-101-4

Radial Basis Function and Elman Networks for Pollutant's Parameter Prediction in the Region of Annaba Algeria

Mohamed Tarek Khadir
University of Badji Mokhtar, Dep. of Computer Science
Laboratoire de Gestion Electronique de Documents
(LabGED)
Annaba, Algeria
khadir@labged.net

Sabri Ghazi
University of Badji Mokhtar, Dep. of Computer Science
Laboratoire de Gestion Electronique de Documents
(LabGED)
Annaba, Algeria
ghazi@labged.net

Abstract— **This paper describes the development of air pollutants concentration prediction models of five different pollutants (O3, PM10, SO2, NOx, COx), using Radial Basis Function, and Elman Networks, two neurocomputing paradigms. Each Artificial Neural network (ANN) predicts, therefore the concentration of the five different pollutants. These models are developed in order to give 12 hours ahead prediction for the region of Annaba, northeast of Algeria (north of Africa). Receiving the measurement of air pollutant concentration and the metrological parameters (wind speed, temperature and humidity) at time t; the models are designed to predict air pollutant concentration at t+12 hours. Once predicted pollutant concentrations are obtained, and the validity of each ANN model is proven, the performances of both ANN models are comprehensively compared and assessed. Conclusions are finally drawn and the use of a particular ANN network over another is justified on the light of the obtained results.**

Keywords- ***Pollutants concentration prediction, Artificial Neural Network, Elman Network, Radial Basis Function, neurocomputing.***

I. Introduction

All air pollution generated by rapid urbanization, population growth and industrialization has taken alarming dimensions, and is one the greatest evil that mankind has to face in the coming years. To prevent the further decline of air pollution, scientific planning of analysis methods and pollution control are required. Within this framework it is necessary to implement air quality forecasting tools in order to take needed measures, such as reducing the effect of a predicted pollution peak on the surrounding population and eco-system. Many factors influence air pollutants concentrations. Among the most important are: metrological conditions, topology and population density. This makes air pollution difficult to model. Many air pollution prediction models have been studied [1] such as, mathematical emission models, linear models; Artificial Neural Networks-based models and hybrid models, in order to design air quality prediction systems, moderates air pollution and limit the influence of peaks periods by informing community by taking the necessary precaution. Air pollutant can be chemical such as: (SO_2: dioxide of sulfur, O_3: ozone, NO_x: oxide of nitrogen, CO_x: oxide of Carbon), or solid such as PM10 (Particulate Matter with an aerodynamic diameter of 10 micrometer). PM10 are the sum of all solid and liquid particles suspended in air, many of which are hazardous. This complex mixture includes both organic and inorganic particles, such as dust, pollen, soot, smoke, and liquid droplets. These particles vary greatly in size, composition, and origin. Particles in air are either: directly emitted, for instance when fuel is burnt and when dust is carried by wind, or indirectly formed, by a photochemical interaction between gaseous pollutants in the air space.

The paper is organized as follows. Section II introduces the specifications of the studied area and the nature and amount of used data. Section III explains the usage of Artificial Neural Networks as pollutants prediction tool. Section IV details the design of the two ANN approaches, i.e., RBF and Elman. In the light of the available data, chosen networks topologies are justified. Section V presents all obtained results and establishes a comparison study. Finally, Section VI concludes the research paper.

II. Studied area and availlable data

A. *Studied Area*

In Algeria, respiratory infections remain the leading cause of child mortality after measles and diarrhea [2]. Chronic bronchitis, lung cancer and asthma, among other diseases are also caused by pollution. This is especially apparent in some unventilated and highly industrialised regions; Annaba may be counted among these. In the region, the prevalence of asthma is higher than the national rate, 55% of asthmatics have more than one crisis per month and 42% of patients were hospitalized at least once during the year. Annaba region is located in the Eastern part of Algerian coast (600 km of Algiers); its town is constituted

ISBN: 978-1-61208-101-4

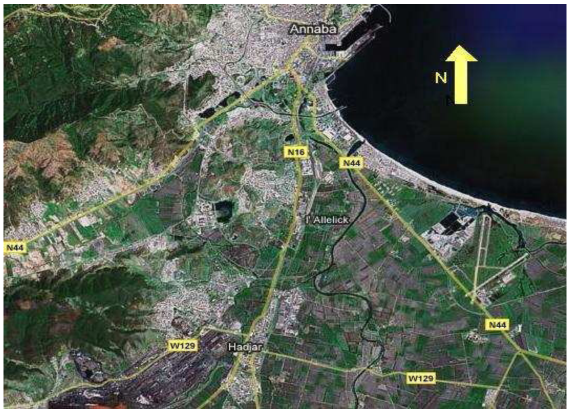


Fig. 1 Geographic map of Annaba, northeast of Algeria (Google Earth ©).

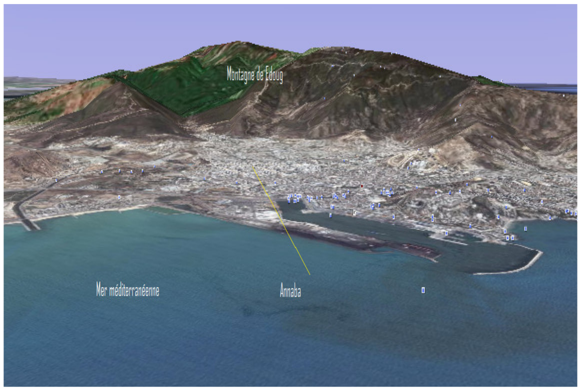


Fig.2: Topology of Annaba (Google Earth ©)

of a vast plain bordered in the South and West, of a mountainous massive in North, and by the Mediterranean Sea in the East. Its bowl-shaped topography favors air stagnation and the formation of temperature inversions. These situations allow the pollutants accumulation and the increase in concentration that result. Sea and land breezes contribute to slope transport of clouds of pollutants. Indeed, the clouds of pollutants are carried away by land breeze to the sea. These clouds of pollutants return to the city as a result of sea breezes along the Seraidi Mountain. The clouds look over the city in a form of circle. The pollutants are deposited slowly by gravity and therefore there is pollution affecting the three receivers (sea, land and air) [2]. Contaminants Air emissions are distributed differently depending on industry. The industry is the engine of growth and environmental degradation in Annaba and its surrounding area, where the most industrial sites are located nearby, such as the complex of phosphate and nitrogen fertilizers, Asmidal and the Metalsteel complex of El-Hadjar. These industrial activities are the main source of particulate matter and sulfur oxides, while emissions of carbon monoxide, nitrogen and lead are mainly due to the transport sector. Emissions from road traffic are the main pollutants of the atmosphere. Air pollution has increased with the increasing number of vehicles (an annual increase of 5% in Algeria) and the absence of emissions control. Meanwhile, the treatment of waste (domestic, industrial, hospital, toxic), which is to put them in wild sites, is also a source of air pollution to such an extent, that they are incinerated in free air.

B. Data Pre-processing

The dataset used in this work covers the period 2003-2004 and was provided by SAMASAFIA network center [3] on a continuous basis of 24 hours. Air pollutants monitored continuously include the concentrations of: nitric oxide (NO), carbon monoxide (CO), ozone (O3), particulate matter (PM10), nitrogen oxides (NOx), nitrogen dioxide (NO2) and sulfur dioxide (SO2). The used dataset includes also three meteorological parameters: Wind Speed (WS), Temperature (T) and relative humidity (H).

The available data is an hourly measurements of pollutants concentration of two years, form 1/1/2003 to 31/12/2003 and form 1/1/2004 to 31/12/2004, in Annaba. The 2003 dataset was used for training and the one of 2004 for validation; this will help us to efficiently asses and then get the performance of the model. The pollutant concentration measurements are in microgram/ m3. To adapt the data with our model, we have applied normalization using equation (1). Negative values, resulting from faulty measurements where replaced using (2), this will make the input ranges between [0 1], and in the output we will obtain a percentage indicating the degree of the peak alarm.

$$Vp = \frac{Vp}{(max(Vp) - min(Vp))} \tag{1}$$

where V_p is a vector of parameter, *min* and *max* are function that returns the maximum and minimum of the vector.

$$T = T + (T_{max} - T_{\min}) \tag{2}$$

where *T* is vector of temperatures, *min* and *max* being the maximum and the minimum values of temperature.

III. Artificial Neural networks for air pollutant's concentration

Artificial Neural Networks (ANNs) are an alternative to classical statistics methods used for prediction in air quality monitoring stations [1]. Large amount of available data from monitoring stations can be very useful in order to design efficient ANN predictions models. Relationship between meteorological parameter and air pollutant concentration is very difficult to model. Mlakar and Boznar [4] present a Multi Layered Perceptron (MLP) based model for SO_2 concentration prediction, it receives the meteorological and emission parameters as inputs, the model shows an efficient predictions and outperform the ARMA (Autoregressive Moving Average) based model. Mlakar and Joseph [5] present a pattern recognition inspired approach to select the parameter for the air pollution prediction model, in order to optimize the training time and increases model performance.

To get the most adaptable model for air pollution prediction Jorquera *et al.* [6] present a comparison between three models: ANN based model, linear model and Fuzzy logic based model. These models have been applied to predict Ozone concentration in Santiago Chile, the Fuzzy and ANN model has shown an efficient prediction and outperform the linear model. Gardner and Droling [7] present NO_2 prediction model based on MLP topology, compared with linear regressor using the same input data and parameter, the MLP based model shows better accuracy. In Gardner and Droling [8], the long-term tendency of Ozone concentration in London has been studied using ANN based model. The prediction of PM2.5 (Particulate matter with diagonal smaller the 2.5 micrometer) concentration has been well studied in Perez *et al.* [9], using ANN based model which is able to predict short-term concentration of PM2.5. Perez and Reyes [10] focused on the prediction of PM10.

This is to cite but a few, other research work may be found in the literature such as: Foxall *and al.* [11] for ozone, Corani [12] for Ozone and PM10, Bianchini *et al.* [13] for NO_2, etc.

IV. MODEL DEVELOPMENT

A. Radial Basis Function (RBF)

Radial Basis Function networks has been proposed and used in many studies [14][15], a detailed presentation can be found in Chen *et al.* [16]. As illustrated in Figure 3, RBF networks consist in three layers. The first layer is composed by *n* input neurons connected to each input. The input neurons (or processing before the input layer) standardizes the range of the values by subtracting the median and dividing by the interquartile range. The input neurons then feed the values to each of the neurons in the hidden layer. The neurons of the hidden layer have the radial basis function (3) as activation function,

$$f(x) = e^{(-x-M)^2/2\delta^2} \qquad (3)$$

where: M and δ are two parameters respectively the input variable mean and standard deviation.

The outputs of the nodes are combined linearly to give the final network output.

The main advantages of radial basis network may be: The time taken in designing a radial basis network is often less when compared to the training a sigmoid/linear networks; and the number of neurons required for designing the network is considerably less when compared to standard back propagation network [17].

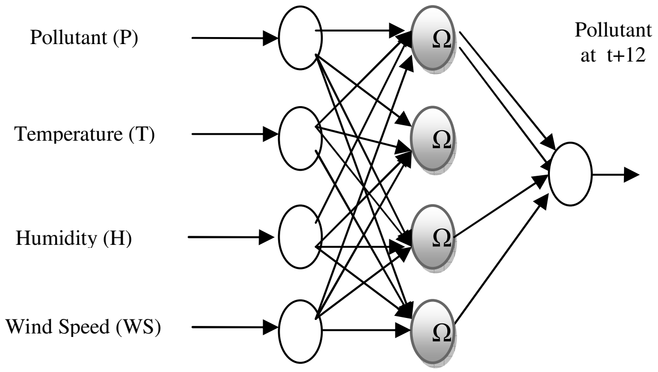


Figure 3: RBF network architecture.

The training algorithm, used iteratively, creates a blank RBF network (no neurons at first), and sequentially adding nodes by including one neuron at a time. Neurons are added to the network until the Sum of Squared Error (SSE) reaches a set target or the maximum numbers of neurons are reached. At each iteration the input vector, resulting in lowering the network error, is used to create a radial basis neuron. The distance between the connecting weights determines the output of hidden neurons and input vector, which is further, multiplied by bias, an additional scalar quantity being added between neuron and fictitious neuron. The output is propagated in a feed forward direction to output layer neurons.

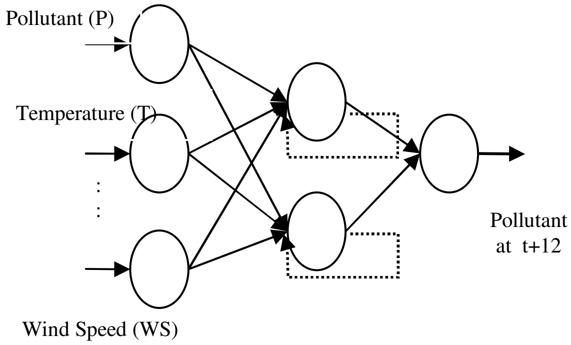


Figure 4: Elman's network architecture.

These, will give a response if the connection weights are close to input signal. This output is, then compared to a target vector. If the error reaches the error goal (a set value given the desired difference between the actual and modelled output), then training is terminated, otherwise the next neuron will be added. The complete training procedure is implemented in the function *newrb* available in the neural network toolbox supported by MATLAB.

B. *Elman networks*

Elman networks [18] are considered as recurrent neural networks, because they have feedback connections in the neurons of the hidden layer as illustrated in Figure 4. Elman networks are called buffered networks, as they take in consideration the previous output value and use it to calculate the next step output. The network leaves a trace of its behavior and keeps a memory of its previous states which may enhance the accuracy of the predictions.

V. RESULTS AND DISCUSSION

A. *Measurement indeces*

The performance of both modeling ANN approaches are evaluated in terms of Index of Agreement (IA) and the Mean Squared Error (MSE). The formulation of IA and MSE are given in equation (4) and (5) respectively.

$$IA = 1 - \frac{\overline{(Cp - C0)^2}}{\overline{\left|Cp - \overline{C0}\right|^2 + \left|C0 - \overline{C0}\right|^2}} \qquad (4)$$

$$MSE = \sum_{t}^{n} \frac{(C_{pt} - C_{0t})^2}{n} \qquad (5)$$

where C_p is the predicted value and C_0 is the measured value of pollutant concentration.

B. *Obtained results*

Table 1 presents the performance and the architecture of the RBF based models, the topology of the final selected networks as well as the performance of the models in terms of Index of Agreement (IA) and the Mean Squared Error (MSE) as performance measures.

TABLE 1. PERFORMANCE OF THE RBF MODELS

Pollutant	*Topology*	*IA*		*MSE*	
		Training	*Validation*	*Training*	*Validation*
PM10	10-320-1	0.9999	0.9987	0.0004	0.5508
O_3	10-180-1	0.9999	0.6922	0.0229	1.2571
NO_X	10- 105-1	0.9966	0.4866	0.1358	2.7515
SO_2	10-90-1	0.9976	0.5930	0.0995	2.9806
CO_X	10-45-1	0.9900	0.9716	0.1090	1.0513

It can be seen that RBFs show good performances for PM10 and Cox. For O_3, it gives medium accuracy, but for NO_x, it has the worst performance.

The training mechanism for Elman networks is similar to the one used for MLPs and is based on the back propagation algorithm. The overtraining phenomena affect EN in the same manner than it affect MLPs [19]. Table 2 shows the obtained results for all five pollutants.

TABLE 2: THE PERFORMANCE OF THE TESTED ELMAN NETWORKS

Pollutant	*Topology*	*IA*		*MSE*	
		Training	*Validation*	*Training*	*Validation*
SO_2	10-14-1	0.93362	0.92574	0.012658	1.9379
PM10	10-11-1	0.9947	0.89638	0.75991	4.3207
CO_2	10-10-1	0.99861	0.88857	0.039862	1.7877
NO_2	10-11-1	0.9927	0.94002	.88212	2.9005
O_3	10-13-1	0.90107	0.74896	0.37845	0.27922

Figures 5 shows the model performances over a range of 12h for both ANN paradigms (RBF and Elman's Networks) for PM10 predictions. Similarly, Figure 6 shows the results for O_3, Figure 7 for NO_X, Figure 8 for CO_X and Figure 9 for SO_2.

According to these figures and the summarizing Tables 1 and 2, RBF models seem the best adapted to predict efficiently pollutant concentration, regardless of their easier training and validation procedure. Elman models shows good generalization and predicts well pollutants concentration at the exception of SO_2, where this type of models shows the worst performances due to the existence of several peaks of this pollutant within the used data.

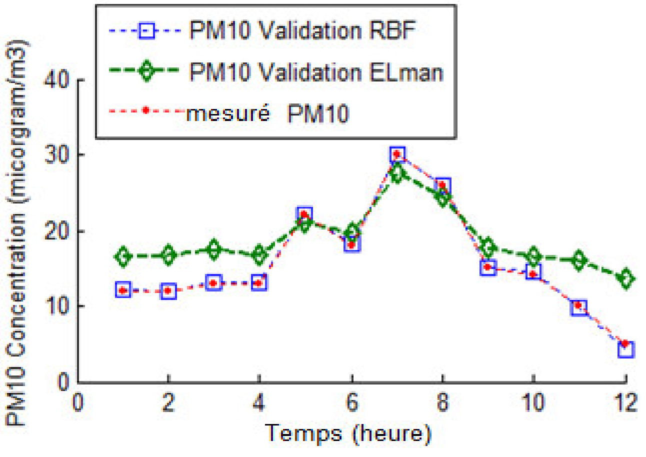


Figure 5 : Predicted and measured values for PM10

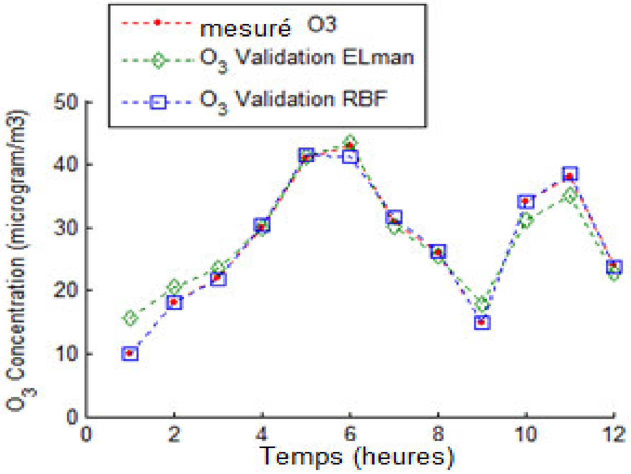


Figure 6 : Predicted and measured values for O_3

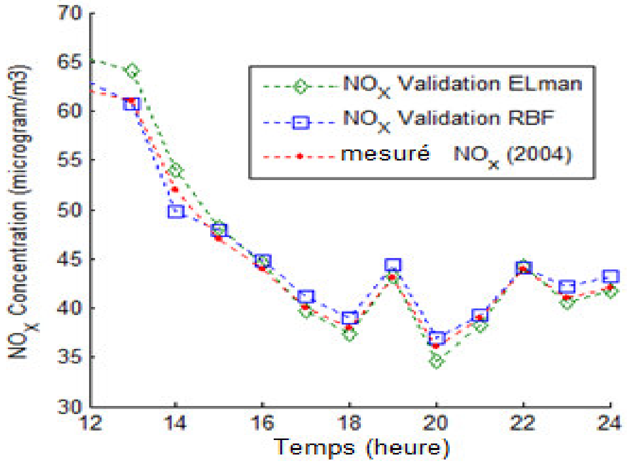


Figure 7 : Predicted and measured values for NO_x

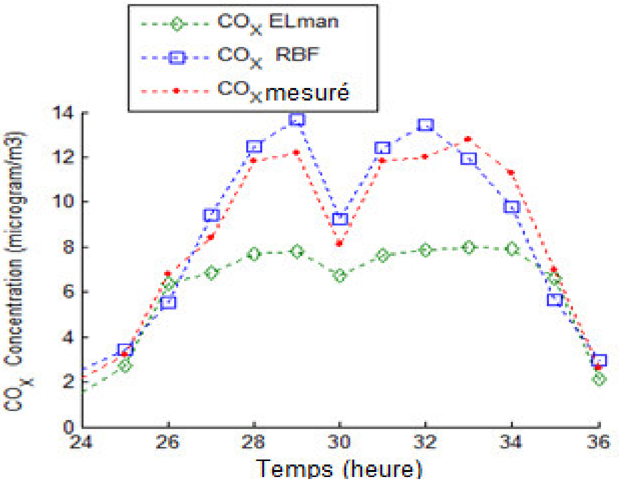


Figure 8 : Predicted and measured values for CO_x

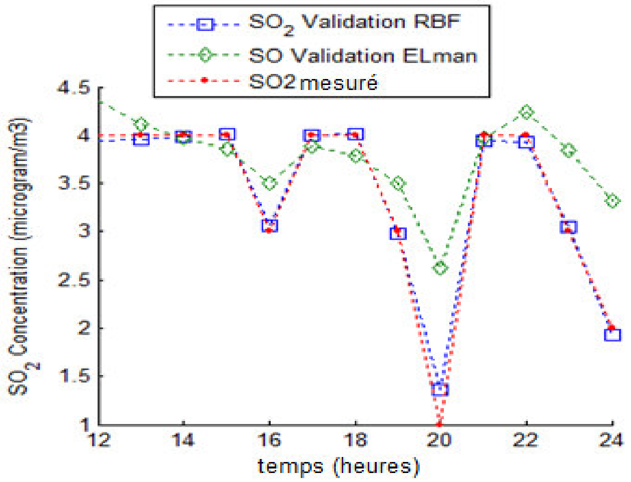


Figure 9 : Predicted and measured values for SO_2

VI. Conclusion

Two types of ANN models have been designed, validated and tested for the prediction of five air pollutants. RBF models have shown the best performances followed closely by Elman Network for single step prediction. However these types of models required longer training time and were computationaly effort consumming, even if no much design effort were taken for the choice of the topology number of neurons and feed back loops as in Elman networks..

Elman's networks overestimate or under estimate some pollutants peaks. This is seen for PM10 and CO_x. The addition of emission parameters related to cars and industries may improve prediction quality.

Finally, these results are specific to the region of Annaba. The use these models for others region must comply with data specific proprieties and training must be performed in order to identify the vest network topology.

References

[1] Mlakar P. and Boznar M. (2007), Air Pollution Modeling and Its Application XIV : Artificial Neural Network-Based Environmental Models, Springer, USA.

[2] Mebirouk H. And Mebirouk-Bendir F. (2007). Principaux acteurs de la pollution dans l'agglomération d'Annaba. Effets et développements, Colloque International sur l'Eau et l'Environnement, Alger.

[3] www.samasafia.dz\journaux (access 10/10/2009).

[4] Mlakar P. and Boznar M. (1994). Short-term air pollution prediction on the basis of artificial neural network. 2nd International Conference on Air pollution, Barcelona, Spain, pages 545-552.

[5] Mlakar P. and Josef S. (1997) Determination of Features for Air Pollution Forecasting Models, Proceedings of 3rd ed. International Conference on Intelligent Information Systems (IIS '97), Grand Bahama Island, Bahamas, pages 345-356.

[6] Jorquera H., Pérez R., Cipriano, A., Espejo, A., Letelier M.V., and Acuna, G. (1998). Forecasting ozone daily maximum levels at Santiago, Chile, Atmospheric Environment, Vol 32(20), pages 3415-3424.

[7] Gardner, M.W. and Dorling S.R. (1999). Neural network modeling and prediction of hourly NOx and NO2 concentration in urban in London, Atmospheric Environment, vol. 33, pages 709-719.

[8] Gardner M. and Dorling S. (1999). Meteorologically adjusted trends in UK daily maximum ozone concentrations, Atmospheric Environment, Vol. 34(2), pages 171-176.

[9] Pérez P., Trier, A., and Reyes, J. (2000). Prediction of PM2.5 concentrations several hours in advance using neural networks in Santiago, Chile. Atmospheric Environment, Vol. 34, pages 1189-1196.

[10] Perez P. and Reyes J. (2002). Prediction of maximum of 24-h average of PM10 concentrations 30 h in advance in Santiago, Chile, Atmospheric Environment Vol. 36, pages 4555-4561.

[11] Foxall R., Krcmar I., Cawley G., Dorling, S., and Mandic, D. (2001). Nonlinear modeling of air pollution times series, IEEE International Conference On Acoustics, Speech, And Signal Processing, Salt Lake City, Utah, USA, vol.6 Pages 3505– 3508.

[12] Corani G. (2005). Air quality prediction in Milan: feed-forward neural networks, pruned neural networks and lazy learning, Ecological Modeling, Vol. 185, Issues 2-4, Pages 513-529.

[13] Bianchini M., Di Iorio F., Maggini M., Mocenni C., and Pucci A. (2006). A Cyclostationary Neural Network Model for the Prediction of the NO2 Concentration, ESANN'2006 proceedings – European Symposium on Artificial Neural Networks, Bruges, Belgium,.

[14] Chen, S., Cowan, C.F., and Grant, P. M. (1991) Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks, IEEE Transactions on Neural Networks, Vol. 2(2), pages 302-309.

[15] Govindarajan L. and Sabarathinam P.L. (2006). Prediction of Vapor liquidEquilibrium Databy Using Radial Basis Neural Networks, Chem. Biochem. Eng. Q. Vol. 20 (3), pages 319–323.

[16] Cybenko, G. (1989). Approximation by superposition of sigmoidal function, Math. Cont. Sig. syst., vol. 2, pages 303-314.

[17] Costa, M., Pasero, E., Piglione F., and Radasanu D. (1999). Short term load forecasting using a synchronously operated recurrent neural network, Proceedings of the International Joint Conference on Neural Networks, vol.5, pages 3478-3482, Washington, DC, USA.

[18] Elman J. (1990). Finding structure in time. Cognitive Science vol 14, pages 179-211.

[19] Sjoberg, J. and Ljung, L. (1992). Overtraining, regularisation, and searching forminimum in neural networks. In 10th IFAC Symposium on System Identification, Copenhagen, Vol 2, pages 49-72.

Advances in Generalization and Decoupling of Software Parts in a Scientific Simulation Workflow System

Arne Bachmann, Markus Kunde, Markus Litz, Andreas Schreiber
German Aerospace Center
Simulation and Software Technology
Cologne, Germany
{Arne.Bachmann, Markus.Kunde, Markus.Litz, Andreas.Schreiber}@dlr.de

***Abstract*— Scientific simulation workflows today consist of a pool of simulation models of different domains that are linked together. In the past this was often done with highly specific connections between the simulation models, e. g., batch-scripts or use of commercial integrated systems prescribing certain procedures. This strong coupling led to several problems like the non-reusability of a simulation model in other contexts or other software environments. To address this situation a concept called *Chameleon* was developed to provide a general decoupled approach between the models. The *separation of concerns* principle was applied to disconnect the models, their data and a underlying simulation framework as clearly as possible. The *Chameleon* ideas have been realized on top of the integration frameworks *ModelCenter* and *Remote Component Environment.* The feasibility and the advantages of this concept will be pointed out in this paper. After discussing our experiences with drawbacks and merits of the currently used commercial framework and the transition to an open-source framework we give an outlook on future topics, which are relevant for a simulation software integration in scientific collaboration on a daily basis.**

***Keywords*—Scientific simulation; integration; workflow; collaboration; framework.**

I. Introduction

Interdisciplinary collaboration in scientific simulation, modeling and exploratory research is a common activity among scientists in universities, companies and federal institutions. The requirements for this kind of locally distributed cooperations between researchers from very diverse fields of science are well-known [1], [2], and many software systems try to fulfill them and provide researchers with tools such as integrated environments, simulation data browsers and provenance explorers to deal with the associated technical challenges in scientific knowledge accumulation.

To our knowledge, there is no all-in-one solution available yet, but rather a lot of specialized and/or experimental academic tools, and a huge amount of commercial products that often fail to address the specific needs of scientific data management, flexibility to change workflows at any time, real-time monitoring and powerful post-processing facilities. Therefore, aside from similar solutions for this vast software field like Keppler & Taverna [1], [3], iSight [4], and many more, also at the German Aerospace Center (DLR) efforts were undertaken to create a suitable system, originally in the field of aircraft predesign, later adapted for use in more loosely related fields such as assessment of air traffic climate impact in the DLR-project *climate-compatible air transport system (CATS)*, innovative concepts in engine design in *Evaluation of Innovative Turbo Engines (EVITA)*, and of national and international air traffic modeling *(IML2)*. The software system created at DLR and presented in this paper certainly does not want to be the future all-in-one solution described above, but strives to reach a new level of conceptual and implementation abstraction and decoupling of its inter-component relations.

As already presented in previous papers [5], [6], there has been software development going on at DLR to provide researchers in aerospace-related fields with tools necessary to run distributed simulations and experiments on top of the existing commercial integration framework *ModelCenter* [7]. Use cases in scientific software integration were already presented elsewhere [8], [9]. In this paper we present a generalization of the simulation environment architecture we developed over the last few years.

The paper is organized as follows: Section II gives an overview about what we call the *”Chameleon-principle”* and shows the advantage of using a generalized and decoupled approach in scientific simulation workflow systems. Section III describes the original simulation environment set-up we used and outlines the main advantages and challenges with regards to the tenets of the *Chameleon* principle. In Section IV the new simulation environment is introduced with its (dis-)advantages regarding *Chameleon.* Section V draws a conclusion including a brief evaluation of current outcomes and how to proceed with further development to steadily increase the usefulness and universality of our software design for researchers and engineers.

II. The Chameleon simulation environment

In short, *Chameleon* is the name of a concept for simulation and scientific software integration environments, targeted at experimenting and collaborating in interdisciplinary scientific simulation, modeling and assessment.

During the DLR-projects TIVA, TIVA-II, UCAV-2010, CATS and EVITA [10] work on a generalized simulation environment began. Those efforts culminated recently in a product called ”Chameleon Suite”, which denotes rather a design principle than a singular software product, because there are several ways that *Chameleon* can be (and was to be) turned into executable software. Since commencing development it was determined that for the software to succeed there must be a modular and highly abstracted software design, since the projects listed above all stretched over several years and follow-up projects needed and need to be able to build upon the existing software product, while new requirements may arise at any point. The design therefore follows the principle of *separation of concerns (SoC)* to allow for later changes in any separate part without inflicting consecutive adaptations in the other parts. Figure 1 shows the general idea of separating three distinct parts of a simulation environment, namely the scientific simulation applications (from now on called ”tools”) with their respective specific input / output demands, the common data format *Common Parametric Aircraft Configuration Scheme (CPACS)* [11], the software libraries that can be used by framework components as well as users' applications, and the software integration framework (also called ”platform”), on which our software is built on.

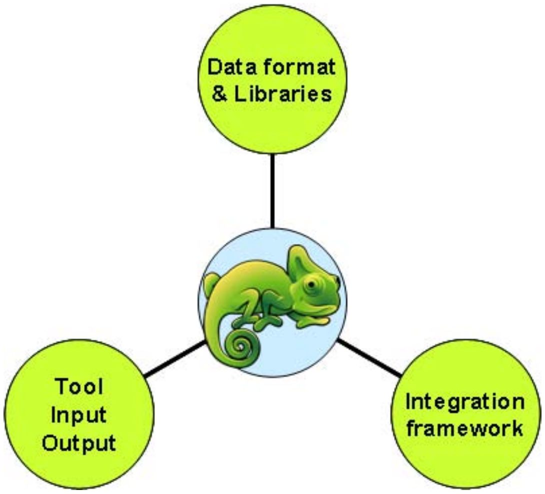


Figure 1. Separation of concerns by decoupling core components in the Chameleon simulation environment.

A. Scientific tool integration

The integrated simulation tools need to be separated from the integration platform as well as from common functions (externalized into reusable software libraries) and the data format used to communicate between tools. Also, the input and output data formats of integrated tools may differ from the data format used to transfer information *between* tools, when they are called from a workflow engine that controls the execution flow, which is most often provided by the integration framework.

B. The common data format

The data format and its underlying data model as the original driving core component, originally launched in the TIVA project, was specifically designed *independently* of any implementation details, to capture the structural and parametric description of aircraft configurations. The data format is wished to progress and extend independently from the integration environment on its own, although often changes arise from current project demands. Changes to the data format must only conform to the data integrity and domain constraints, but not to the technical realization of any tool using it. This way progress in disciplinary research is encouraged and not hindered by dependencies on libraries and existing integration state. Chameleon offers components to work with CPACS data; this contains importing, exporting and updating, spanning over several XML-files, including remote web-resources.

C. The software integration framework

The software integration framework is the biggest asset but also often the most aggravating lock-in for every collaborative simulation, or other workflow environment. In order to keep even this heavy-weighing part on which everything else rests upon interchangeable, we developed all other core components in a way to stay decoupled and independent of the underlying framework. This approach seems quite bold, because frameworks differ largely in their capabilities regarding parallelization, workflow, numeric optimization, infrastructure, architecture, but also their supported data types, structures and runtime models. Regardless of this, we will show that it is possible to abstract a simulation framework from the underlying integration platform while not only retaining a useful and usable workflow and experimentation system, but also gaining all features that emerging frameworks may offer.

D. The common software libraries

As shown in Figure 2, there are two helper libraries for data manipulation and direct data access, namely *TIXI* and *TIGL*. They have been named after the TIVA project for historical reasons.

The first supporting library that was developed was the *TIVA XML Interface (TIXI)*. Since CPACS data is completely based on XML, to simplify the access to the central data format the TIXI library has been developed to shield the application developer from dealing with the complexities of XML structure handling. Dealing with the full functional range of XML is mostly not necessary because many applications use input files based on quite simple data structures. These data structures are often single floating point or integer numbers, and their meaning depends on the exact position of these numbers in the input or output file. More advanced types of these files are name/value pairs or lists of numbers to reflect vector or array data. On top of the full-fledged XML-library *libXML2* [12], the C-library TIXI was designed to shield the developer from XML processing when performing simple tasks like writing an integer or reading a matrix.

A second dynamic library we developed is called *TIVA geometric library (TIGL)* and can be used for easy processing of geometric data stored inside CPACS data sets. With TIGL it is possible to directly execute geometric functions on fuselage and wing geometries. In the future, the functional volume of the TIGL library shall be extented to handling of other aircraft parts like for instance engine nacelles. The geometric library itself uses again the TIXI library to access CPACS data sets, while leveraging data manipulation of all supported geometry types to, e. g., build up a prevalent 3D-Model of the contained aircraft in memory. For the time being only wings and fuselages are supported, with more to come. The functional range of the library goes from the computation of surface points in Cartesian coordinates up to export of airplane geometry to different file formats (IGES, STL and VTK). Beside these computational functions, TIGL can be used to obtain information about the geometric structure itself, for instance how many wings and fuselages the current airplane configuration has.

The overall architecture regarding those mostly independent software parts is displayed in Figure 2.

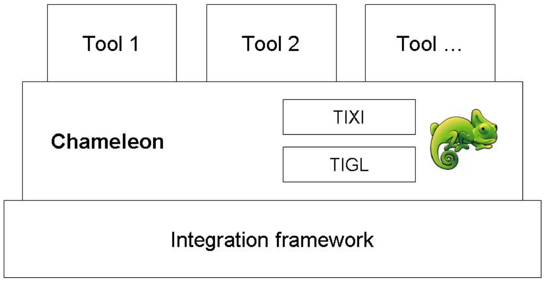


Figure 2. Coarse depiction of Chameleon’s architecture.

E. Advantages of the Chameleon approach

The general idea of our decoupled architecture is to simplify the technicalities connected with distributed computing and provisioning of highly specialized simulation tools of different scientific fields as much as possible. Therefore, the simplicity to set up and configure workflows and to introduce new tools into the environment was one of the main drivers during development.

The second most important advantage of the Chameleon approach is the additionally gained ”*business*” value stemming from the decoupling principle: Every newly integrated tool that is based on the abstraction of core components as explained above instead of on legacy technology provides an instant bonus to current and future simulation projects because it is abstracted in a way that allows not only its reuse in other projects and contexts, but also in other integration frameworks and with other data formats. This is, in a nutshell, the main advantage of adhering to a strict SoC in a scientific workflow environment, that, to our knowledge, is unprecedented. Interfacing with a common data format increases interoperability and cooperations become easier to start with each newly added tool. Inclusion of many tools into one environment boosts productivity and reproducibility of results. Validation capabilities are much stronger when working with workflows instead of plain, manually set up process chains (like batch files or hand-crafted converter scripts).

Additionally to the separation of core parts in Chameleon, a host of different helper components is deployed for every Chameleon version, spanning convenient helper components for finding errors, sending email-notifications (useful in long-running workflows), graphical components for data extraction, data mapping, three-dimensional visualization plugins, script language integration, logging and more. These components cannot always be copied as they are from one Chameleon instantiation to another Chameleon implementation based on another integration framework, but they are rather provided, pre-manufactured and supported by the *Chameleon* team to comprise a comprehensive collection of tools needed to design scientific workflows, experiment with domain values and exchange data and ideas with colleagues. The usage of the libraries, of the data format, and most of the workflow features are virtually just the same, from a user’s perspective, regardless of the platform used.

In the following two subsections we will present the currently supported integration frameworks that Chameleon was developed to run on, along with their respective advantages and drawbacks.

III. PHOENIX MODELCENTER

The simulation environment *ModelCenter (MC)* is a commercial product from Phoenix Integration, a software vendor located in Philadelphia, USA. MC is based on a client / server architecture where the *ModelCenter* application itself is a client-side *Microsoft Windows* application to build and monitor simulation workflows. On the server-side an application server called *Analysis Server* is used to provide the software infrastructure for distributed execution of simulation models. Alternatively, a product called *CenterLink* can be used, which allows asynchronous and queued execution of packaged models independently of the MC client. Simulation models can be constructed as scientific workflows in MC and run either in the the integrated workflow environment or on the above-mentioned server. A scientific workflow can be either a data- or process-driven tool chain with capabilities similar to the *Business*

Process Modelling Notation (BPMN). The following sections enumerate the key advantages and drawbacks of MC when used as the foundation for the Chameleon simulation environment:

A. Advantages

- Tool based simulation approach; application wrapping
- Runtime tool execution depending on fixed number of input- and output *variables*, synchronous data transfer
- Data- and process-driven approach available
- Choice of several workflow-driving schedulers
- Many integration components included: Optimizers, Calculation software integration (e. g., , Excel, Matlab, ANSYS)

B. Disadvantages

- Even moderately large data sizes not supported
- No integrated support for XML-based data formats (yet), although internally running on XML-technology
- No server-based GUI for the individual tools providable (availability not foreseeable in the future)
- No runtime support for wiring up complex workflows (e. g., wizard functionality)
- Hard to conceive ways on how to add support for collaboration in teams
- No possibility to add additional cross-cutting technologies like, e. g., provenance, data management, user access rights; may be included in an upcoming version

On the one hand, the listing shows that the simulation environment comes with all general functionality to work with the basic idea of *Chameleon* in combination with simulation workflows. But on the other hand, the disadvantages bar us from building a really integrated simulation environment, which addresses the needs of researchers regarding a full-fledged collaboration software. *ModelCenter* is a useful software for creation of simple simulation workflows, but a simulation environment should address not only the process, but also potentially complex and dynamic data (-management) and the collaboration issues.

IV. Remote Component Environment

The integration framework *Remote Component Environment (RCE) [13]*, formerly known as *Reconfigurable Computing Environment*, is an open-source product from the DLR institute *Simulation and Software Technology (SC)* [14], department *Distributed Systems and Component Software (VK)* and the Fraunhofer Institute for Algorithms and Scientific Computing (SCAI). *RCE* is a grid component framework for distributed computing based on the *Eclipse Rich Client Platform (RCP)* and provides core functions needed in a distributed environment. These features, among others, are:

- Authorization and authentification (AA)
- Detaching of running workflows from the client's GUI
- Operating system independency due to use of the RCP and Java platform (Linux, Windows, Solaris, AIX, Mac)
- Modular framework based on OSGi [15] allows for further enhancements regarding emerging technologies, e. g., provenance and knowledge management
- Inter-instance file transfer and notifications
- Visual workflow editor
- Data management
- Distributed logging view

RCE was originally designed in a project concerned with the predesign of ships in collaboration with dockyard companies. Because predesign processes for ships have a lot in common with the predesign tasks for aircraft, RCE was a natural candidate to create a Chameleon environment on. Similar to the structure of the previous section, an examination regarding advantages and disadvantages follows:

A. Advantages

- Tool-wrapping for simulation tool integration
- Runtime tool execution based on a data-channel concept, leading to asynchronous data transfers
- Data-driven approach, allowing streaming of input and output data values
- GUIs allowed for server-side simulation tools, accessible from every RCE instance in a network
- Support for wiring up complex workflows (wizards)
- Possibility to extend the system for collaboration tasks
- Open architecture allows to add additional technologies like provenance, data management and visualization

B. Disadvantages

- No process-driven approach available yet
- No workflow-scheduling approaches, logic is currently implemented only in the integrated components
- Young product; needs an active community, code committers and operational experience

It is obvious that *RCE* is able to execute simulation workflows as well. The main advantage of *RCE* is the flexibility of being built on top of the open source RCP and OSGi platforms and therefore having the opportunity to add collaboration topics and additional technologies to it. For institutions like DLR, working on strongly multidisciplinary projects with locally distributed teams of manifold professions there is a need for supporting and integrating new technologies in addition to create and execute workflows whenever need arises.

V. Conclusion and future work

The transition from a data format and tool integration in ModelCenter – which was developed over several years and in several projects as mentioned in Section I – to the new integration framework RCE, was prototypically exercised and evaluated. The software development overhead was quite small in comparison to the former projects, because of several facts: The data format and its accessor libraries were already created, tested and broadly in use. Installation overhead is minimized, since the infrastructure is already in place and cooperation in server configuration is widely given and responsibilities are accepted. The component framework offered by the OSGi-platform reduces the overhead (”boiler-plate code”) needed to create and integrate new components. The shortcomings of the existing framework were largely known and many workarounds had been in place. Therefore in the new framework we could often directly use its capabilities to reach the same goal without having to resort to a workaround. In many parts, the RCE framework had differing concepts and a different philosophy to the definition of workflow. Here the challenge lay in the fact that most logic for RCE integration goes into the components instead of the workflow engine. We hope that upcoming versions of RCE will have more powerful features to ease the developer’s overhead to integrate components in a useful and usable way.

In this paper we have shown the advantage of a generalized and decoupled concept for integrating data, tools and frameworks. We demonstrate how easily our *Chameleon* principle can be adapted to the open source grid computing framework *RCE*. The reason for changing the framework is, beside the proof of generalizability, that the new framework is a more eligible candidate for a real simulation system. This is because we learned that collaboration in teams and support of users in creating a workflow is much more important than just to execute the same process over and over again without changes. The wide field of collaboration is, beside the core integration technology, the next main approach we want to address in our *Chameleon* software suite. Our future work will be about the integration of provenance information and to provide a possibility to semantic checks during creation of workflows.

References

[1] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, “Scientific workflow management and the kepler system,” *Concurrency and Computation: Practice and Experience*, vol. 18, no. 10, pp. 1039–1065, 2006.

[2] T. Schlauch and A. Schreiber, “Datafinder. a scientific data management solution,” in *PV 2007*, Oct. 2007. [Online]. Available: http://elib.dlr.de/53369[Online;2010-07-19]

[3] T. Oinn, M. Greenwood, M. Addis, M. N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe, “Taverna: lessons in creating a workflow environment for the life sciences,” *Concurrency and Computation: Practice and Experience*, vol. 18, no. 10, pp. 1067–1100, 2006.

[4] “SIMULIA iSight website,” http://www.simulia.com/academics/isight.html, [Online; 2010-07-19].

[5] A. Bachmann, M. Litz, M. Kunde, and A. Schreiber, “A dynamic data integration approach to build scientific workflow systems,” in *Proceedings of the 4th International Conference on Grid and Pervasive Computing*, vol. 1, Geneva, May 4. – 8. 2009.

[6] A. Bachmann, M. Kunde, M. Litz, A. Schreiber, and L. Bertsch, “Automation of aircraft pre-design using a versatile data transfer and storage format in a distributed computing environment,” in *Advanced Engineering Computing and Applications in Sciences, 2009. ADVCOMP ’09. Third International Conference on*, Oct. 11. – 16. 2009, pp. 101 – 104.

[7] “Process integration & design optimization,” http://www.acel.co.uk/pdfs/designsimulation/, [Online; 2010-03-19].

[8] L. Bertsch, G. Looye, T. Otten, and M. Lummer, “Integration and application of a tool chain for environmental analysis of aircraft flight trajectories,” in *The 9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, Sep. 21. – 24. 2009.

[9] C. M. Liersch, T. Streit, and K. Visser, “Numerical implications of spanwise camber on minimum induced drag configurations,” in *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, USA, Jan. 5. – 8. 2009.

[10] “DLR project websites,” TIVA: http://www.dlr.de/fa/desktopdefault.aspx/tabid-1474/2079_read-3561/, TIVA-II: http://www.dlr.de/fw/desktopdefault.aspx/tabid-2959/4455_read-20454/, UCAV-2010: http://www.dlr.de/sc/desktopdefault.aspx/tabid-5141/8654_read-11626/, CATS: http://www.dlr.de/pa/desktopdefault.aspx/tabid-4618/, [Online; 2010-07-19].

[11] D. Böhnke, “Dataintegration in preliminary airplane design,” Master’s thesis, Universität Stuttgart, Jul. 2009, Work in progress, to be published in Sep. 2009.

[12] “libxml2,” http://www.xmlsoft.org, [Online; 2010-04-01].

[13] D. Seider, “RCE homepage,” http://www.rcenvironment.de, 2008, [Online; 2010-07-19].

[14] “German Aerospace Center website,” http://www.dlr.de/sc, [Online; 2010-07-19].

[15] “OSGi Wikipedia entry,” http://en.wikipedia.org/wiki/OSGi, [Online; 2010-07-19].

Communication Delay Modelling and its Impact on Real-Time Distributed Control Systems

Rachna Dhand
School of Engineering and Energy
Murdoch University
Murdoch, WA, Australia
R.Dhand@murdoch.edu.au

Gareth Lee
School of Engineering and Energy
Murdoch University
Murdoch, WA, Australia
Gareth.Lee@murdoch.edu.au

Graeme Cole
School of Engineering and Energy
Murdoch University
Murdoch, WA, Australia
G.Cole@murdoch.edu.au

***Abstract*—Communication delays are random in nature. A distributed real-time control system linked through a communication network is bound to be affected by the randomness of communication delay patterns. Statistical modelling techniques, like Auto-regressive Integrated Moving Average (ARIMA), may be used to model the network traffic. This paper provides a comprehensive coverage of network traffic modelling through the stochastic approach ARIMA with a case study of National Instruments (NI) DataSocket Transport Protocol (DSTP) based on high bandwidth Ethernet. In real-time control systems the controller optimization requires accurate temporal specification of sensitive controller tasks. Logical computation languages such as Time Definition Language (TDL) have successfully eliminated the temporal inaccuracies in designing control software. The paper provides an analytical and programmatic view on the impact and compensation of unpredictable network delays through discrete-time control algorithms, that are designed in Time Definition Language (TDL). The results validate that the discrete implementations are able to compensate for the delay, thus guaranteeing the stability of the control loop in the presence of unpredictable delays.**

Keywords-Real-Time Control Systems; Time Definition Language (TDL); Auto-Regressive Integrated Moving Average (ARIMA; Network Delay, Smith Predictor; Buffered Time-stamped Dahlin Algorithm

I. INTRODUCTION

A complex industrial control system is designed in a hierarchy as:

1) Supervisory and Control Software at the top layer.
2) Control equipment and PLC's linked together to form a middle layer.
3) Control devices like sensors and actuators at the bottom layer.

The system is finally connected via a communication network. Thus a network control system requires at least one link to be carried by a real-time network [11][12]. The most preferred network protocols for control systems are Ethernet-based MODBUS, PROFIBUS, or Controller Area Network (CAN). The time delays are not always local to the controller tasks. They can occur as transmission delays from a sensors to a controller and from controller to an actuator [9][10] because control equipment is connected via network.

The aim of this paper is to model network induced delays and analyse their impact on control operations through a posterior analysis. The paper provides comprehensive coverage of the statistical modelling technique ARIMA, used to model network traffic with a case study exploring National Instruments' (NI) DataSocket Transport Protocol (DSTP) based on Ethernet. The last section analyses the impact of delays on control loop operation and compares several delay compensation algorithms for control system design. A second order interacting system for liquid level within two tanks is considered. The control data is managed through National Instruments' DataSocket Server Manager. The delay modelling is consequently done for Gigabit Ethernet traffic based on DataSocket Transport Protocol (DSTP).

The paper is organized in four sections. The following two sections explain the role of delays and their modelling for Industrial Control Systems. Section four presents an overview of the statistical delay modelling technique ARIMA for network traffic. The case study for Ethernet traffic is analysed . The last section provides a brief overview of the Smith Predictor and Dahlin technique, and their implementation in Time Definition Language (TDL) for delay compensation of control loop operation. The aim is to analyse a typical control problem

ISBN: 978-1-61208-101-4

for real-time requirements, precise modelling and compensation of network delays to optimize the performance of the system.

II. Industrial Control System Architecture

Communication delays are random in nature, hence are difficult to model. Moreover they are affected by the type of network protocols in use. In an industrial control system, the role of the network becomes more crucial if the plant operates in real-time mode. The network layer forms the central data highway with which display and control equipment are exchanged. A challenging problem in control of an industrial plant is minimization of delay within a control loop. The time delay in executing the control algorithm originates from:

1) Control operation;
2) Sampling time chosen if a discrete-time controller is used;
3) Transmission delays due to network characteristics like network protocol in use, the network topology, or the type of physical network hardware used.

Real-time programming methodologies like RT-Java or Timing Definition Language (TDL) have evolved in the last decade to address the increasing complexity of control systems and scalability of equipment in the industrial automation domain. As explained by C.M. Kirsch, and R. Sengupta in their work in [1], the programming abstractions for real-time systems can be classified according to the processor execution time cycle for a given control task. The developments achieved by the computing community through RT-Languages like TDL conform to compensation of one-unit delay in control algorithm execution. TDL achieves timing predictability through the time-triggered architecture [2]. It executes the real-time code by separating the platform dependant issues like schedulability from platform independent issues like generating code from a given SIMULINK model of the system. TDL integrates well with simulation and modelling environments such as SIMULINK [3]. TDL guarantees control stability once the computational delays within the control loop operation are known. Transmission delays, on the other hand also play a vital role in control system's stability. The transmission delays are tough to model and estimate because of the stochastic nature of network traffic. Many statistical models are available these days to model the behaviour of network traffic and hence estimate the average network delay. The focus in this paper is on ARIMA.

III. Mathematical Modelling for Time-delayed Control System

The Laplace transform models delays in transfer function using the exponential term e$^{-\alpha s}$, where α is the associated time delay that can either represent an input delay or output delay [1]. Thus mathematically the transfer function can be defined as:

$$g(s) = \frac{K}{\tau^2 S^2 + 2\zeta\tau s + 1}\, u(s)\, e^{-\alpha s} \qquad (1)$$

where g*(s) is the non-delayed transfer function for the system. Here α signifies the delay that must be taken into account while designing continuous or discrete controllers. The stability of an overall system depends on all elements that make up the control system architecture, including the communication network. So long as real-time operating system and software development methods are employed, the computational delays can be assumed to be fixed. A well-designed discrete model of the system makes critical assumptions about the controller gain and sampling time. The critical assumptions for a closed loop system are given with characteristic equation specified as [4]:

$$1 + g_p(z)*g_c(z) = 0 \qquad (2)$$

where $g_p(z)$ is the z-transformation of a continuous plant transfer function and $g_c(z)$ is the discrete controller transfer function. For a discrete-time closed loop system to be stable, all the roots of equation 1, must lie within the unit circle [4].

One important factor that is still not taken into consideration is the effect of random delays that come from the communication medium in the control loop operation. The stability margins specified above lay stress on retuning the controller parameters for a discrete-time system to take into consideration the impact of the A/D and D/A converters introduced within the control loop. In order to analyse the impact of delays that originate from a communication medium, the first step is to mathematically model the control system for a communication delay.

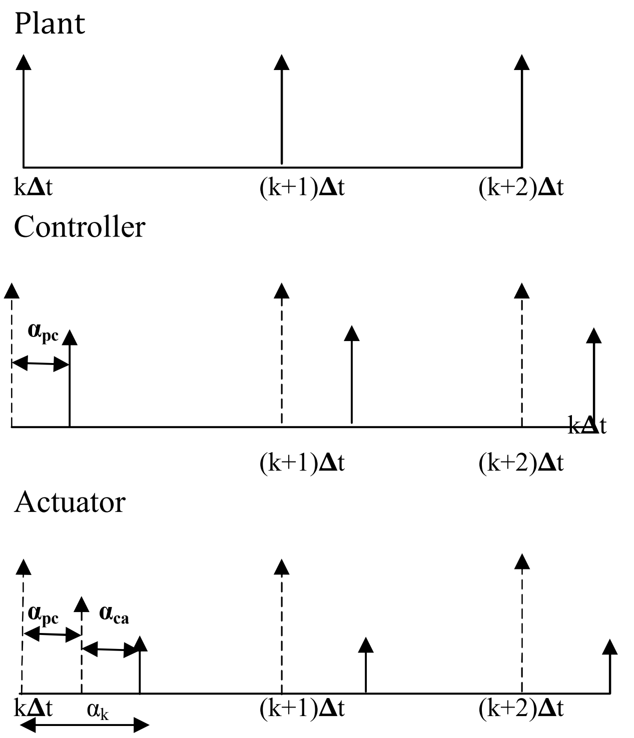


Figure 1. Delay effect for a particular time instant

The delayed response is characterized in two cases for a discretized system for any time instant k and for **Δ**t sampling period. The first case is when the delay magnitude is smaller than the sampling period. The control loop operation is delayed, but tolerable by the system. As is evident from Fig. 1, the total delay is specified as

$$\alpha_k = \alpha_{pc} + \alpha_{ca} \quad (3)$$

where α_{pc} is the network delay from Plant to Controller; α_{ca} is the network delay from the Controller to an Actuator. The maximum delay tolerable by the system, guaranteeing stability, is specified as [5],

$$\frac{d}{dx}(x_p(k+1)\Delta t)=\phi_P x_P[k\Delta t]+\phi_P \alpha_k u_P[k\Delta t]+\gamma_p[\alpha_k]u_p[(k-1)\Delta t] \quad (4)$$

where $\phi_P = \int_0^{\Delta t-\alpha_k} e^{As}\, ds$ and $\gamma_p = \int_{\Delta t-\alpha k}^{\Delta t} e^{As}\, ds$

The maximum tolerable delay can be estimated from the relative magnitudes of ϕ_P and γ_p[5].

The second case arises when the delay magnitude becomes larger than the sampling period. The main consequence for such a case is loss of information or jitter. It is very important to compensate for the delay by controller optimization to stabilize the control loop behaviour. One novel way to handle the communication delay is to estimate the delay magnitude by appropriate stochastic models like ARIMA and then generate forecasts that can be used for controller modelling and delay compensation. The delay magnitude can be compensated by retuning the discrete models with appropriate predictive control techniques such as the Smith Predictor.

IV. Statistical Modelling of Network Delay for Control Networks

Communication network traffic is stochastic in nature, since it arises from multiple independent sources, and is therefore often modelled using statistical approaches. The statistical approaches, specifically time-series models help to generate forecasts that can be used effectively. ARIMA (Autoregressive Integrated Moving Average) [6] is a time series model to capture the behaviour of the network traffic. An ARIMA (p, d, q) is a process where p is the autoregressive order, q is the moving average order and d is the differencing order [6]. Since network traffic always predicts non-stationary behaviour, the most effective modelling technique must consider both autoregressive and moving average terms. The importance of an ARMA processes lies in the fact that a stationary time series may often be adequately modelled by an ARMA model involving fewer parameters than a pure AR or MA processes alone [6]. The general ARIMA (p, d, q) process is of the form [6]

$$W_t = \alpha_1 W_{t-1}+\ldots.+\alpha_p W_{t-p}+Z_t+\ldots..+\beta_q W_{t-q} \quad (5)$$

or such that $\Phi(B)W_t = \theta(B)Z_t$

where B is the backward shift operator specified as $BX_t = X_{t-1}$, Wt is $(1-B)^d X_t$, ϕ and θ are polynomials of order p and q respectively, and X_t represents the original non-stationary time series with mean zero and variance σ^2.

The non-stationary series is often dealt with using the Box-Jenkins [6] approach. It assumes the parameter d is used for differencing the series to induce stationary behaviour. This approach deals with the periodic component of the time series. Box-Jenkins generalized the ARIMA(p,d,q) model to deal with seasonality, and defined the general multiplicative seasonal ARIMA model as [6][7]

$$\Phi_p(B)\Phi_P(B^s)W_t = \theta_q\Theta_Q(B^s)Z_t \qquad (6)$$

where Φ_P and Θ_Q are autoregressive and moving average polynomials of the order P and Q , B^s is specified as $B^sX_t = X_{t-s}$ and s is the seasonal span. The model in equation (6) is a multiplicative seasonal ARIMA model of the order $(p,d,q)*(P,D,Q)_s$ [6][7].

When fitting a seasonal model, the preliminary observation regarding the non-stationary series is to identify the difference (d) and seasonal difference (D) order. Then the values of p, q, P, and Q are determined from the autocorrelation and partial autocorrelation functions of the differenced series. The differencing parameters yield the W_t series which can be modelled using equation (6), to finally produce a fit of a SARIMA (Seasonal ARIMA) model. The fitted model is finally checked for its preciseness i.e. how adequately it represents the original time series data. This step is called the residual analysis. The residuals are often obtained as a difference of actual and fitted values.

For a good model, the residuals are random and have small variance. The most common test used for residual analysis is the Portmanteau lack-of-fit test [6] that uses the chi-square statistic. The test statistic is given as [1]

$$Q = N\sum_{k=1}^{K} r_{z,k}^2 \qquad (7)$$

where N is the number of terms in the differenced series, and K lies in the range 15 to 30.

Q is the chi square statistic with (K-p-q) degrees of freedom. One useful application of this model is for making predictions based on the past data values. The time series that shows trend and seasonality in its behaviour is often forecasted using the *model equation [6]*.

Consequently the ARIMA algorithm for traffic prediction can be summarized by the following steps:

1) Model Specification which involves identification of trend and seasonality within the original series (Random Walk Model), finally identifying the differencing order for the trend and seasonal components.
2) Model Building, which involves identifying the Autoregressive (p,P), and Moving Average (q,Q) orders for general and seasonal patterns from the autocorrelation and partial autocorrelation patterns of Original Series.
3) Model Validation, which involves a validation check for residuals obtained after fitting the model with appropriate orders for AR and MA terms.
4) Forecasting, this involves generating the forecasts if the model is validated and the residuals are random and have small variance.

Experiments were performed to capture real-traffic traces to enable prediction of Ethernet traffic carried using the DataSocket Transport Protocol (DSTP) as application layer protocol. DSTP is the TCP based protocol that manages the National Instruments' (NI) DataSocket Server Manager. The DataSocket Server Manager is a data repository for control data. The delays induced by the network were modelled and forecasted using SARIMA model with (p,d,q)*(P,D,Q) order. DSTP uses a publish-subscribe pattern for data exchange. A system participating in a DSTP data exchange usually consists of three components – a publisher, the DataSocket Server and one or more subscribers [8]. A publisher acquires data from a data acquisition device and sends it to the server [8]. The server may be located on the same machine or, remotely on a local network. Subscribers can subscribe to receive the data from the server [8]. Complex applications may have decentralized setup for subscribers and publishers [8]. The interesting factor that determines the data transfer from publisher to subscriber is that publisher broadcasts the entire active control data to a subscriber thus creating lot of network overhead. This can lead to poor scalability of the network as more and more subscribers become involved.

ARIMA modelling was undertaken for one and three subscribers. The response time behaviour induces a seasonal impact in the inter-arrival time gaps. The seasonal component is more obvious in the case of a single subscriber and dies out as more subscribers come into existence. Since the entire control data is broadcasted to the subscribers, the transmission involves fifteen data packets of equal size. The fifteen data packets are transferred within the time range of 10 milliseconds to 150 milliseconds, thus causing seasonal impact on the autocorrelation patterns. The traffic patterns for a period of ten seconds are recorded and autocorrelation graphs for Random Walk models are shown in Fig. 2. As is evident from Fig. 2, the seasonality of the time gaps is obvious in case of one subscriber but is less clearly defined as the number of subscribers increases. The seasonal component is identified at the multiples of time-lag 16, as there are fifteen response data packets transferred to the subscriber for every request. The time gap for the first response packet is high and decreases for subsequent response packets, thus causing the seasonal impact.

The autocorrelation patterns and partial autocorrelation patterns specify the order required by the SARIMA model. The best fits for both traffic patterns were obtained using Minitab 15 for orders as specified in Table 1. The residual autocorrelations for both models are shown in Fig. 3. As evident from the Fig. 3, the residuals are random and close to zero. Consequently the fitted models were deemed adequate. Forecasts obtained through ARIMA are based on specific assumptions that are assumed to be fixed for a particular set of experimental data.

TABLE 1. FITTED MODELS FOR DSTP TRAFFIC PATTERNS

Subscribers	Model

1 Subscriber	**SARIMA(0,0,1)*(1,1,0)$_{16}$**
	`Type           Coef    SE Coef`
	`SAR  16     -0.6317     0.0403`
	`MA   1       0.3603     0.0487`
	`Constant  0.0001944  0.0007260`
	`Differencing: 0 regular, 1 seasonal of order 16`
	`Number of observations:  Original series 443, after differencing 427`

3 Subscribers	**SARIMA(5,0,0)*(0,1,1)$_{56}$**
	`Type              Coef      SE Coef`
	`AR   1        0.0147      0.0327`
	`AR   2       -0.0394      0.0327`
	`AR   3       -0.0578      0.0327`
	`AR   4       -0.0379      0.0327`
	`AR   5       -0.0556      0.0328`
	`SMA  56       0.9484      0.0139`
	`Constant  0.00014991  0.00004716`
	`70.90  0.000`
	`Differencing: 0 regular, 1 seasonal of order 56`
	`Number of observations:  Original series 994, after differencing 938`

V. DELAY COMPENSATION ALGORITHMS: A-POSTERIOR ANALYSIS

The ARIMA technique helps in modelling and forecasting network delays. The forecasted delay pattern can be used by discrete control algorithms for delay compensation. The control algorithm for the second order water tanks problem was optimized to compensate for delays induced by the network. The control data was managed by National Instruments' DataSocket Server Manager. The delay forecasts generated through ARIMA were utilised to predict communication delays within the control loop. An analytical view of control loop operation was generated in the presence of these forecasted delays using MATLAB/SIMULINK. The control algorithms were implemented using the Time Definition Language (TDL) to guarantee the computation time stability.

The two most common techniques for delay compensation in discrete control design are [4]:

1) The Smith Predictor
2) The Buffered Dahlin Algorithm

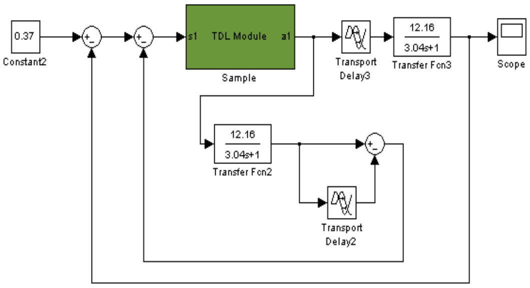
In a closed loop system, the conventional control design strategies allow delay compensation through reduction of controller gain [4]. Consequently the control systems will display sluggish response when compared to a non-delayed system. A better approach is to deal with the delay explicitly and introduce compensation techniques to the allow system to behave like a non-delayed system [4]. The Smith Predictor is a delay compensation algorithm. It assumes that the system is non-delayed and compensates the delay value by introducing a minor loop modelling the non-delayed control loop operation. The Smith Predictor introduces a minor loop that deals with the model process for delay compensation. Thus the error signal after compensation of delay α is

$$e_c = y_c - y^*(s) \qquad (8)$$

where y*(s) is process output for the non-delayed control loop operation. The Smith predictor algorithm allows the controller gains for undelayed process to be used without instability arising within the control loop. The closed loop transfer function for the delayed process using a Smith predictor is defined as

$$Y = \left(\frac{g*g_c}{1+g*g_c}\right)e^{-\alpha s}y_c \qquad (9)$$

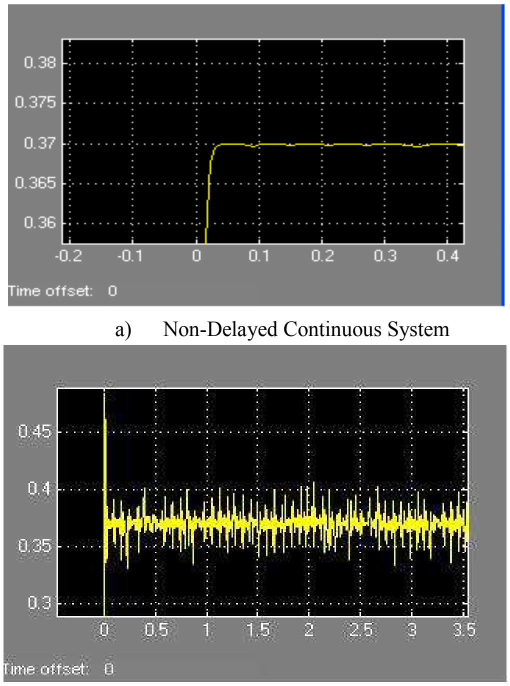
The SIMULINK model for a delayed process using Smith Predictor designed in TDL is shown in Figure 4.



(K_c=0.0833, τ_i=0.0433, $\Delta t_{critical=}$0.009)

Figure 4. Smith Predictor with TDL Controller Optimization

Now the non-delayed process is delayed by the certain amount of time, α, which is assumed to be derived from the ARIMA generated forecasts. The delay value is assumed to have average, best and the worst case values of 60ms, 10ms, and 255ms respectively.



a) Non-Delayed Continuous System

b) Delay Magnitude is 255ms

Figure 5. Control Loop Stability in the Presence of Delay

As is evident from Fig. 5, the communication delays, if greater than the sampling period, can destabilize the system. The discretized system may be compensated for delay using a Smith Predictor. The Smith Predictor is able to compensate the delay and induce stability. But it has certain disadvantages. It assumes the magnitude of the delay is constant, which is not with network communication delays. Network delays are more random and non-stationary over time.

The application of this algorithm must be restricted to accommodate the worst-case delay value which has a rare probability of occurrence. This conventional design strategy does not provide the controller type and parameters. Thus it can act as a major loophole for stability in the discrete-time domain. A realistic system with time delay is inherently unable to respond instantaneously to a control event. The high controller gains fail to provide a realistic image of the process. The remedy to this problem is to design the controller with parameters found using the direct synthesis control strategy with a reference trajectory that can provide desired control loop behaviour. Fig. 6 shows the delayed system behaviour using a Smith Predictor with Direct Synthesis Controller Parameters. The PI Controller used for the α delay in the system is realizable as

$$g_c = \frac{\tau s+1}{K}\left(\frac{1}{\tau_r s+1-e^{-\alpha s}}\right) \quad [4] \qquad (10)$$

The choice of τ_r allows the system to have a more realistic view of the process behaviour in the presence of time delay.

Discrete Model (K_c=0.25 τ_i=0.0822 τ_r=1)

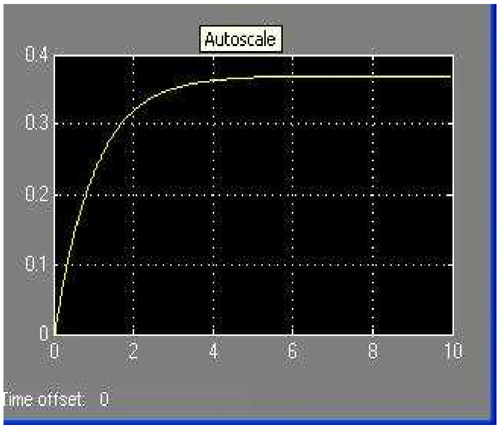


Figure 6. Smith Predictor with Direct Synthesis Control

Another useful strategy for controlling time-delayed systems is the Dahlin Algorithm. It is a digital control design technique that assumes the controller realization including the ZOH (zero order hold) and explicit time delay is [4]:

$$g_c(z) = \frac{1}{g(z)}\left[\frac{(1-e^{-\varphi}{}_{r})z^{-m-1}}{1-e^{-\varphi_r}-(1-e^{-\varphi_r})z^{-m-1}}\right] \qquad (11)$$

where φ_r is the assumed reference trajectory for the desired control loop behaviour. G(z) is the delayed plant model with **Δ**t sampling period such that the time delay α = m **Δ**t. The Dahlin algorithm is a digital control algorithm having two tuning parameters:

1) M, chosen such that the controller is realizable in the presence of delay
2) φ_r determines the speed of the closed loop response.

The forecasted delay pattern follows an ARIMA based model. The model equation can be used to predict the delay for control loop. The scheme uses buffered approach to save the preceding timestamps for the next prediction. The Dahlin strategy is simulated with buffered timestamps for indicating the magnitude of delays using TDL/SIMULINK with sampling time **Δ**t ranging from 0.004 to 0.09 minutes. The plant model is assumed to have the delay specified as ranging from α. = 100 to 255 milliseconds.

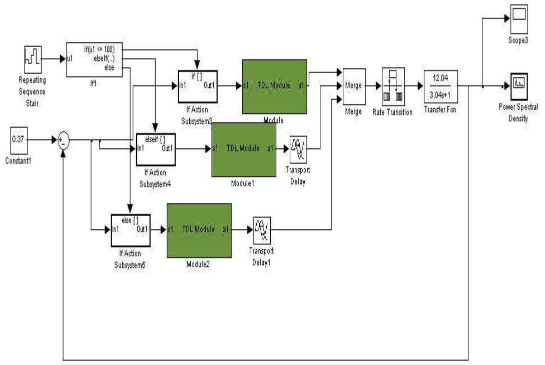
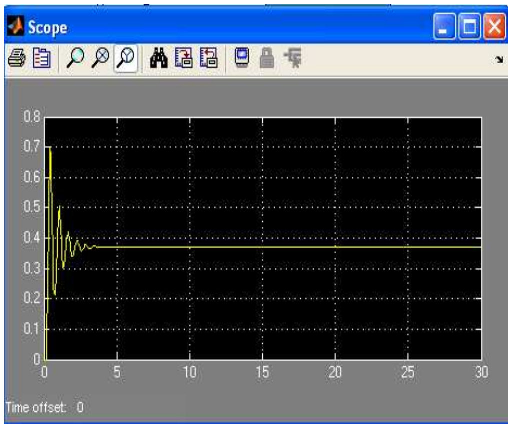


Figure 7. TDL/Simulink Model for Delay Compensation

Listing 1. Buffered Dahlin Algorithm

```
Repeat
// For any time instant k
 $\tau_{k-1}$ = Buffer(k), $\tau_{k-2}$ = Buffer(k-1) // Get the timestamp for the //preceding delay
$\tau_k$ = ARIMA fun($\tau_{k-1}$, $\tau_{k-2}$) // Next Prediction
Read h(k)    // Current Process Variable
e (k) = sp – h(k)
//Random delay block
// Introduce the Computational Delay
//$g_c$(e(k) optimized for delay $\tau_k$ with m units
d = Δt/ $\tau_k$
compute f(k) = $g_c$(e(k), $\tau_k$ , d)
//Manipulated Variable
write f(k)
compute h(k+1) = $G_p$(f(k))
// New Error Signal
Until (Set point)
```

The TDL/SIMULINK model behaviour is shown in Fig. 7 and the response is analysed in Fig. 8.



Kc = 53.933, I = 18.6125 for Delays less than 100ms
Kc = 5.7421, I = 1.8869 for 100< Delays < 200ms
Kc = 2.2734, I = 0.9243 for 200<Delays<300

Figure 8. Dahlin Strategy for Variable Network Delay Compensation

As evident from the discrete implementations of the Smith Predictor and Dahlin Algorithm from Fig. 7 and Fig. 8, stability is achieved but the response time is high in comparison to the continuous system response time. These algorithms are implemented in TDL to guarantee computation time stability, thus providing the precise analytical impact of stochastic network delays on control loop stability.

VI. CONCLUSION

The response times and TDL computation times are summarized in Table 2. These comparative estimates of response times are for the second order system with time constant of 3 minutes. In the first case, the network delays are modelled using the stochastic approach, ARIMA. The forecasts are then utilized to view their impact on control loop operation. The discrete implementation of control algorithm uses TDL to guarantee the computation time stability. RT-Languages such as TDL can help in analysing and guaranteeing the computational stability.

Finally, the unpredictability of communication delays can be dealt with using appropriate discrete algorithm implementations. As is evident from the Table 2, the discrete implementations are able to compensate for the delay, thus guaranteeing the stability of the control loop in the presence of unpredictable delays. However, in practice, the response time maybe high. Secondly, in both strategies the delays are treated as a constant and varied through average, best and worst case values.

TABLE 2. RESPONSE TIME ESTIMATES IN TDL/SIMULINK

Non-Delayed Continuous System		**Smith Predictor**		**Dahlin Algorithm**	
TDL Computation Time(ms)	*Response Time(sec)*	*TDL Computation Time(ms)*	*Response Time(sec)*	*TDL Computation Time(ms)*	*Response Time(sec)*
20	2.4	40	5	40	240
Reference Trajectory (3 Minutes) without Computation time stability		-	230	-	480
Controller Optimization through TDL		40	150	40	350

REFERENCES

[1] C.M. Kirsch and R. Sengupta, “The Evolution of Real-Time Programming”, in Handbook of Real-Time Systems and Embedded Systems by I. Lee, J. Y-T. Leung, S.H. Son, 2007.

[2] T.A. Henzinger, C.M. Kirsch, and B. Horowitz, “GIOTTO : A Time-Triggered Language for Embedded Programming”, in T.A. Henzinger, C.M. Kirsch, Editors, Proc. of the 1st International Workshop on Embedded Software (EMSOFT), number. 2211 in LNCS, Springer, Berlin, Oct 2001

[3] T.A. Henzinger, C.M. Kirsch, M.A.A. Sanvido, and W. Pree, “From Control Models to Real-Time Code using GIOTTO”, IEEE Control Systems Magazine, Feb 2003.

[4] B. A. Ogunnaike and W. H. Ray, “Process Dynamics, Modelling, and Control”, Oxford University Press, 1994.

[5] J. Nilsson, B. Bernhardsson, and B. Wittenmark, “Stochastic Analysis and Control of Real-Time Systems with Random Time Delays”, Automatica, Vol. 34 Issue 1, Publication Year 1998, pp. 57-64.

[6] C. Chatfield, “The Analysis of Time Series An Introduction”, Chapman and Hall/CRC, 2003.

[7] W. Vandaele, “Applied Time Series and Box-Jenkins Models”, Academic Pree, INC, 1983.

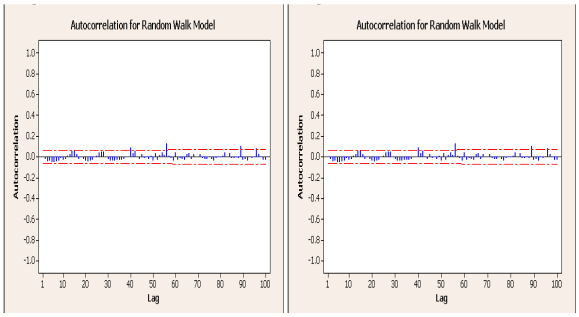
[8] “DataSocket Transport Protocol- Overview”, NI Developer Zone. URL- http://zone.ni.com/devzone/cda/tut/p/id/3223 accessed date: 21st May 2009.

[9] A. Ray, “Distributed Data Communication Networks for Real-Time Process Control”, Chem. Eng. Commun., vol 65, pp. 139-154,1998.

[10] M. S. Mahmoud and A. Ismail, “Role of Delays in Networked Control Systems”, In Proc. of IEEE ICECS-2003.

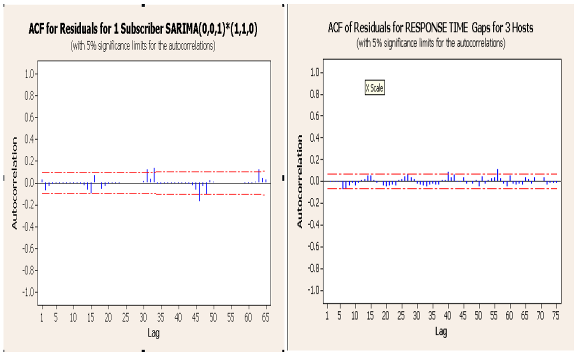
[11] Z. Wei, M.S. Branicky, and S.M. Phillips, “Stability of Networked Control Systems”, IEEE Control System Magazine, vol. 21, no. 1, pp. 84-99, 2001.

[12] Ray, Y and Halevi, “Integrated Communication and Control Systems: Part II Design Considerations”, ASME Journal of Dynamic Systems, Measurement and Control, vol. 110, pp. 374-381, 1988.



1 Subscriber 3 Subscribers

Figure 2. Autocorrelation for Original Series



1 Subscriber 3 Subscribers

Figure 3. Residual Autocorrelations for Fitted Models

Real-time Processor Interconnection Network for FPGA-based Multiprocessor System-on-Chip (MPSoC)

Stefan Aust, Harald Richter

Department of Computer Science
Clausthal University of Technology
Julius-Albert-Str. 4
38678 Clausthal-Zellerfeld, Germany
e-mail: stefan.aust|harald.richter@tu-clausthal.de

Abstract—**This paper introduces a new approach for a network on chip (NOC) design which is based on a NlogN interconnect topology. The intended application area for the NOC is the real-time communication of multiprocessors that are hosted by a single Field Programmable Gate Array (FPGA). The proposed NOC is an on-chip multistage interconnection network for which an upper limit can be guaranteed that is at most needed for the latency while delivering data between sending and receiving processors. The reason for the deterministic interprocessor communication is the constant path length from input to any output port of the NOC. In contrast to contemporary NOCs, no intermediate routers exist. Thus, no overloaded router with hot spot problems can occur, and the proposed NOC can be used for real-time applications. Example NoCs of size 4x4 and 8x8 were implemented in VDHL, together with their softcore processors on Spartan3 and Virtex-4 and -5 FPGAs from Xilinx.**

Keywords–network on chip; multistage interconnection network; softcore processor; real-time multiprocessor; FPGA-based multiprocessor

I. INTRODUCTION

The increasing quantity of logic cells that can be integrated into a single FPGA allows novel solutions by using the system on chip (SoC) paradigm. Just recently, multiprocessor system on chip (MPSoC) applications have become feasible that are hosted by a single FPGA [1,2]. In such MPSoCs, each processor exists only as Verilog or VDHL [3] description that can be extended or modified as needed, and that is afterwards synthesized for a target FPGA such as Spartan3 or Virtex-4/-5/-6 from Xilinx, for example. Because of the adaptability of the processor architecture to the demands of the real-time system, such computing devices are called soft-core or soft processors.

MPSoCs with soft processors exhibit both, the high performance of parallel computers and the flexibility of reconfigurable hardware [4]. In real-time systems, data- and computing-intensive applications can make use of this technological progress. For instance, driver assistant systems in cars require to service more sensors and actuators than ever. Such applications demand higher computing power and less electrical power at the same time, while the system size has to be minimized. To match such demands, the proposed network on chip (NoC) design can be used in MPSoCs. In the future, we believe that MPSoCs will replace in part conventional electronic controller units in automobiles as well as in complex machinery [5].

The majority of embedded systems are located in real-time applications. Amongst others, the real-time performance of multiprocessor computers relies on the predictability of the interprocessor communication. For an MPSoC, deterministic behaviour of the interconnection network has to be guaranteed. This requirement is hardly to implement with conventional packet routing that takes place in direct, i.e. static networks. In static networks, adaptive multi-hop routing together with packet prioritization induces an undesirable indeterminism to network latency. The formation of hotspots due to excessive data traffic in router nodes excludes predictability also. We therefore propose, a new paradigm for MPSoCs, which makes use of multistage interconnection networks (MINs) as a network on chip.

This paper is organized as follows: in section 2, the state of the art in NoCs is given. Section 3 makes a recap of MINs. Their utility and their problems in on-chip usage are investigated in section 4. In section 5, the topology of the proposed NoC is presented, and in section 6 its chosen implementation is described, together with the MPSoC for which it was developed. The paper ends with conclusion and literature reference in section 7.

II. STATE-OF-THE ART IN NETWORKS ON CHIP

Interprocessor communication in MPSoCs with tens of cores or more is no longer feasible by using shared buses due to their low intrinsic scalability in bandwidth and latency [6,7]. Also crossbar structures are no longer practicable due to their O(N^2) complexity if N becomes large. To overcome the von Neumann bottleneck and the O(N^2) increase in hardware, alternative architectures have been introduced by NoCs as the new paradigm in SoC design [8]. Since then considerable number of NoC designs have been proposed which provide diverse communication types and network topologies [7,9]. NoCs with direct (static) networks have been proposed by [10,11,12,13] such as mesh, tree, torus or hypercube. Some examples of these static topologies are given in Fig. 1. The basic principle of direct network topologies is that each processor is connected directly to a smaller number of neighbour processors where each processor acts in addition as a switch or router node for

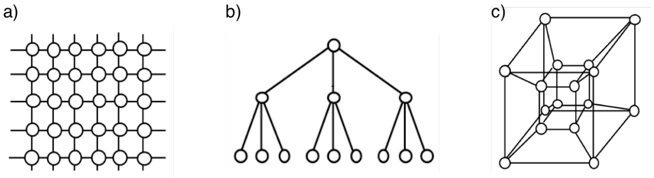


Figure 1. Toplogies of direct networks: a) mesh; b) tree; c) hypercube.

frames or packets, respectively. Routing is performed either statically or dynamically via the source and target information contained in the frame/packet headers. The communication channels between the processor nodes are operating on layer 3 of the ISO 7-layer model. Finally, some nodes provide additional communication channels for the necessary I/O.

A desirable property of NoCs for MPSoCs is scalability, which means that for small and large processor numbers as well, the same basic interconnect structure, can be used in principle. Another property which is mandatory for real-time systems is that the achievable bandwidth and the maximum latency for data transfer is deterministic i.e. predictable. This means that an upper limit for the latency must be guaranteed that is at most needed for data transfer, as well as a lower limit for the bandwidth. This is required to build and program systems that can react in due time.

However, all direct networks have the potential hazard of hot spots that are overloaded router nodes. Hot spots result in unpredictable bandwidth and latency, which in turn is not tolerable for real-time systems. Furthermore, scalability is also not possible if hot spots occur.

This is why we propose an alternative to direct networks that can be used for NoCs in MPSoCs and that is based on indirect networks. In indirect networks, computing nodes are connected via a cascaded set of switches. Because of the switch arrangement, each path from source to target is of the same length, and every switch has to serve only a fixed number of traffic streams. Thus, hot spots cannot occur, if the rearrangable non-blocking subtype of indirect networks is used. Indirect networks are described in the next section.

III. MULTISTAGE INTERCONNECTION NETWORK (MIN)

By origin, MINs were proposed for telephone exchange systems, and later for parallel computers. Vector supercomputers, multiprocessors and multicomputers with processors on individual silicon chips were introduced two decades ago for high-performance computing. MINs have been designed to match their bandwidth and latency constraints and to support effective execution of parallelized algorithms. Therefore, MINs are known from parallel computers, and we have adopted these structures to provide for deterministic on-chip interprocessor communication.

MINs connect computing nodes through a set of elementary switches that are organized in 1, 3 or logN stages, where N is the number of the ports the network features. The mathematical patterns between the switch stages are permutation functions, such as perfect shuffle, butterfly or bit reversal [14]. The Omega network, for example, which was introduced by Lawrie [15], consists of shuffle and exchange permutations in logN stages and can be defined by

$$\Omega_n = (\sigma_n \circ E)^n \tag{1}$$

where n is the total number of stages, σ is the shuffle permutation over n bits, and E is an exchange stage [16]. An example for an Omega network of size 8x8 is given in Fig. 2.

By using the smallest possible switch size of 2x2, the construction of a MIN needs $(N/2)\log_2 N$ switches only, which is the minimum number possible at all. For comparison, a crossbar network requires N^2 switches. Typical representatives for logN-MINs are Omega, Baseline and Butterfly networks [16,17,18,19]. They belong to the so-called delta subclass of MINs which means that routing through the logN-MIN is easily accomplished by using bit after bit of the target port address in order to set each switch so that it routes data either „=“ (parallel) or „x“ (cross). Because of the constant number of switches that data has to pass from the network input port to output port, a constant routing time or at least an upper limit for the routing canbe guaranteed. MINs are therefore beneficial for interprocessor communication with respect to latency, which is important for real-time applications. However, constant routing time cannot be guaranteed for transfers that take place at all input ports simultaneously. The reason is that each output can be reached from every input in principle, but there are permutations of inputs to outputs that can not be realized which is why MINs are called fully reachable but blocking. This is the main disadvantage of logN-MINs.

There are two other categories of MINs, which are called Clos and Benes networks that do not belong to the logN type and that are non-blocking [19]. Unfortunately, the Clos network has a switch complexity of $(3/2)N\sqrt{N}$, and the Benes network has $N\log_2 N$ complexity which is both not the minimum logN-MINs have. This means for the applicability of Clos and Benes MINs as on-chip networks that they consume more chip area as needed, and that they need more electrical power as logN-MINs do. Both are disadvantages for VLSI integration. Furthermore, Clos and Benes networks are non-blocking only for the price of rearranging already existing internal paths through the network, which is a problem for ongoing real-time transmissions. During path rearrangement, no data transfer can take place. Finally, path

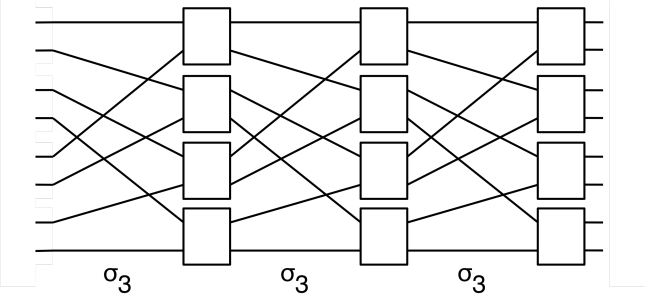


Figure 2. Omega topology of size 8x8.

rearrangements require a central control instance that executes the rearranging algorithm. However, a central control unit prevents from scalability, and the rearranging algorithm is so complex that it consumes a processor by its own, together with considerable computing time.

IV. Ideal on-chip MIN

In general, we can state that there is no ideal on-chip interconnection network with good scalability, deterministic routing latency for real-time capability, minimum chip area and minimum power consumption at the same time. However, with the introduction of FIFO buffers, each disadvantage of the above networks could be solved, at least for many practical applications. In the case of logN-MINs for example, FIFOs at all switch stages are needed as temporary storage for incoming data to hold them until the blocking situation is managed. Blocking management has to be made every time when the MPSoC requests a forbidden permutation from input to output ports. Delaying and serializing the needed transfers via FIFO accomplish this. Data is then delayed until blocking is over, and afterwards data is read out from the FIFO one after the other. A positive aspect is that blocking management can be achieved in a fully decentralized manner.

In the case of Clos and Benes MINs, FIFOs are required only at the input ports to store incoming data until the internal path rearrangements have been accomplished. If a permutation from input to output needs rearrangement, then the input FIFOs are filled while the network is drained. When all network-internal paths are empty, rearrangement can take place by setting switches newly. After that, data is let again into the network.

In both cases, the FIFO solution is not perfect because it introduces indeterministic delays in the MPSoC interprocessor communication. Depending on the filling state of a FIFO and depending on the needed transfers per time unit, more or less data frames or packets have to be temporarily stored in the FIFOs. Only by means of a fixed FIFO depth, an upper limit for the maximum latency can be stated for data delivery. However, this is sufficent in practice for many real-time applications. FIFO overflow can occur, but it is considered as a programming fault of the MPSoC. It has to be mentioned here also that is not the fault of the network but of the programmer if two input ports want to deliver data to the same output port at the same time. This is comparable to writing the same variable in a shared memory from two processors at the same time.

To summarize, the state of the art in on-chip networks is that logN-MINs are the best option because of their O(NlogN) scalability and their minimum chip and power consumption compared to busses, crossbars, Clos and Benes networks. Therefore, we propose a logN-MIN as the preferred NoC for MPSoC. In the next section we will explain which type of logN-MIN is best suited, and what we did to improve its real-time behaviour.

V. Topology of the Proposed MINoC

The network topology we decided for is known as Baseline network [20]. In Fig. 3, a Baseline network of size 16x16 is presented, together with two routing examples. This topology has been introduced in 1980 by C. Wu and T. Feng to proof equivalence among logN-MINs. The stages in the Baseline network are connected via an unshuffle wiring. The topology of the Baseline network is mathematically isomorphic to other networks of the $\log_2 N$ class but the network has technological advantages compared to other logN-MINs. The production of the Baseline network is characterized by a recursive construction. Each stage is of 1,2,4,... sub-networks of the same type. From the view-point of the first stage, the Baseline consists of one switch block of size NxN. The second stage contains two switch blocks of size (N/2) x (N/2) and so forth. That iterates down to the smallest blocks of size 2x2 as atomic elements. In addition, each stage has the minimum possible number of crossing wires, and the wires have minimum lengths [14]. Both features, recursive construction and minimum wiring, are advantages for implementation in VLSI or FPGA that are not found in other logN-MINs. In Baseline networks, the routing algorithm evaluates the most significant bit of the n=logN bits of the target address first [16]. With every bit evaluation, the interval of possible output ports is halved. After n steps, the target output port is exactly specified. This routing algorithm is a good example of the divide-and-conquer principle known from theoretical informatics. Finally, the recursive construction of the Baseline network eases its definition in VHDL. The VHDL code of a 2x2 switch is for example:

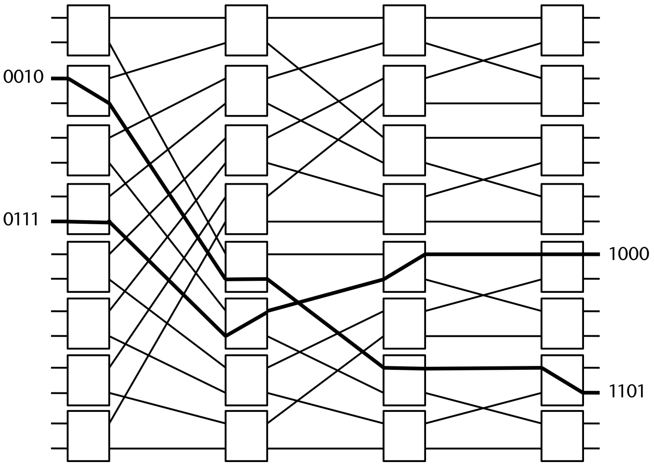


Figure 3. Baseline topology of size 16x16.

```
signal A, B, C, D: std_logic_vector(0 to 31);
shared variable S: boolean;

C <= A when S = false -- parallel connection
else B;               -- cross connection
D <= A when S = true  -- parallel connection
else B;               -- cross connection
```

where A and B are inputs, C and D are outputs and S is the switch state. Not shown are FIFO buffers, but as we have learned from practice, the FPGA synthesis of the switch controller is much more complex than the switches and the FIFO. With our preferred MINoC, we yield a MPSoC of the symmetric multiprocessor type that is depicted in Fig. 4. Its architecture includes softcore processors P, local memory M_P and shared memory M_S.

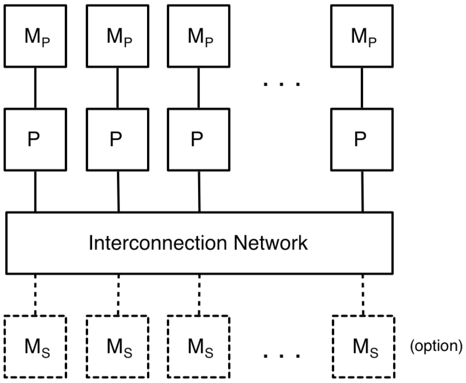


Figure 4. Block diagram of the resulting MPSoC Architecture.

With this architecture, both programming paradigms of message passing and shared memory are supported simultaneously.

VI. Implementation of the MINoC

The MINoC is implemented by adding a custom VHDL core to the existing description of a Xilinx Microblaze soft processor [21]. The block diagram of the so enhanced MicroBlaze is shown in Fig. 5.

Our custom IP core realizes the MINoC for the MPSoC. It consists of three components: 1.) switches 2.) network interface, and 3.) network controller. The first component (switches) implements the described Baseline topology. The second component is the network interface. It connects the MicroBlaze via its proprietary FSL bus [22,23] to the MINoC input ports and output ports. The third component is the network controller, which we have introduced to improve real-time behaviour. The network controller allows for interprocessor communication only in fixed points in time. This can guarantee a better upper limit for latency in data delivery.

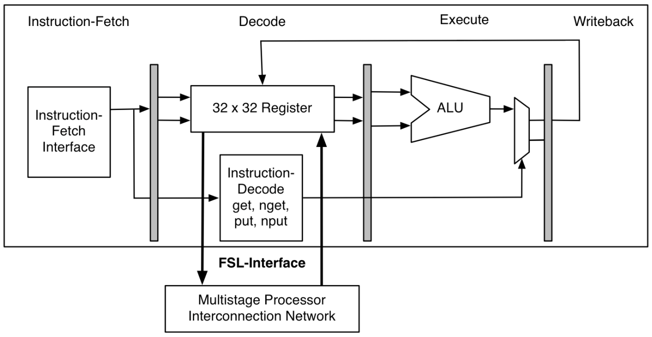

Instruction-Fetch
Decode
Execute
Writeback
Instruction-Fetch Interface
32 x 32 Register
ALU
Instruction-Decode get, nget, put, nput
FSL-Interface
Multistage Processor Interconnection Network

Figure 5. Block diagram of the soft processor enhanced by a MINoC.

A. Switches and Wiring

In the following sections of this paper we refer to message-based interprocessor communication. However, with the network coupling of shared memory (M_S) interprocessor communication via shared variables becomes feasible as well.

As seen before, the switches feature two states for direct and crossed connection paths, but for parallel computing mechanisms for task synchronization are needed also. These can be implemented by two additional switch states called upper and lower broadcast (Fig. 6). Direct and crossed connection paths are used in point-to-point communication between sender/receiver pairs via message passing. Broadcast communciation is needed for synchronous task start and stop and for distributing input data to processors. Both types of communication are required in the intended application domain of real-time embedded systems.

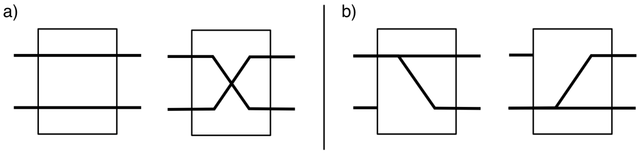


Figure 6. States of the switch: a) straight and cross b) upper and lower broadcast.

The wire patterns between the stages are implemented via bidirectional communication channels that are defined as signals in VHDL. With bidirectional channels, handshake protocols between sender and receiver are implemented.

B. Network Interface

The soft processors are connected to the interconnection network via the Fast Simplex Link (FSL) from Xilinx [23]. Each FSL interface provides an uni-directional point-to-point communication channel that includes a FIFO buffer. The FIFO buffer decouples the processor clock from the network but it introduces indeterminism as described that cannot be avoided. However, the network controller reduces the jitter in message latency during data transfer. Since the FSL interface is an internal part of the soft processor, communication takes 2 processor cycles only for transferring a 32-bit word from sender to receiver if the FIFO buffers are free. Therefore, the FSL enables a high-speed interprocessor communication. When the FSL interface is added to the soft processor, the MicroBlaze instruction set is augmented with four additional instructions:

- Blocking Read (get)
- Non-blocking Read (nget)
- Blocking Write (put)
- Non-blocking Write (nput)

These instructions are for reading from and writing data to the FIFO of the FSL interfaces. As soon as data are written to FIFO, the put instruction terminates and the NoC moves all data from source to destination FIFO. Finally, a get instruction reads the data out of the receive FIFO.

C. Network Controller

The MIN operates not by packet- or message switching but circuit switching. This provides a direct connection between sender and receiver and delivers maximum speed for interprocessor communication since no frame or packet header is required that would be overhead only. Thus all data are received in sent order.

Furthermore, a time-slot that is returning periodically is granted to each processor according to a scheduling policy where messages can be sent. The scheduling policy has been implemented in the network controller in hardware by means of VHDL. When the network controller schedules a data transfer, a communication time-slot opens, and the network controller establishes a physical path between a pair of processors. As long as the time-slot stays open, data can be sent directly from sender to receiver with full speed of the processor interfaces. After a time-slot has closed, the next path will be opened round-robin. In our tests we have used preemptive scheduling with a fixed slot time. For this purpose, a central network controller was implemented and tested. This network controller serves all connection requests from the processors. Preemption is made if a processor whose time slot has arrived does not want a data transfer through the network. The desribed scheduling policy is identical to task scheduling in real-time operating systems. The usage of a scheduling algorithm for data transfer results in better predictability to the network latency which suffers from the indeterminism of the FIFOs. Furthermore, several scheduling algorithms are possible, such as priority scheduling or earliest-deadline-first which are known from real-time operating systems.

D. Overall Architecture

The entire MPSoC including the MINoC has been implemented and tested with evaluation boards carrying FPGAs from Xilinx of the Spartan-3, Virtex-4 and Virtex-5 types. As boards have been used the ML 505 from Xilinx with a Virtex-5 FPGA, the XpressFX100 from PLDA with a Virtex-4 and the Spartan-3 starter kit board from Digilent with a Spartan-3 chip. With Spartan-3, a 4x4 network was implemented together with 4 MicroBlazes on the same FPGA. On the Virtex-4 and the Virtex-5 board, an MPSoC with a MINSoC of size 8x8 has been implemented. All processors are Xilinx MicroBlaze softcores that emulate a 32-bit RISC processor. Each soft processor features private local memory with Block RAM for instructions and data. Multiple Block RAMs are linked to processors via the Local Memory Bus (LMB) [21]. All MicroBlazes in turn are coupled to the interconnection network via the FSL-MIN interface as shown in Fig. 5. An overview of the system architecture as it was implemented is given in Fig. 7.

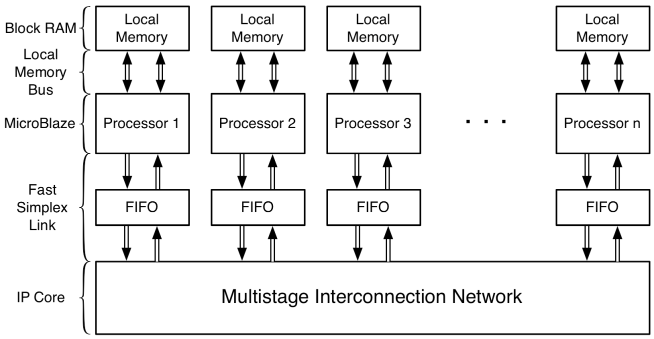


Figure 7. Overall architecture of the MPSoC.

With using the FSL processor interface we can measure that it takes two clock counts of the processor to write or read a single 32-bit data to or from the network interface. Once the connection from sender to receiver has been established no additional latency of the circuit-switched network exists, so that a network clock rate of 100 MHz induces a real data transfer rate of 1.6 Gbit/s per connection. Depending on the topology of the network multiple connections between sender/receiver pairs can be established at the same time without additional latencies. Higher network clock rates are feasible because the FSL can be operated in asynchronous mode. The maximum frequency depends on the FPGA chip that is used for implementation.

Our design studies have shown, that the network topology can be implemented in FPGA hardware without problems. The challenge is the implementation of the network controller with its routing algorithm in an efficient way. But, once the network is fully implemented in FPGA hardware, the MINoC provides a deterministic high-performance network for interprocessor communication in MPSoCs.

VII. Conclusion and Future Work

This paper proposes an on-chip multistage interconnection network with the least possible number of hardware, the minimum amount of wiring between stages and the minimum wire lengths. It can be used for high-performance interprocessor communication in real-time applications. Although logN-MINs have been already researched and used in parallel super computers, they can be adapted also for network-on-chips as well. High bandwidth and low latency are combined with a deterministic behavior of interprocessor communication in the proposed NoC. The objective is to use MPSoCs in high-performance embedded systems with hard real-time constraints that can be found in electronic control units for cars or for production machinery.

In future work we want to discuss the pros and cons of blocking and non-blocking MINs for real-time computing and their implementation in FPGA hardware. Blocking networks such as the Baseline network minimize the costs in hardware but they require a suitable scheduling strategy because not all permutations of sender/receiver pairs can be realized. Therefore further scheduling algorithms such as priority scheduling or earliest-deadline-first have to be considered for hardware implementation. In comparison with blocking networks, non-blocking networks as the Benes network can be operated without complex scheduling strategies since messages can be sent to each free receiver port at any time. On the other hand non-blocking networks exhibit extra costs in hardware plus they require complex routing algorithms due to the rearrangement of alternative connection paths.

REFERENCES

[1] A. Jerraya and W. Wolf, “Multiprocessor Systems-on-Chip,” San Francisco, CA: Morgan Kaufmann, 2005.

[2] M. Grant, “Overview of the MPSoC design challenge,” in Proceedings of the 43rd annual Design Automation Conference, San Francisco, CA, July 2006, pp. 274-279.

[3] U. Heinkel, M. Padeffke, W. Haas, and T. Buerner, “The VHDL Reference,” Cichester, England: John Wiley & Sons, 2000.

[4] C. Bobda, T. Haller, and F. Muehlbauer, “Design of Adaptive Multiprocessor on Chip Systems,” in SBCCI 2007, Rio de Janeiro, Sept. 2007, pp. 177-183.

[5] S. Aust and H. Richter, “Space Division of Processing Power For Feed Forward and Feed Back Control in Complex Production and Packaging Machinery,” in WAC 2010, Kobe, Japan, Sept. 2010

[6] P. Guerrier and A. Greiner, “A generic architecture for on-chip packet switched interconnections,” in DATE2000, March 2000, pp. 250-256.

[7] H. G. Lee, N. Chang, U. Y. Ogras, and R. Marculescu, “On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches,” in ACM Transactions on Design Automation of Electronic Systems, Vol. 12, No. 3, Article 23, August 2007.

[8] L. Benini and G. De Micheli, “Networks on chips: a new SoC paradigm,” in IEEE Computer magazine, vol. 35, no. 1, Jan. 2002, pp. 70-78.

[9] D. Atienza, F. Angiolini, S. Murali, A. Pullini, L. Benini, and G. De Micheli, “Network-on-Chip design and synthesis outlook,” in Integration, the VLSI Journal, 41(2), Feb. 2008.

[10] S. Kumar, A. Jantsch, J.-P. Soininen, M. Forsell, M. Millberg, J. Öberg, K. Tiensyrjä, and A. Hemani, “A Network on Chip Architecture and Design Methodology,” in Proceedings of the IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2002, pp. 105–112.

[11] T. Bjerregaard and S. Mahadevan, “A Survey of Research and Practices of Network-on-Chip,” in Integrated Circuits and Systems Design (SBCCI) 2003, pp. 169-174.

[12] C. A. Zeferino and A. A. Susin, “SoCIN: A Parametric and Scalable Network-on-Chip,” in ACM Computing Surveys (CSUR), vol. 38, issue 1, 2006.

[13] L. Benini and G. De Micheli, “Networks on Chips: Technology and Tools,” San Francisco, CA: Morgan Kaufmann, 2006.

[14] H. Richter, US-Patent 5,175,539 ”Interconnection Network”.

[15] D. H. Lawrie, “Access and Alignment of Data in an Array Processor,” in IEEE Transactions on Computers, vol. C-24, no. 12, December 1975, pp. 1145-1155.

[16] H. Richter, “Verbindungsnetzwerke für parallele und verteilte Systeme,” in German, Heidelberg: Spektrum, 1997.

[17] J. L. Hennessy and D. A. Patterson, “Computer Architecture. A Quantitve Approach,” 4th edition, San Francisco, CA: Morgan Kaufmann, 2007.

[18] J. Duato, S. Yalamanchili, and L. Ni, “Interconnection Networks. An Engineering Approach,” San Francisco, CA: Morgan Kaufmann, 2003.

[19] W. Dally and B. Towles, “Principles and Practices of Interconnection Networks,” San Francisco, CA: Morgan Kaufmann Publishers Inc., 2003.

[20] C.-L. Wu and T.-Y. Feng, “On a Class of Multistage Interconnection Networks,” in IEEE Transactions on Computers, Vol. C-29, No. 8, August 1980, pp. 694-702.

[21] Xilinx Inc., “MicroBlaze Processor Reference Guide,” Product Specification: UG081, v9.0, Jan. 2008

[22] Xilinx Inc., “Fast Simplex Link (FSL) Bus (v2.11b),” Product Specification: DS449, June 2009

[23] H. P. Rosinger, “Connecting Customized IP to the MicroBlaze Soft Processor Using the Fast Simplex Link (FSL) Channel,” Xilinx Application Note: XAPP529, v1.3, May 2004

Dynamic Local Search Algorithm for Solving Traveling Salesman Problem

Kambiz Shojaee Ghandeshtani
Low-Power High-Performance Nanosystems Lab.
University of Tehran
Tehran, Iran
E-mail: k.shojaee@ece.ut.ac.ir

Mojtaba Behnam Taghadosi
Mechatronics Laboratory (LIM)
Politecnico di Torino
Torino, Italy
E-mail: mojtaba.behnam@polito.it

Seyed Mohammad Hossein Seyedkashi
Dept. of Mechanical Engineering
Tarbiat Modares University
Tehran, Iran
E-mail: seyedkashi@modares.ac.ir

Keyvan Shojaii
Dept. of Electrical Engineering
Sadjad Institute of Higher Education
Mashhad, Iran
E-mail: keyvan_shojaii@yahoo.com

***Abstract*— In this paper, developing a new local search approach based on 2-Opt operator and its implementation for TSP solution in SA algorithm (as a global search algorithm) is purposed. It is shown that more favorable results are expected by meaningful correlation between local search approach and global search algorithm in annealing process. In order to compare the performance of the proposed operator with the 2-Opt as a basic operator, 24 benchmarks of TSP is selected from TSPLIB and both algorithms are implemented for 20 times for solving these benchmarks. The results show the improvement of error average for about 27%.**

Keywords- TSP; Local search; 2-Opt; Global search; Simulated annealing

I. Introduction

Traveling Salesman Problem is the problem of searching the shortest closed route (shortest Hamiltonian cycle) among N cities, the cities which the traveling salesman has passed one and only one time and in the end has returned to the start point. TSP problem is one of the combinatorial optimization problems and includes all aspects of a combinatorial optimization problem in addition to a very simple definition. Needed time to solve this problem using algebraic algorithms is a non-polynomial function of the problem size [1]. This is why this problem is also categorized in NP-complete problems. Traveling sales man problem has many practical applications in science and engineering such as vehicle routing, integrated circuits design, automated guided vehicles scheduling, robot control and etc.

In recent decades, many researchers have tried to solve TSP problem using metaheuristic methods such as Neural Network (NN) [2-4], Simulated Annealing (SA) [5-12], Genetic Algorithm (GA) [13-15], Tabu Search (TS) [16-18], Ant Colony Optimization (ACO) [19-21], and Particle Swarm Optimization (PSO) [22-23]. Also, integration of these algorithms is widely used, i.e., integration of Simulated Annealing and Genetic Algorithm [6, 11], Simulated Annealing and Neural Network [7] and Simulated Annealing and Particle Swarm Optimization algorithm [23].

In this paper, solving the traveling salesman problem using modified 2-Opt [25] operator in Simulated Annealing algorithm is proposed. Consequently, a new operator for local searching is proposed. So in Section 2, basic simulated annealing algorithm is defined. In Section 3, with redefinition of 2-Opt operator, a dynamic operator is introduced according to the existing conditions in simulated annealing algorithm. Simulation results and their comparison on presented benchmarks in TSPLIB site [25] are presented in Section 4 and the authors' suggestions and conclusion are given in the final section.

II. Basic Simulated Annealing Algorithm

The idea of simulated annealing was first presented by Nicholas Metropolis in 1953 as a modified Monte Carlo integration method [26]. He resembled the paper to the material which is made by cooling after heating of them. Simulated annealing for integrated optimization applications such as Travelling Salesman Problem was introduced the first time by Kirkpatrick et al. [5] in 1983 inspired from Metropolis algorithm. This algorithm is adapted from the cooling process which metal is heated to its melting point and then cools gradually. This temperature decrease is such that the system is approximately in thermodynamic equilibrium. During the process of gradual reduction of temperature, system becomes more regular and moves toward steady state with minimum energy. The main Metropolis scheme in determination of temperature and the initial energy state of thermodynamic system is that if the energy changes are negative, new structure (energy and temperature) will be accepted but if the energy changes are positive, the acceptation is subjected to the Boltzmann distribution function with $\exp(-\Delta E/\kappa_B T)$ in which κ_B is Boltzmann's constant with positive value [27]. The whole process will be repeated until the energy is minimized and the system reaches a steady state. This algorithm is suitable for solving mixed discrete problem and complicated

ISBN: 978-1-61208-101-4

nonlinear problem. In simulated annealing algorithm, cooling schedule parameters have role of process controlling in the search algorithm. Cooling schedule has three parameters which are:

1) Initial temperature (T_0).

2) Convergence criterion or Freezing temperature (T_F).

3) Cooling function.

In this algorithm, if initial and final temperature are appropriately defined and temperature reduction is selected so that the slope of temperature reduction curve is less than the slope of $T_{K+1}=T_0/(1+\log(k+1))$, then simulated annealing algorithm will converge to the absolute minimum when the number of iterations (k) tends to infinity [28] but the temperature reduction based on this slope requires a lot of computational times. So faster cooling functions are used such as $T_{K+1}=\alpha.T_K$ in which $0.8<\alpha<1$ or $T_{K+1}=T_K/(1+\log(k+1))$. The number of temperature change steps from melting temperature to freezing point will have a considerable reduction using these functions and subsequently the probability of passing through suitable temperature range for optimum search will also decrease.

Therefore, in each temperature, it is tried to give sufficient search time to the algorithm in suitable temperature range by defining several iterations in inner search loop including new generation, evaluation and decision making. In the algorithm, the number of repetitive frequencies in a constant temperature is named Markov Chain Length. What distinguishes this algorithm in discrete or continuous problems is how it generates a new generation based on the current generation. In continuous problems, some definitions such as neighborhood radius are used for production of new generation in a neighborhood of the current generation, so that the next generation will be produced by determination of random changes around the variables of the current generation with value of the neighborhood radius. In simulated annealing process, neighborhood radius is reduced proportional to temperature reduction in order to increase convergence speed. The same production of new generation is performed in discrete problems using some operators which generate the next generation implicit in a radius of the neighborhood of current generation. These operators are also called Move Set. Some of the effective operators in generation of discrete problems such as Traveling salesman are the following. In a TSP problem we need a means of representing the tour. Each tour can be described by a permuted list of the numbers 1 to N, which represents the cities in TSP.

1) Switching: Randomly selects two nodes from tour and replaces with each other.

2) Translation: randomly selects a portion of a tour and enters between another randomly selected node.

3) Inversion: Or 2-Opt which is a state of k-Opt operator. In 2-Opt move, the tour is broken into 2 parts, then the 2 parts are reconnects in the other possible way.

4) Lin-Kernigan, which is a kind of variable-Opt, was presented in 1973 [29] and many researchers have tried for efficient implementation of this operator. One of the most efficient LK operators is proposed by Helsguan [30], which employs a number of important innovations including sequential 5-opt moves and the use of sensitivity analysis to direct the search.

In fact, these operators are used as local search approaches in global search approaches such as Tabu search, Simulated annealing and Genetic algorithm.

III. Dynamic 2-Opt

In this paper, a new operator inspired by 2-Opt and the neighborhood radius concept in generation of continuous problems is developed. 2-Opt is used as the base operator in definition of this operator but in this new definition, the operator's behavior will change dynamically according to the behavior of algorithm in search process. Kirkpatrick et al. [5] have emphasized that simulated annealing algorithm will show a more efficient behavior in its intelligent search process in the temperature range of the annealing process, called as intermediate temperatures. With this explanation, it will be seen in TSP problem solving that the algorithm is not able to do direct search in initial stages of algorithm and initial temperatures, and will make mistake in its orientation. But over time and its entrance to the algorithm intermediate temperature range, it will have a suitable orientation for achieving the global minimum in addition to have the probability of passing through local minimums and hill-climbing ability. After temperature reduction and passing through intermediate range it will be seen that in the simulated annealing algorithm is only able to perform minor changes in TSP problem solving to improve goal function. In definition of the new operator, different steps of the search algorithm has been considered and change ranges of the 2-Opt operator is restricted according to each step and proportional to its requirements.

In this method, 2-Opt operator starts dynamically from its local search behavior at the beginning of the algorithm and with reduction of effective amplitude in its inversing operation performs a better search according to the algorithm progress and temperature reduction in comparison with its normal operation. In fact, the idea of such definition from local search operator of SA algorithm is how this algorithm converges to the absolute minimum in its search process. In SA algorithm, hilling up possibility is reduced by passing time. In fact searching with long steps in the search space of a problem has less chance for acceptance. Thus, acceptance chance and convergence to the improved results have been provided by reduction of inversion amplitude in 2-Opt operator.

In definition of 2-Opt operator, two random numbers (i, j) are produced which their generation amplitude is the number of cities in TSP. then the tour sequence is reversed between these 2 nodes. In fact, in Dynamic 2-Opt both nodes are not selected randomly. i is a random number and j is a random number in neighborhood radius of i. Using this technique, both indices for reversing operation will have correlation with each other in addition to randomly selection.

This correlation quantity increases with temperature reduction. At the beginning of the SA algorithm when the temperature is high, neighborhood radius is equal to half number of cities (N/2) in Traveling Salesman problem and the behavior of 2-Opt operator is like normal condition.

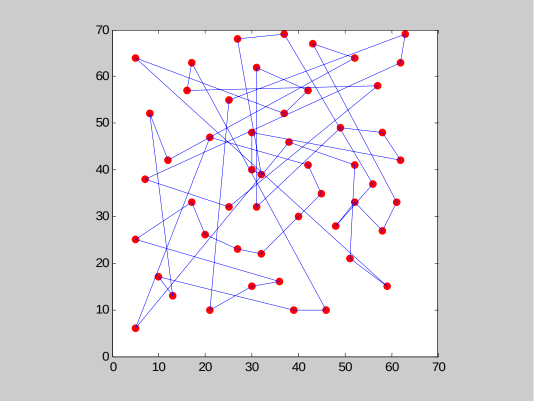


Figure 1. Sample Tour for *eil*51 at the beginning of SA algorithm

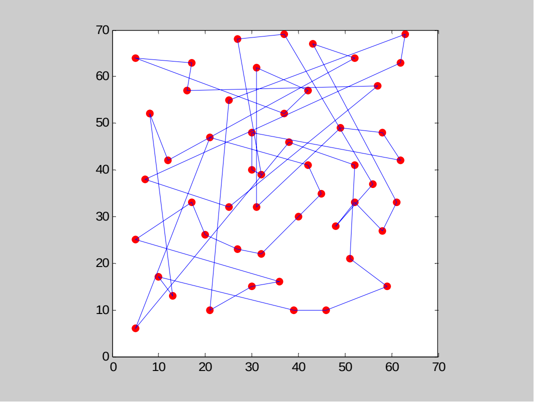


Figure 2. Improved by Dynamic 2-Opt in one operation

But with temperature decrease in next steps, this neighborhood radius will reduce with a multiplication of 0.9. Dynamic 2-Opt operator Pseudo-Matlab code is illustrated as follows.

```
NewTour =  Tour;
i = round(rand*N + 0.5);
j = round((rem((rand*(N/2)*NR + i) , N)) + 0.5);
2-Opt_Index = (min([ i ; j]):max([ i ; j]));
NewTour(2-Opt_Index) = fliplr(Tour(2-Opt_Index));
```

Where:

N = Number of cities in Traveling Salesman problem.

NR = Neighborhood Radius.

With this method in lower temperatures *j* will be generated in nearer radius to *i* and will have a narrower search space. Therefore, the operator's behavior will be dependent to the algorithm's temperature and will change dynamically during the search process based on the algorithm‘s condition. In other words, with this operator a correlation is developed between local search algorithm and global search algorithm which will result in a more intellectual search.

For instance, Figure 1 illustrates *eil*51 TSP benchmark. In this figure there is a Tour at the beginning of the SA algorithm. Figure 2 shows the results of Dynamic 2-Opt when operates in Tour which is shown in Figure 1. By this move set, the tour length is improved by 10% in one operation (Tour length in Figure 1 is 1123, which is improved to 1016 in Figure 2).

Figure 3 shows a sample condition near the end of SA algorithm which is improved by Dynamic 2-Opt in Figure 4. In these four figures, we explain the requirement of SA algorithm to improve the tours according to the algorithm's progress. In other words, the SA algorithm needs to have a long step in 2-Opt operator to improve the search results but during the algorithm progress it is required to reduce the neighborhood radius in generation of 2-opt index in order to improve the local cross. At the beginning of search process, when the neighborhood radius is N/2, the possible amplitude for generation of *j* is such that all other nodes are possible to be selected. But by passing time in search process, the amplitude of generation of node *j* (second selection) will decrease proportional to temperature reduction and will make inversion possible in smaller range. This matter is provable in discussion of the algorithm's behavior in TSP problem solving in such a way that at the beginning of the algorithm, the inversion operation with wide change ranges is efficient in passing through local minimums and proper orientation in optimal tour selection and also the reduction in inversion change range in operator will provide the possibility of minor changes at the end of the process.

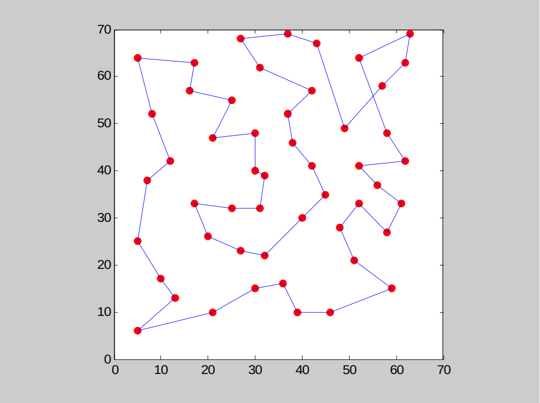


Figure 3. Sample Tour for *eil*51 near the end of SA algorithm

The results obtained by this operator are compared with 2-Opt normal performance in the next section, which declares the acceptable performance of this operator in SA algorithm.

IV. Result and Comparison

In this section, performance of two operators "2-Opt" and "Dynamic 2-Opt" is compared in simulated annealing

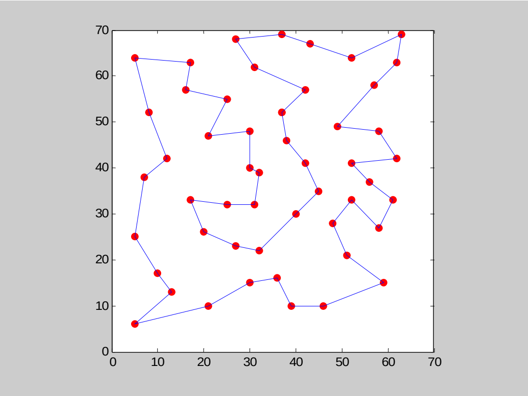


Figure 4. Figure 3 improved by Dynamic 2-Opt in one operation

algorithm process under the same conditions. The initial temperature is defined so that the initial acceptance rate in a first Markov chain is about 50%. Final temperature is defined in a condition that the acceptance condition in internal loop will not be concluded for any variable during two consecutive Markov Chain. Also the cooling function $T_{k+1}=\alpha*T_k$ is used in this algorithm so that α=0.9. The proposed algorithm is implemented for two operators of 2-Opt and Dynamic 2-Opt for 24 benchmarks listed in TSPLIB [46] for 20 times and the results are compared in Table I with [7] and [9].

It is quite easy to realize that using the new operator (Dynamic 2-Opt) has improved performance of SA significantly. The optimal values given by the TSBLIB site, for each case are listed in the second column of the table. We have compared the best, the worst and the average of the error in the results obtained by the new approach with other results given by other works (if the best and/or the worst cases are available). The error percentage is calculated by:

$$\delta = 100\,(E - E^*) / E^*$$

where E* is the optimal (minimum) energy.

The first method chosen for comparison is the Constructive Optimizing Neural Network (CONN) proposed in [2], for which it is claimed that all runs has led to the same results, so that the best, the worst and the average of the solutions are the same. We have compared our results with the best and the average error percentages of the results given in [4] for its memetic neural network.

Table I demonstrates an enhancement in the results of SA algorithm implemented by "Dynamic 2-Opt" operator rather than regular "2-Opt" operator. As it's clear to see, the results of implementing the "Dynamic 2-Opt", has gotten 0.35 percentage improvement in the average error of the best results, 0.49 percentage improvement in the average error of the average results and finally 1.1 percentage improvement in the average error of the worst results.

Also Table I states that the results of SA algorithm with regular 2-Opt operator have obtained 2.14 percentage improvement in the average error of the average results (for 24 benchmarks) in comparison with CONN method, which indicates the high ability of SA algorithm for solving these kind of problems. As well, the 2.71 percentage improvement in implementing the SA algorithm with Dynamic 2-Opt operator toward CONN method, predicates the possibility of enhancement in SA algorithm.

As well, Table I includes the comparison between "memetic neural network" results [4] and SA algorithm results implemented by "2-Opt" and "Dynamic 2-Opt" operators that respectively indicates 1.41 and 1.88 percentage improvements in the average error of the best results and 1.76 and 2.41 percentage improvements in the average error of the average results.

To accomplish our comparison, we have added another set of methods from [11], in which 11 methods are run on 24 benchmarks from lin105 to rat783. For brevity purpose, the problems are categorized into 3 groups, namely: small, medium and large size benchmarks. The results are given in Table II, where the average of the average error in each group is shown. For detailed explanation of each method see [11].

In [10], the result of ABD (Annealed Bounded Demon) algorithm is better than the results of other SA algorithm's family. In this paper, the SA algorithm with implementing "2-Opt" and "Dynamic 2-Opt" operators respectively has obtained 1.05 and 1.28 percentage improvements in the average error of the average results for small size problems.

In medium size problems, the SA algorithm with "2-Opt" operator shows weaker results than ABD algorithm (0.41 percentage error more) but with using the "Dynamic 2-Opt" operator, the amount of improvement in the average error of the average results has reached 0.44 percentage that it's an evidence of good performance of proposed algorithm.

For solving this problem a PIV computer with 1.8GHz CPU and 512MB RAM is used in MATLAB7.7 environment.

As it shown in Table II, with definition of Dynamic 2-Opt the average of SA algorithm results are improved 27% which shows the effect of the redefinition of 2-opt operator in this paper on efficiency of SA algorithm.

V. Conclusion

In this paper, a new definition of 2-Opt operator was presented, which will result in correlation of local search approach and global search algorithm. Using Simulated Annealing algorithm and proportional with temperature reduction in this algorithm, a new operator is designed for generation of new generation in neighborhood of current generation, so that its operation is variable during search process and will orient the local search method according to temperature reduction and the algorithm's correlation. This algorithm is implemented on 24 benchmarks of TSPLIB site for 20 times and its results are categorized in order to be compared with the base algorithm and other algorithms' results. The obtained results show the improvement of SA algorithm efficiency up to 27% which proved the performance of this operator.

Application of this operator in global search algorithms such as Ant colony or Genetic algorithm may have a good effect on their efficiency. Also since recent researches are focused on integrated metaheuristic methods, an integration of this method with others may result in better conclusions.

TABLE I. COMPARISON BETWEEN D2-OPT AND OTHER METHODS FOR 24 BENCHMARKS (THE AVERAGE Δ OF THE AVERAGE ERROR IN 20 RUNS)

TSP Benchmark	Optimal Solution	With 2-Opt			With Dynamic 2-Opt			CONN [7]			Memetic neural network [9]	
		Best δ	Average δ	Worst δ	Best δ	Average δ	Worst δ	Best δ	Average δ	Worst δ	Best δ	Average δ
lin105	14379	0	0.9	2.55	0	0.18	0.87	0.38	0.38	0.38	0	0.34
pr107	44303	0.05	0.51	1.51	0	0.3	0.71	2.77	2.77	2.77	0.14	0.67
pr124	59030	0	0.68	2.22	0	0.29	0.88	1.74	1.74	1.74	0.26	1.52
pr136	96772	0.67	1.98	3.53	0.37	1.42	2.33	2.27	2.27	2.27	0.73	3.1
pr144	58537	0	0.94	4.02	0	0.93	4.15	2.34	2.34	2.34	-	-
pr152	73682	0.21	0.85	2.15	0	0.91	3.93	0.79	0.79	0.79	1.57	2.6
u159	42080	0	1.62	3.8	0	2.38	6.34	-	-	-	-	-
rat195	2323	1.76	3.18	4.69	0.6	2.04	3.18	5.64	5.64	5.64	4.69	6.89
d198	15780	0.2	0.86	1.66	0.37	1.22	2.64	4.16	4.16	4.16	-	-
pr226	80369	0.99	1.68	4.62	0.58	1.31	1.94	1.93	1.93	1.93	-	-
gil262	2378	1.35	2.34	3.24	0.92	1.98	2.94	-	-	-	-	-
pr264	49135	0.54	2.3	5.45	0	2.11	5.62	3.58	3.58	3.58	-	-
pr299	48191	0.5	1.92	4.32	0.6	1.69	3.92	4.8	4.8	4.8	-	-
lin318	42029	1.32	2.77	4.23	1.2	2.45	3.05	-	-	-	3.63	5.51
rd400	15281	1.26	2.66	4.25	0.84	2.13	3.30	5.77	5.77	5.77	-	-
pr439	107217	1.06	3.44	7.31	1.05	1.78	3.24	6.03	6.03	6.03	-	-
Pcb442	50778	2.06	4.68	7.00	1.86	2.81	3.68	5.77	5.77	5.77	3.57	6.08
d493	35002	1.58	2.47	5.07	1.21	2.10	2.82	5.83	5.83	5.83	-	-
u574	36905	1.86	2.75	4.46	1.40	2.04	3.03	5.90	5.90	5.90	4.09	5.08
rat575	6773	3.47	3.94	6.84	1.89	2.91	4.08	6.72	6.72	6.72	4.31	5.47
p654	34643	0.67	1.81	5.94	0.80	1.70	3.48	4.13	4.13	4.13	2.51	5.13
d657	48912	2.59	3.36	4.15	1.82	2.50	3.32	7.58	7.58	7.58	3.97	5.02
u724	41910	3.16	3.62	4.07	1.93	2.44	3.11	6.97	6.97	6.97	4.64	5.36
rat783	8806	2.12	3.05	5.59	1.45	2.87	3.70	7.59	7.59	7.59	5.46	5.95
Average		1.14	2.26	4.28	0.79	1.77	3.18	4.41	4.41	4.41	2.83	4.19
With 2-Opt		-	-	-	1.14	2.26	4.28	1.18	2.27	4.35	1.42	2.43
With D2-Opt		0.79	1.77	3.18	-	-	-	0.80	1.70	3.04	0.95	1.78

ACKNOWLEDGMENT

This work is supported by Nano-Age Technology Group in Mashhad, Iran (www.nanoage.ir).

REFERENCES

[1] C. H. Papadimitriou, "The Euclidean Traveling Salesman Problem is NP-complete", Theoretical Computer Science, 4(3): 237-244, 1977.

[2] M. Saadatmand-Tarzjan, M. Khademi, M. R. Akbarzadeh-T., and H. A. Moghaddam, "A Novel Constructive-Optimizer Neural Network for the Traveling Salesman Problem", IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 37, No. 4, pp.754-770 Aug. 2007.

[3] Sitao Wu and T. W. S. Chow, "Self-Organizing and Self-Evolving Neurons: A New Neural Network for Optimization", IEEE Transaction on Neural Networks, Vol. 18, No. 2, pp. 385-396, Mar. 2007.

[4] J. C. Creput and A. Koukam, “A memetic neural network for the Euclidean traveling salesman problem”, Neurocomputing Accepted 22 Jan. 2008.

[5] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by Simulated Annealing", SCIENCE, Volume 220, Number 4598, pp. 671-680, 13 May 1983.

[6] F. T. Lin, C. Y. Kao, and C. C. Hsu, "Applying the Genetic Approach to Simulated Annealing in Solving Some NP-Hard Problems", IEEE Transaction on Systems, Man and Cybernetics, Vol. 23. No. 6, pp. 1752-1767, Nov./Dec. 1993.

[7] L. Wang, S. Li, F. Tian, and X. Fu, "A Noisy Chaotic Neural Network for Solving Combinatorial Optimization Problems: Stochastic Chaotic Simulated Annealing", IEEE Transaction on Systems, Man and Cybernetics, Part B:Cybernetics, Vol. 34, No.5, pp. 2119-2125, 2004.

TABLE II. COMPARISON OF D2-OPT WITH OTHER METHODS GIVEN IN [10]
(THE AVERAGE OF THE AVERAGE ERROR IN 20 RUNS)

Algorithms in [10]	Average Error in Small Size			Average Error in Medium Size		
SA (Simulated Annealing)	2.76			3.25		
TA (Threshold Accepting)	5.37			4.18		
RRT (Record-to-Record Travel)	4.22			6.79		
BD (Bounded Demon)	5.26			4.44		
RBD (Randomized Bounded Demon)	4.33			9.38		
AD (Annealed Demon)	3.24			3.27		
RAD (Randomized Annealed Demon)	2.82			4.38		
ABD (Annealed Bounded Demon)	2.65			2.77		
RABD (Randomized Annealed Bounded Demon)	2.63			3.64		
ADH (Annealed Demon Hybrid)	2.97			2.95		
ABDH (Annealed Bounded Demon Hybrid)	2.69			2.89		
MSSA (the proposed method)	Min.	Ave.	Max.	Min.	Ave.	Ma x
With 2-Opt	0.54	1.60	3.42	1.98	3.18	5.47
With D2-Opt	0.33	1.37	3.04	1.43	2.33	3.38

[8] L. Wang, S. Li, F. Tian, and X. Fu, "A Noisy Chaotic Neural Network for Solving Combinatorial Optimization Problems: Stochastic Chaotic Simulated Annealing", IEEE Transaction on Systems, Man and Cybernetics, Part B: Cybernetics, Vol. 34, No. 5, pp. 2119-2125, Oct. 2004

[9] S. Andrew, "Parallel N-ary Speculative Computation of Simulated Annealing", IEEE Transaction on Parallel and Distributed Systems, Vol. 6, No. 1O, pp. 997-1005, Oct. 1995.

[10] D. C. W. Pao, S. P. Lam, and A. S. Fong, "Parallel implementation of simulated annealing using transaction processing", IEE Proc-Comput. Digit. Tech.. Vol. 146, No. 2, pp. 107-113, March 1999.

[11] J. W. Pepper, B. L. Golden, and E. A. Wasil, "Solving the Traveling Salesman Problem With Annealing-Based Heuristics: A Computational Study", IEEE Transaction on Systems, Man and Cybernetics —Part A: Systems and Humans, Vol. 32, No. 1, pp. 72-77, Jan. 2002.

[12] H. Chen, N. S. Flann, and D. W. Watson, "Parallel Genetic Simulated Annealing: A Massively Parallel SIMD Algorithm", IEEE Transaction on Parallel and Distributed Systems, Vol. 9, No. 2, pp.126-136, Feb. 1998.

[13] H. Shakouri G., K. Shojaee, and M. Behnam T., "Investigation on the choice of the initial temperature in the Simulated Annealing: A Mushy State SA for TSP", 17th Mediterranean Conference On Control And Automation, Thessaloniki, Greece, 24-26 June 2009.

[14] G. Magyar, M. Johnsson, and O. Nevalainen, "An Adaptive Hybrid Genetic Algorithm for the Three-Matching Problem", IEEE Transaction on Evolutionary Computation, Vol. 4, No. 2, pp. 135-146, Jul. 2000.

[15] C. H. Cheng, W. K. Lee, and K. F. Wong, "A Genetic Algorithm-Based Clustering Approach for Database Partitioning", IEEE Transaction on Systems, Man and Cybernetics —Part C: Applications and Reviews, Vol. 32, No. 3, pp. 215-230, Aug. 2002.

[16] H. D. Nguyen, I. Yoshihara, K. Yamamori, and M. Yasunaga, "Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems", IEEE Transaction on Systems, Man and Cybernetics —Part B: Cybernetics, Vol. 37, No. 1, pp. 92-99, Feb. 2007.

[17] F. Glover, "Tabu Search Fundamentals and Uses", Graduate School of Business, University of Colorado, Boulder, 1995.

[18] Y. Peng, B. H. Soong, and L.P. Wang, "Broadcast scheduling in packet radio networks using a mixed tabugreedy algorithm", Electronics Letts., vol.40, no.6, pp.375-376, Mar., 2004.

[19] A. Misevˇcius, "Using iterated tabu search for the traveling salesman problem," Informacin˙es Technologijos ir Valdymas, vol. 3, no. 32, pp. 29–40, 2004.

[20] M. Dorigo, Member, ZEEE, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents", IEEE Transaction on Systems, Man and Cybernetics —Part B: Cybernetics, Vol 26, No. 1, pp. 29-41, Feb. 1996.

[21] M. Dorigo, Senior, and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", IEEE Transaction on Evolutionary Computation, Vol. 1, No. 1, pp. 53-66 Apr. 1997.

[22] L. Wang and Q. Zhu, "An Efficient Approach for Solving TSP: the Rapidly Convergent Ant Colony Algorithm", Fourth International Conference on Natural Computation pp. 448-452, 2008.

[23] X. H. Shi , Y. C. Liang, H. P. Lee, C. Lu, and Q. X. Wang, "Particle swarm optimization-based algorithms for TSP and generalized TSP", Information Processing Letters, No. 103, pp. 169–176, 2007.

[24] H. Shakouri G., K. Shojaee, and H. Zahedi, "An Effective Particle Swarm Optimization Algorithm Embedded in SA to solve the Traveling Salesman Problem", 21sh Chinese Control and Decision Conference (CCDC09), Guilin, China, 17-19, Jun. 2009.

[25] D. Johnson and L. McGeoch, "The Traveling Salesman Problem: A Case Study in Local Optimization" chapter of "Local Search in Combinatorial Optimization", pp. 215-310, London 1997.

[26] G. Reinelt. Tsplib95, 1995. Available at: http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95.

[27] N. Metropolics, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," J. Chem. Phy, vol.21, pp. 1087-1092, 1953.

[28] V. Cerny, "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm," J. Opt. Theory Appl, vol.45, pp. 41-51, 1985.

[29] E. Aarts and J. Korst, "Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing", New York: Wiley, 1989.

[30] S. Lin and B. Kernighan, "An effective heuristic algorithm for the traveling salesman problem," Oper. Res., vol. 21, no. 4598, pp. 498–516, 1973.

[31] K. Helsgaun, "An effective implementation of the Lin–Kernighan traveling salesman heuristic," European Journal of Operational Research, vol. 126, no. 1, pp. 106-130, 2000.

Computational Models for Estimating Liver Iron Overload with the Magnetic Iron Detector

Barbara Gianesin
Istituto Nazionale di Fisica Nucleare (INFN)
Genova, Italy
Email: gianesin@ge.infn.it

Luca Baldassarre
Computer Science Department
Università di Genova
Genova, Italy
Email: baldassarre@disi.unige.it

***Abstract*—An accurate measurement of the liver iron overload is essential for the management of diseases such as thalassemia and hemocromathosis. The Magnetic Iron Detector is a susceptometer, which measures the total iron overload in the liver and has been used on more than 800 patients of Galliera Hospital (Genoa, Italy) since February 2005. The iron overload is obtained by calculating the difference between the measured magnetization signal and the patient's background signal, which is the magnetization signal that would be measured for that patient with a normal iron content. This study describes two models for calculating the background signal using the measurements and the anthropometric features of 84 healthy volunteers. The first model introduces a statistical correction to the signal computed from the body shape of the subject assuming it to be made of water. The second model is based on statistical learning and learns from the volunteers' data a mapping from the anthropometric features to the background signal. We present two approaches to combine the models. The assessment of the models on the 84 volunteers show that the performances of the models are comparable and that we can confidently estimate the background signals of patients. The model sensitivity (0.9 g) allows the physicians to monitor the iron overload variations due to the therapy. These models are currently in use at the Galliera Hospital.**

***Keywords*-Medical Computation; Liver Iron Overload; Magnetic Iron Detector; Thalassemia; Statistical Learning; Kernel Methods; Forward Problem.**

I. Introduction

An accurate assessment of body iron accumulation is essential for the diagnosis and therapy of iron overload in diseases such as hereditary hemochromatosis, thalassemia and other forms of severe congenital or acquired anaemia. For example, in hereditary hemochromatosis, the subject adsorbs an excess of iron from the diet every day, while in thalassemia major the iron overload is caused by the frequent blood transfusions administrated to the patient to contrast its anaemia. Being toxic, the iron in excess must be removed by a tuned therapy: for this reason hemochromatosis patients are subjected to phlebotomy therapy while a chelation therapy is administrated to transfusion dependent patients.

The liver is the target organ for evaluation of the iron overload. A normal liver contains about 0.5 g of iron [1], whereas the overload can be higher than 10 g in severe iron-overload states. The liver biopsy is considered the gold standard to evaluate liver iron overload [1], [2]: this invasive measure evaluates the iron concentration in a small sample of hepatic tissue. Validated non-invasive techniques are MRI [3] and SQUID-susceptometer [4].

The Magnetic Iron Detector (MID) susceptometer [5], [6], [7], [8] quantifies the amount of iron in the liver by measuring the susceptibility of the human body. A magnetic field B, applied to the body, induces a magnetization of its tissues. Because the iron deposits in the living biological tissues exhibit paramagnetic behavior [7], [9], the change of the field we are interested in is very small ($\sim 10^{-6}B$ for a normal iron level). Since the MID generates a low frequency magnetic field, the measurement is performed with a pickup coil using the synchronous detection. The apparatus of the MID susceptometer is sketched in Figure 1A. Two pickup coils are assembled symmetrically with respect to the magnet in order to cancel the common mode signal induced with no patient is placed between the magnet and the lower pickup. The signal becomes non-zero when the patient is placed in the measurement area.

Since February 2005, the MID is in use at the Galliera Hospital of Genoa and more than 1300 iron overload evaluations have been performed [7]. MID obtains the iron overload by computing the difference between the measured magnetization signal of the patient and its *background*

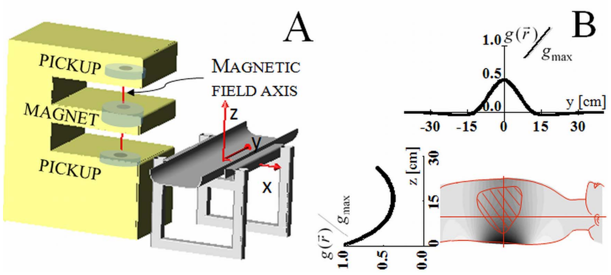


Figure 1. MID Instrumentation. (A) Two pickups are placed symmetrically with respect to the magnet: the sum of their signals is zero in the absence of a patient. (B) Weight function $g(\vec{r})$ of the MID (see Section III-A).

signal, defined as the magnetization signal that the patient would generate with a normal iron level. The evaluation of the background signal is performed under the hypothesis that the magnetization signal of a patient, without iron overload, is the same as the one of a volunteer with the same anthropometric characteristics.

Until January 2010, the background signal was calculated by a statistical model [7], that was developed exploiting the correlation coefficient between the measured magnetization signal of volunteers (i.e., subjects with a normal iron level) and the anthropometric variables. Moreover, a linear dependence of the magnetization signal from the input variables of the model was assumed. The sensitivity of this model is about 1 g, which must be compared with the reproducibility of the iron overload measurement of the same patients (less than 0.5 g) and with the iron contents of a healthy liver (about 0.5 g).

Two models have been developed in order to improve the MID sensitivity. Each one were the object of two PhD theses [8], [10]. The first model was developed following [11], in which the direct calculation of the magnetization signal was proposed assuming for the body a uniform magnetic susceptibility distribution equal to that of water. The second model is based on statistical learning theory [10], [12].

This paper is devoted to the description and the comparison between the two models. Moreover, we propose two different methods for combining their results. The performances of the models, evaluated on a common set of 84 healthy volunteers, are quite equivalent, each one introducing a benefit in the background signal calculation. From January 2010 both this models are being used at the Galliera Hospital (Genoa, Italy) for assessing the iron overload with MID.

The paper is organized as follows: in Section II we describe the measurement of the magnetic signal and the anthropometric features recorded for each subject. In Section III we first detail the two models developed to estimate the background signal and then two approaches adopted to combine their predictions; secondly we describe the experimental protocol adopted to compare them. In Section IV we present the experimental results and conclude the paper with the discussion in Section V.

II. The Measurements

Here we describe the measurement of the magnetization signal and the anthropometric features recorded for each subject.

A. The Magnetization Signal

During the measurement, the patient is placed inside the measurement area and the patient's position along the stretcher is such that the Y axis, positively oriented towards the head, lays along the longitudinal symmetry axis of the body (Figure 1A). The center of the patient's trunk falls on the X axis origin and the abscissa of the liver center of mass is negative. As the stretcher moves along the rails, the vertical axis of the magnetic field slides along the X axis, allowing us to scan the whole liver region. Figure 2A reports the magnetization signals of an anthropomorphic plastic phantom dosed with paramagnetic powder, equivalent to 6.5 g and 30 g of iron. Figure 2B shows the comparison between the signal from a patient with an iron overload of about 9 g in the liver and that of a healthy volunteer with similar anthropometric data.

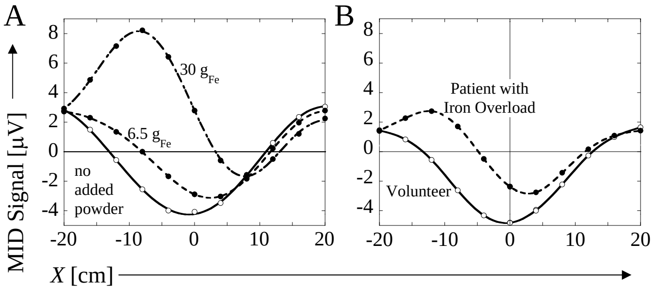


Figure 2. The iron-overload contribution to the magnetic signal. The abscissa X (cm) is the position of the magnetic field axis relative to the center of the trunk. (A) Anthropomorphic plastic phantom with different doses of paramagnetic powder. (B) Magnetization signal of a patient with liver iron overload and of a volunteer with similar anthropometric data.

The contribution to the magnetic susceptibility of the iron overload is obtained by calculating the difference between the magnetization signal and the background signal attributed to the patient on the basis of their anthropometric characteristics. This difference is maximum when the magnetic field axis crosses the liver center of mass. The iron overload in grams is obtained by dividing this signal by the contribution to the signal of 1 g of iron uniformly distributed in the liver of the subject [7].

B. Anthropometric Features

The anthropometric features measured for each subject are: age, body weight and height, body mass index, body surface area, torso cross section areas at the level of the liver, the shoulders and the hips respectively and torso mean thicknesses at the level of the liver, the shoulders and the hips respectively. The last 6 features are calculated from the *3D-shape* of the body which is measured with a system of 6 lasers located on the ceiling. An example of acquisition is reported at the top of Figure 3. Note that the system is not able to detect the empty regions under the body.

The liver iron overload is common to several diseases [7]. As a consequence, no special requirements had been imposed on volunteers enrolling. We applied the Student's t-tests for comparing the means of the anthropometric features of the volunteers and patients populations. We obtained that, with a confidence level of 0.01, the means are the same, with the exception of the age and the torso mean thicknesses.

III. Computational Models for Estimating the Background Signal

In this section we first present the two models developed to compute the background signal. Secondly, we propose two approaches to combine their predictions. Finally, we discuss the experimental protocol adopted to compare their performances.

A. Hybrid Waterman Model

Knowing the geometry of the body and the distribution of its magnetic susceptibility, $\chi(\vec{r})$, the magnetization signal of the body, at position $\vec{X} = (X, 0, 0)$, is obtained by solving the forward problem [13]

$$\phi(\vec{X}) = \int_V g(\vec{r} + \vec{X})\,\chi(\vec{r})\,d\vec{r}, \tag{1}$$

where V is the volume of the body and $g(\vec{r})$ is the weight function (Figure 1B) that gives the contribution to the magnetization signal of a unitary volume of matter with unitary magnetic susceptibility. The function $g(\vec{r})$ was obtained measuring the magnetization signal of a ferromagnetic probe. The quality of this weight function was verified [8] with cylindrical samples of water whose magnetization signals were measured by MID and calculated via (1).

The signal calculated assuming a uniform distribution of susceptibility, equal to that of water ($-9 \cdot 10^{-6}$, SI units), is called the *waterman* [11]. Figure 3A reports the comparison between the measured magnetization signal and the *waterman* signal for an average-built volunteer. Note that in the position around $X = 0$ cm, the measured signal is larger in absolute value than the *waterman*. The opposite observation can be made for the border positions. This is true quite in general as it is shown in Figure 3C, which reports the spatial dependence of the differences between the measured signal and the *waterman* for each volunteer.

The most likely explanation for this discrepancy is that the *waterman* is computed on a volume bigger than the actual body volume, since the *3D-shape* includes the empty regions under it. It was verified that a volume equivalent to that of these regions and filled with water generates a signal comparable to the error of the *waterman* [8]. In addition to this contribution, the error also depends on the difference between the magnetic susceptibility of the water and that of the body tissues. As a first approximation, we used the mean of the errors to correct the *waterman* signal and named this method the *hybrid waterman* model. To compute the background signal, first the *waterman* is calculated from the *3D-shape*, then for each measuring position, the mean of the errors reported in Figure 3C is added.

B. Statistical Learning Model

The statistical learning approach [14] aims to build an estimator of the background signal from a set of given input-output examples, called the *training set*. Another approach would be to directly model the probabilistic relationship between input and output, for instance with Bayesian techniques [15]. The input examples represent the anthropometric features of the volunteers and the output examples are the corresponding background signals. Since we are interested in estimating the background signal only at the fixed measuring positions, it is a vector-valued learning problem, where each output component corresponds to the measure in a specific

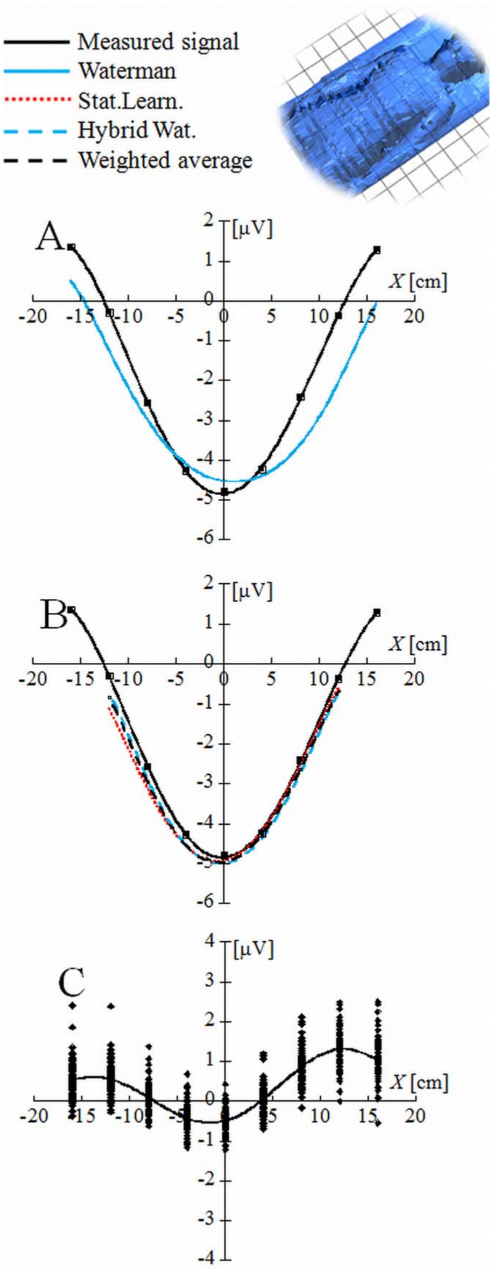


Figure 3. The measured magnetization signal compared to the *waterman* signal (A) and to the estimates of the other models (B) for a volunteer (53 kg). (C) Differences (dots), and their average (solid line), between the measured signal and the *waterman* for the 84 volunteers.

position. It is important that the estimator we learn has good generalization properties on new unseen data and does not predict correctly only the training examples. We begin by formulating the problem in mathematical terms, continue presenting a well-known learning method and conclude the section discussing some choices we made.

We consider a training set of input-output pairs $\{(x_i, y_i) : x_i \in \mathbb{R}^p, y_i \in \mathbb{R}^d\}_{i=1}^n$, where $\mathbb{R}^p$ is called the *input space* (p is the number of anthropometric features) and $\mathbb{R}^d$ is the *output space* (d is the number of measuring positions). We want to estimate a function $f : \mathbb{R}^p \to \mathbb{R}^d$ able to generalize on unseen examples. An estimator, f, can be found minimizing the *empirical error*

$$\mathcal{E}_n(f) = \frac{1}{n} \sum_{i=1}^{n} ||y_i - f(x_i)||_d^2 \, , \qquad (2)$$

which is the average prediction error on the examples of the training set. We search for the estimator in a Reproducing Kernel Hilbert Space (RKHS) [16] defined by a matrix-valued kernel, $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathcal{B}(\mathbb{R}^d)$, where $\mathcal{B}(\mathbb{R}^d)$ is the space of $d \times d$ matrices. The function that minimizes the empirical error $\mathcal{E}_n$ in a RKHS can be written as

$$f(x) = \sum_{i=1}^{n} K(x, x_i) c_i \qquad \text{with } c_i \in \mathbb{R}^d \ \forall i = 1, \cdots, n \, . \qquad (3)$$

The coefficients c_i must satisfy

$$\mathbf{Kc} = \mathbf{y} \, ,$$

where $\mathbf{c} = (c_1, \ldots, c_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ are nd-dimensional vectors where we concatenated the coefficients c_i and the outputs y_i, respectively. $\mathbf{K}$ is called the Gram matrix and is a $n \times n$ block matrix, where each block (i, j) is the $d \times d$ scalar matrix $K(x_i, x_j)$.

To recover the coefficients $\mathbf{c}$ we should invert the matrix $\mathbf{K}$, but $\mathbf{K}$ is not guaranteed to be invertible, nor stable under inversion. Tikhonov regularization [17], [18] solves these issues by adding a regularization term to the empirical risk, thus minimizing the following functional

$$\frac{1}{n} \sum_{i=1}^{n} ||y_i - f(x_i)||_d^2 + \lambda ||f||_K^2 \, .$$

The norm defined by the kernel K controls the smoothness of the candidate vector-valued estimator f and the relationships between its components; λ is called the *regularization parameter* that balances the trade-off between fitting the training data and choosing simpler estimators. The estimator obtained with Tikhonov regularization can still be written as in (3), but the coefficients are now given by

$$\mathbf{c} = (\mathbf{K} + n\lambda \mathbf{I})^{-1} \mathbf{y} \, ,$$

where $\mathbf{I}$ is $nd \times nd$ identity matrix. We note that the penalty term has the effect of stabilizing the inversion of the matrix $\mathbf{K}$ by increasing all its eigenvalues by $n\lambda$.

Several interesting kernels for vector-valued functions have been recently introduced in the literature [19], [20]. Unfortunately, often these kernels require to fine tune many parameters in order to properly leverage the relationships among the output components.

After some preliminary assessment, we opted for a simple matrix-valued kernel of the form $K(x, x') = k(x, x')A$, where $k(x, x')$ is a scalar kernel and A is a positive semi-definite $d \times d$ matrix. We chose a linear kernel $k(x, x') = x \cdot x'$ that proved to produce results as accurate as the non linear gaussian kernel, which requires the tuning of the width and is more expensive to compute. The matrix A was chosen to be the d-dimensional identity matrix. This choice renders the vector-valued learning problem equivalent to solving d scalar learning problems and the relationships that exist among the output components are not exploited. The only coupling among them is the common regularization parameter λ, that imposes the same level of regularization on each output. To choose the proper value of λ we followed the procedure described in Section III-E.

C. Weighted Average Combined Model

The first combined model consists in computing the weighted average of the predictions of the statistical learning model (y_{sl}) and of the *hybrid waterman* model (y_{hw}). For each measuring position, the weighted average is computed according to the formula

$$\overline{y} = \frac{\frac{y_{sl}}{\sigma_{sl}^2} + \frac{y_{hw}}{\sigma_{hw}^2}}{\frac{1}{\sigma_{sl}^2} + \frac{1}{\sigma_{hw}^2}} , \qquad (4)$$

where σ_{sl}^2 and σ_{hw}^2 are the variances of the differences between the measurements and the predictions.

D. Learning the Error of the Waterman Model

The second combined model uses the statistical learning framework to estimate the difference between the measured signal and the *waterman*, using as input variables the anthropometric features. If there exists a relationship between the anthropometric features and the error of the *waterman* model, we should be able to learn it and use the estimates to correct the predictions of the *waterman* model. In the following, we refer to this model as *learning waterman*.

E. Assessment of the Models

In order to assess the performance of a model we need to separate the data from which the model is trained from the data on which it is evaluated. Usually these are called the training and the test sets. All model parameters must be chosen using only the training set, otherwise we obtain a biased estimate of the performance of the model. Since our data is scarce we cannot split it into two sets. Therefore we resort to the Leave-One-Out Cross Validation (LOO-CV), which consists in holding out one example for testing and using the remaining $N-1$ for training. The procedure is repeated until all examples have been used for testing. We thus obtain the predictions for each volunteer and the corresponding errors. The distribution of these errors can be used to assess and compare different models.

For the *statistical learning* and the *learning waterman* models, we used two loops of LOO-CV. The inner loop is used to assess the estimators obtained with different values of the regularization parameter, λ, while the outer loop is used to estimate the performance of the model trained with the optimal value of λ. More precisely, one volunteer at a time is held out for testing. The model is trained on the remaining $N-1$ volunteers with a regularizing parameter λ_1. This parameter is chosen via the inner LOO-CV loop. That is, one of the $N-1$ volunteers is held out to assess the predictions error of the models trained on $N-2$ volunteers with regularizing parameters λ_j, $j = 1, \dots, 150$. Therefore, 150 models will be trained and tested. Then another volunteer (of the $N-1$) is held out, and other 150 models are trained on the remaining $N-2$ volunteers. The prediction errors associated to the same regularizing parameter value λ_j are averaged. The parameter value associated to the smallest average error is used to train the model on all $N-1$ examples. We repeat this procedure until all volunteers are used once for the external loop, yielding N models with their respective regularization parameters and predictions.

IV. Results

Figure 4 reports the distributions (as box-plot representation) of the differences between the measurements and the background signal computed with the different models for the position $X = 0$ cm. The LOO-CV procedure described in the previous section was used to obtain the predictions. Table I reports the standard deviations of the LOO-CV error distributions for positions between $X = -8$ cm and $X = +8$ cm, which is the most significant range for the background signal calculation.

We observe no significant differences between the models in all positions, but for the position X = -8 cm, for which the F-test yields $p < 0.01$ for the differences between the variances of their error distributions. However, combining the two models we do not achieve a significant increase in prediction accuracy. This indicates that most of the background signal is explained by the *3D-shape* and that the other features do not carry additional information.

Furthermore, we found a positive linear correlation (angular coefficient 0.7, R=0.77) between the errors of the *hybrid waterman* and the statistical learning model, indicating that the two models are not independent. In fact, we recall that the statistical learning model uses 6 features that are directly computed from the *3D-shape*.

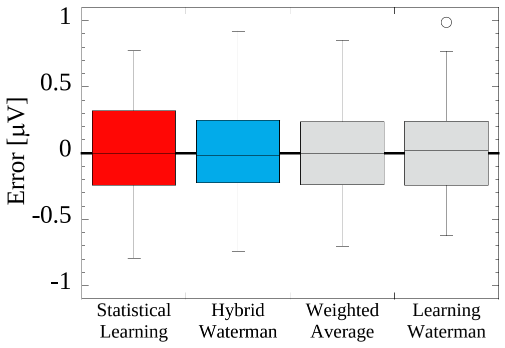


Figure 4. Box-plots of the distributions of the errors

Table I
Standard Deviation of the distribution of the differences between the measured and calculated signal [μV]

Model	Measurement Position [cm]				
	-8	-4	0	$+4$	$+8$
Statistical Learning	0.51	0.41	0.36	0.42	0.50
Hybrid Waterman	0.36	0.37	0.35	0.39	0.44
Weighted Average	0.39	0.37	0.33	0.38	0.43
Learning Waterman	0.39	0.38	0.34	0.38	0.41

Since the center of mass of the patient's liver falls between $X = -8$ cm and 0 cm, the accuracy of the predictions in the positions $X = -8$,-4 and 0 cm is paramount. The average error in these position for the statistical learning model is 0.43 μV, while for the other models is 0.36 μV, which corresponds to an error of about 0.9 g of iron. This error is 20% better than the error (1.1 g) of the first model developed for calculating the background signal, that was evaluated on a dataset consisting of 142 volunteers [8].

V. Conclusion and Future Work

In this paper, we presented two models for calculating the background signal of patients measured with the MID susceptometer. Since 2005 this apparatus is in use at the Galliera Hospital of Genoa and more than 1300 iron overload evaluations have been performed. Both models have been developed on the measurements and the anthropometric features of 84 healthy volunteers. The first model, named *hybrid waterman*, calculates the magnetization signal generated by a water volume with the same external geometry of the subject (*waterman*) and corrects it by adding the mean error of the *watermen* evaluated in a population of healthy volunteers. The second model is based on statistical learning and learns from the volunteer data a mapping from the anthropometric features to the background signal. Finally, two methods to combine these models were proposed. Their performances are very similar and the combination of the two does not introduce significant accuracy gains. The evaluated model error (about 0.36 μV, equivalent to 0.9 g of iron) allows the physicians to monitor the iron overload variations due to the therapy of patients. In order to detect mild overloaded states (between 0.5 and 1 g of iron) this error should be reduced. The limit of 0.5 g corresponds to the reproducibility error of the instrument [6], [7]. To improve the MID error, the number and distribution of healthy volunteers must be increased, while also improving the quality and the number of measured anthropometric parameters. Furthermore, we believe that classifying the volunteers with respect to their anthropometric features and developing of a different model for each category (e.g., babies, adults, oversize, etc) would reduce the error. Regarding the statistical learning model, current work is focused on developing a matrix-valued kernel that is able to exploit the correlations among the measurements in the different positions. A new generation Magnetic Iron Detector is now under construction, all the techniques presented here will be the basis for the development of new models for the estimation of the background signal of the patients that will be examined with the new susceptometer.

From January 2010 the *statistical learning*, the *hybrid waterman* and the *weighted average* models are in use at the Galliera Hospital in Genoa, Italy, for assessing the iron overload with MID.

ACKNOWLEDGMENT

B.G. wishes to deeply thank Prof. M. Marinelli, her Master and Ph.D. supervisor and the inventor of the MID susceptometer. Moreover she wishes to thank: C. Bruzzone, A. Chincarini and P. Beruto, for their help in the *waterman* calculation, and all the members of Galliera and of INFN-Genoa involved in the experiment. L.B. wishes to thank A. Barla for her help in developing the statistical model and Prof. A. Verri for insightful discussions on the problem.

REFERENCES

[1] A. Maggio and A. Hoffbrand, Eds., *Clinical Aspect and Theraphy of Thalassemia*. SEE-Firenze, 2004.

[2] G. Brittenham and D. Badman, “Noninvasive measurement of iron: report of an niddk workshop,” *Blood*, vol. 101, pp. 15–19, 2003.

[3] T. G. St. Pierre, P. R. Clark, W. Chua-anusorn, A. Fleming, G. Jeffrey *et al.*, “Noninvasive measurement and imaging of liver iron concentrations using proton magnetic resonance,” *Blood*, vol. 105, pp. 855–861, 2005.

[4] G. Brittenham, D. Farrell, J. Harris, E. Feldman, E. Danish *et al.*, “Magnetic susceptibility measurement of human iron stores,” *N. Engl. J. Med.*, vol. 307, pp. 1671–1675, 1982.

[5] M. Marinelli, S. Cuneo, B. Gianesin, A. Lavagetto, M. Lamagna *et al.*, “Non-invasive measurement of iron overload in the human body,” *IEEE Trans. Applied Superconductivity*, vol. 16, no. 2, 2006.

[6] M. Marinelli, B. Gianesin, M. Lamagna, A. Lavagetto, E. Oliveri *et al.*, “Whole liver iron overload measurement by a non cryogenic magnetic susceptometer,” *International Congress Series 1300*, pp. 299–302, 2007.

[7] M. Marinelli, B. Gianesin, M. Balocco, P. Beruto, C. Bruzzone *et al.*, “Total iron overload measurement in the human liver region by the magnetic iron detector (mid),” *IEEE Trans. Biom. Eng.*, 2010, (to appear).

[8] B. Gianesin, “Total iron overload measurement in the human liver region by the susceptometer magnetic iron detector (mid),” Ph.D. dissertation, Dept. of Physics - University of Genoa, 2010.

[9] R. Brooks, J. Vymazal, R. Goldfarb, J. Bulte, and P. Aisen, “Relaxometry and magnetometry of ferritin,” *Magnetic Resonance in Medicine*, vol. 40, no. 2, pp. 227–235, 1998.

[10] L. Baldassarre, “Multi-output learning with spectral filters,” Ph.D. dissertation, Dept. of Physics - University of Genoa, 2010.

[11] INFN, DIFI, Ospedale Galliera, DISI, MedService.com Srl, and Genova Robot Srl., “Sistema integrato intelligente di supporto alla decisione per il trattamento delle patologie caratterizzate dal sovraccarico di ferro,” Parco Scientifico e Tecnologico della Liguria, Report di attivita’, 2008, (Pos. N. 26 Avv. 2/2006).

[12] L. Baldassarre, A. Barla, B. Gianesin, and M. Marinelli, “Vector valued regression for iron overload estimation,” in *19th International Conference on Pattern Recognition*, 2008.

[13] G. Tripp, “Physical concept and mathematical models.” in *Biomagnetism: an Interdisciplinary Approach: Proc. NATO Advanced Study Institute on Biomagnetism*, ser. Series A: Life Science, N. A. S. Institutes, Ed., 1983.

[14] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2001.

[15] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[16] N. Aronszajn, “Theory of reproducing kernels,” *Transactions of the American Mathematical Society*, vol. 68, no. 3, pp. 337–404, 1950.

[17] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*. John Wiley, 1977.

[18] C. A. Micchelli and M. Pontil, “On learning vector–valued functions,” *Neural Computation*, vol. 17, 2005.

[19] ——, “Kernels for multi-task learning,” in *NIPS*, 2004.

[20] A. Caponnetto, C. A. Micchelli, M. Pontil, and Y. Ying, “Universal kernels for multi-task learning,” *JMLR*, vol. 9, 2008.

A New Generating Set Search Algorithm for Partially Separable Functions

Lennart Frimannslund
Department of Informatics
University of Bergen
Bergen, Norway
Email: lennart@ii.uib.no

Trond Steihaug
Department of Informatics
University of Bergen
Bergen, Norway
Email: trond@ii.uib.no

***Abstract*—A new derivative-free optimization method for unconstrained optimization of partially separable functions is presented. Using average curvature information computed from sampled function values the method generates an average Hessian-like matrix and uses its eigenvectors as new search directions. For partially separable functions, many of the entries of this matrix will be identically zero. The method is able to exploit this property and as a consequence update its search directions more often than if sparsity is not taken into account. Numerical results show that this is a more effective method for functions with a topography which requires frequent updating of search directions for rapid convergence.**

The method is an important extension of a method for non-separable functions previously published by the authors. This new method allows for problems of larger dimension to be solved, and will in most cases be more efficient.

***Keywords*-Generating Set Search, Derivative-Free Optimization, Partial Separability, Sparsity.**

I. Introduction

Continuous optimization is an important area of study, with applications in statistical parameter estimation, economics, medicine, industry — simply put, anywhere a mathematical model can be used to represent some real-world process or system which is to be optimized. Mathematically, we can express such a problem as

$$\min_{x\in D\subseteq\mathbb{R}^n} f(x), \tag{1}$$

where f is the objective function, based on the model which is defined on the domain D. These models can range from simple analytic expressions to complex simulations. Well known optimization methods such as Newton's method use derivatives to iteratively find a solution. These derivatives must be provided, either through explicit formulas/computer code, or, for instance, automatic differentiation.

Suppose, however, that the objective function is produced by some sort of non-differentiable simulation, or that it involves expressions which can only be computed numerically, such as the solution to differential equations, integrals, and so on. In this case derivatives might not exist, or they may be unavailable if the numerically computed function is subject to some kind of adaptive discretization and truncation and therefore is non-differentiable, unlike the underlying mathematical function. In these cases derivative-based methods are not directly applicable, which leads to the need for methods that do not explicitly require derivatives. For an introduction to derivative free methods the reader is referred to [1].

Generating set search (GSS) methods are a subclass of derivative-free methods for unconstrained optimization. These methods can be extended to handle constraints, but we will focus on the unconstrained case when the domain D in the problem (1) is equal to $\mathbb{R}^n$. A comprehensive introduction to these methods can be found in [12]. In their most basic form these methods only use function values and do not collect any information such as average slope or average curvature information. Computing this information, however, can significantly speed up convergence, and this is done in the methods presented in [2], [3], [4].

In addition, information about the structure of the function known a priori can also be useful. Suppose that the objective function f can be written as a sum of element functions,

$$f=\sum_{i=1}^{m} f_i,$$

where each element function has the property that it is unaffected when we move along one or more of the coordinate directions. For example, we might have

$$f(x_1,x_2,x_3)=f_1(x_1,x_2)+f_2(x_2,x_3). \tag{2}$$

Then, the function is said to be partially separable [9] and we say that f_i has a large null space. If f is partially separable and twice continuously differentiable, then its Hessian matrix,

$$\nabla^2 f(x)=\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1\partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n\partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2}\end{bmatrix},$$

will be sparse. For the function (2) the Hessian element $\frac{\partial^2 f}{\partial x_1\partial x_3}$ will be zero. If the function (2) is not twice continuously differentiable, then the matrix of the corresponding finite differences, that is, the matrix with

$$\Big[f(x_1+h,x_2,x_3+k)-f(x_1+h,x_2,x_3)$$
$$-f(x_1,x_2,x_3+k)+f(x_1,x_2,x_3)\Big]\Big/(hk)=0, \quad (3)$$

in position $(i,j) = (1,3)$ (and with similar expressions for all other (i,j)-pairs) will be sparse for any x, and any nonzero h and k, none of which have to be the same for each (i,j)-pair. The sparsity structure is the same as for the differentiable case, so that the expression (3) is identically zero. This result can be extended to any partially separable function, as proved in [5].

In [15] a GSS method which exploits such structure is presented, which is applicable to the case where these element functions are individually available.

In this paper we present a GSS method which takes advantage of the structure of partially separable functions, without requiring the element functions (which may or may not be differentiable) to be available. It is an extension of the paper [4]. We use the concept of average curvature introduced in [4].

This paper is organized as follows: In section II we outline a basic framework for GSS, as well as the previous work of the authors on which the present paper is based. In Sections III and IV we present our main contribution, which is the framework for handling partially separable functions. Section V contains numerical results, and concluding remarks are given in Section VI.

II. Generating set search using curvature information

We restrict ourselves to a subset of GSS methods, namely sufficient decrease methods with $2n$ search directions, the positive and negative of n mutually orthogonal directions, of unit length. These directions will in general *not* be the coordinate directions. A simplified framework for the methods we consider is given in Figure 1. The univariate function ρ must be nondecreasing and satisfy $\lim_{x\downarrow 0}\frac{\rho(x)}{x} = 0$. For simplicity, increasing the step length can be thought of as multiplying it by 2, and decreasing it as dividing by 2, although these rules may be more advanced. For the formal requirements on these rules, see [12]. Given mild requirements on the function f the step length δ will ultimately go to zero, and the common convergence criterion for all GSS methods is that δ is smaller than some tolerance.

As can be seen from the pseudo code in Figure 1, the set of search directions can be periodically updated. In [4], the authors present a method that computes average curvature information from previously sampled points, assembles this information in a Hessian-like matrix and uses the eigenvectors of this matrix as the search directions, which amounts to a rotation of the old search directions. Once this rotation has been performed, the process restarts, and new curvature information is computed, periodically resulting in new search directions. It is shown that the efficiency of the method can be greatly improved compared to just using the coordinate directions as the search directions throughout.

```
Given set of search directions $\mathcal{Q}$, step length $\delta$ and an
initial guess $x \leftarrow x_0$.
While $\delta$ is larger than some tolerance
    Repeat until $x$ has been updated or all $q \in \mathcal{Q}$ have
    been used:
        Get next search direction $q \in \mathcal{Q}$.
        If $f(x+\delta q) < f(x) - \rho(\delta)$
            Update $x$: $x \leftarrow x + \delta q$.
            Optionally increase $\delta$.
        End if
    End repeat
    If no search direction provided a better function
    value, decrease $\delta$.
    Optionally update $\mathcal{Q}$.
End while
```

Figure 1. Simplified framework for a sufficient decrease GSS method.

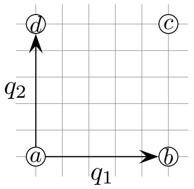


Figure 2. Location of sampled points used for curvature computation.

The computation of curvature information can be done in the following way, which is a slight modification of the methodology presented in [4]. Consider Figure 2, and assume that the current point is the point marked a, and that the next two search directions in the repeat-loop in the pseudo code are the directions shown, q_1 and q_2. When searching along two directions in a row, there are four possible outcomes. Success-success (both the search along q_1 and q_2 produce function values which satisfy the sufficient decrease condition), success-failure (the search along q_1 produces a sufficiently lower function value, but the search along q_2 does not), failure-success, and finally failure-failure. In all of these four cases, by computing the function value at a fourth point, the function values at four points in a rectangle can be obtained. The details are given in Table I. The function values at four such points a, b, c and d can be inserted into the formula

$$\frac{f(c)-f(b)-f(d)+f(a)}{\|b-a\|\;\|d-a\|}. \quad (4)$$

If the objective function is twice continuously differentiable, then (4) is equal to $q_1^T\nabla^2 f(\hat{x})q_2$, where $\hat{x}$ is some point within the rectangle $abcd$. If the function is not twice continuously differentiable, (4) captures the average curvature

Outcome	Notes
SS	The search along q_1 moves the current best point to b, and the search along q_2 moves the current best point to c. The function value at d must be computed separately.
SF	The search along q_1 moves the current best point to b, and the search along q_2 computes the function value at c, but does not move the current best point. The function value at d must be computed separately.
FS	The search along q_1 computes the function value at point b, but does not move the current best point. The search along q_2 computes the function value at point d. The function value at point c must be computed separately.
FF	Neither the search along q_1 nor q_2 update the current best point, but the function values at points b and d are obtained. The function value at point c must be computed separately.

Table I
THE FOUR POSSIBLE OUTCOMES WHEN SEARCHING ALONG TWO CONSECUTIVE DIRECTIONS. S MEANS SUCCESS, F MEANS FAILURE.

in the rectangle.

The rectangle lies in the plane spanned by the search directions q_1 and q_2 since these were used consecutively. By successively reordering how the "get next search direction" statement considers the directions in $\mathcal{Q}$, one can obtain curvature information with respect to all the $n(n-1)/2$ possible different combinations of search directions, in a finite and uniformly bounded number of steps, which depends on n since there are $O(n^2)$ elements of curvature information which must be assembled. (For this reason, the method is not suitable for n larger than about 30, but exploiting structure can allow for much larger n, as will be explained in Section III.)

The information can be assembled in a matrix C_Q, so that C_Q, in the case of a twice continuously differentiable f, contains $q_i^T \nabla^2 f(\hat{x}) q_j$ in positions (i,j) and (j,i), which is curvature information with respect to the coordinate system defined by the n directions in $\mathcal{Q}$. (Note that the point $\hat{x}$ is different for each (i,j)-pair.) The diagonal elements of C_Q must be computed separately, for instance when the step length is reduced, since the preceding repeat-loop, combined with the current f-value then gives the function values at three equally spaced points on a straight line for all n search directions.

Once the matrix C_Q is complete, it is subjected to the rotation

$$C \leftarrow Q C_Q Q^T, \tag{5}$$

where Q is the matrix with the n unique search directions as its columns, ordered so that they correspond to the ordering of the elements in C_Q. C now contains curvature information with respect to the standard coordinate system. The search directions in $\mathcal{Q}$ are then replaced with the positive and negative of the eigenvectors of C.

III. EXTENSION TO SEPARABLE FUNCTIONS

Suppose the function f is partially separable. As mentioned in the introduction, the Hessian will be sparse if f is twice continuously differentiable, and if the Hessian is not defined, the matrix of average curvature information will be sparse [5]. Let r be the number of nonzero elements in the lower diagonal of these curvature matrices. Then, even though the matrix C can be restricted to have this sparsity pattern, the matrix C_Q cannot be assumed to be sparse, since we cannot expect the finite differences (4) to be zero for arbitrary search directions $\mathcal{Q}$. However, sparsity can still be exploited.

Define the Kronecker product. Given two matrices $A \in \mathbb{R}^{m \times n}$ and B, then the Kronecker product $A \otimes B$ is given as

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix}. \tag{6}$$

The Kronecker product is useful in the present context because of the relation

$$AXB = C \Leftrightarrow (B^T \otimes A)\mathbf{vec}(X) = \mathbf{vec}(C). \tag{7}$$

Here $\mathbf{vec}(X)$ and $\mathbf{vec}(C)$ are vectors containing the entries of the matrices X and C stacked row-wise [11].

Using (6) and (7) the rotation (5) can be written implicitly as

$$(Q^T \otimes Q^T)\mathbf{vec}(C) = \mathbf{vec}(C_Q). \tag{8}$$

Since we impose a sparsity structure on C as well as symmetry, all the entries in the upper triangle, as well as all the zero entries of $\mathbf{vec}(C)$ can be removed from (8), resulting in the overdetermined equation system

$$(Q^T \otimes Q^T)P_c\overline{\mathbf{vec}}(C) = \mathbf{vec}(C_Q), \tag{9}$$

where the vector $\overline{\mathbf{vec}}(C)$ contains the r elements of C to be determined, and the $n^2 \times r$ 0-1 matrix P_c adds together the columns corresponding to upper and lower diagonal elements C_{ij} and C_{ji} for all off-diagonal elements, and deletes the columns corresponding to zero entries in C. For example, if C is to be tridiagonal and is of size 3×3, that is,

$$C = \begin{bmatrix} \times & \times & \\ \times & \times & \times \\ & \times & \times \end{bmatrix},$$

then it has one zero element and five nonzero elements in the lower triangle, so that P_c has size 9×5 and reads:

$$P_c = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

Since the equation system (9) is overdetermined, we can select r rows from the coefficient matrix and the right-hand side, resulting in the $r \times r$ equation system

$$P_{\text{row}}(Q^T \otimes Q^T)P_c\overline{\mathbf{vec}}(C) = P_{\text{row}}\mathbf{vec}(C_Q), \tag{11}$$

where P_{row} is an $r \times n^2$ 0-1 matrix which selects r rows. P_{row} will be the first r rows of a permuted $n^2 \times n^2$ identity matrix. The resulting equation system (11) will be significantly smaller than its counterpart (8) when a sparsity structure is imposed on C, and the corresponding effort required to compute the right-hand side is similarly smaller. If there are only $O(n)$ elements to be determined, then the number of steps needed to compute the entire right-hand side $P_{\text{row}}\mathbf{vec}(C_Q)$ does not depend on n, which does away with the practical limit on dimension discussed in the previous section.

Exactly which rows P_{row} should select in order to create a well-conditioned coefficient matrix is nontrivial, and is sometimes called the subset selection problem in the literature (see e.g., [7]). One suitable solution procedure is to determine these rows by computing a strong rank-revealing QR factorization of the transpose of $P_{\text{row}}(Q^T \otimes Q^T)$ and selecting the rows chosen by the theory and algorithms of Gu and Eisenstat, presented in [10]. An implementation of this selection procedure can be found in [14].

IV. Convergence theory

The method presented so far, being a sufficient decrease method with $2n$ search directions which are the positive and negative of n mutually orthogonal directions, adheres to the algorithmic framework and convergence theory of Lucidi and Sciandrone [13]. We can therefore state the following theorem, without proof.

Theorem 1: Suppose f is continuously differentiable, bounded below and the level set $\mathcal{L}(x) = \left\{ y \middle| f(y) \leq f(x) \right\}$ is compact. Then, the method converges to a first-order stationary point.

We now prove that if f is twice continuously differentiable, then the computed curvature matrix C converges to the true Hessian in the limit.

Define

$$A = P_{\text{row}}(Q^T \otimes Q^T)P_c.$$

Let f be twice continuously differentiable and the Hessian Lipschitz-continuous in the sense that

$$\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|. \tag{12}$$

Define r pairs of vectors $p^{(k)}, q^{(k)}$ $k = 1, \ldots, r$, all of unit length, such that the kth row of A is equal to

$$\left(p^{(k)T} \otimes q^{(k)T}\right)P_c. \tag{13}$$

This means some of these vectors will be equal, but the pairs will be unique. In addition let r points $x^k,\ k = 1, \ldots, r$, be such that element k of $P_{\text{row}}\mathbf{vec}(C_Q)$,

$$(P_{\text{row}}\mathbf{vec}(C_Q))_k = p^{(k)T}\nabla^2 f(x^k)q^{(k)}.$$

Let η be such that

$$\max_{i,j} \|x^i - x^j\| = \eta.$$

Let $\mathcal{N}$ be the neighborhood of points such that

$$\mathcal{N} = \left\{ x \middle| \|x - x^k\| \leq \eta,\ k = 1, \ldots, r \right\}.$$

For convenience, let us restate (11), as

$$A\overline{\mathbf{vec}}(C) = P_{\text{row}}\mathbf{vec}(C_Q). \tag{14}$$

Lemma 2: Assume A is invertible. Let C be the symmetric $n \times n$ matrix constructed from the solution of (14). Then, there exists an $x \in \mathcal{N}$ such that

$$\|\nabla^2 f(x) - C\| \leq \|A^{-1}\|nL\eta.$$

Proof. Let us rewrite the contents of $P_{\text{row}}\mathbf{vec}(C_Q)$:

$$\begin{aligned} & (P_{\text{row}}\mathbf{vec}(C_Q))_k \\ = \; & p^{(k)T}\nabla^2 f(x^k)q^{(k)}. \\ = \; & p^{(k)T}\left(\nabla^2 f(x) + \nabla^2 f(x^k) - \nabla^2 f(x)\right)q^{(k)} \\ = \; & \left[p^{(k)T}\nabla^2 f(x)q^{(k)}\right] + \\ & \left[p^{(k)T}(\nabla^2 f(x^k) - \nabla^2 f(x))q^{(k)}\right]. \end{aligned} \tag{15}$$

Then, and in addition defining $h = \overline{\mathbf{vec}}(\nabla^2 f(x))$, equation (14) can be written as

$$A\overline{\mathbf{vec}}(C) = Ah + \epsilon. \tag{16}$$

Here $(Ah)_k$ is the expression in the first parenthesis of (15), and ϵ_k is the expression in the last parenthesis of (15). If we consider the norm of a single element in ϵ, this is

$$\begin{aligned} |\epsilon_k| & \leq \|p^{(k)}\|\|\nabla^2 f(x^k) - \nabla^2 f(x)\|\|q^{(k)}\| \\ & \leq L\eta, \end{aligned} \tag{17}$$

using (12) and the fact that p and q have unit length. When solving (14), we get

$$\overline{\mathbf{vec}}(C) = h + A^{-1}\epsilon.$$

If we consider a single element of $\overline{\mathbf{vec}}(C)$ and h we can write

$$|(\overline{\mathbf{vec}}(C))_k - h_k| \leq \|A^{-1}\||\epsilon_k|,$$

which can also be written

$$|C_{ij} - (\nabla^2 f(x))_{ij}| \leq \|A^{-1}\||\epsilon_k| \tag{18}$$

Using the property of the 2-norm that

$$\|A\|_2 \leq n \max_{i,j} |a_{ij}|,$$

as well as (17) we can extend (18) to

$$\|C - \nabla^2 f(x)\| \leq \|A^{-1}\| nL\eta,$$

which completes the proof. □

We must now prove that there always exists a matrix A with rank r, and that the term $\|A^{-1}\|$ is uniformly bounded. Since A is made up of the rows of the matrix $(Q^T \otimes Q^T)P_c$, there will be a choice of rows which imply full rank if the matrix $(Q^T \otimes Q^T)P_c$ has rank r.

Lemma 3: For any orthogonal matrix Q and any sparsity structure to be imposed on C, the matrix $(Q^T \otimes Q^T)P_c$ has full rank r, and its smallest singular value σ_r satisfies $\sigma_r \geq 1$.

Proof. Since Q is orthogonal, so is Q^T, and also $(Q^T \otimes Q^T)$. For any sparsity structure, right-multiplying $(Q^T \otimes Q^T)$ with P_c either adds together two columns, or deletes columns. Consequently, the columns of the resulting matrix $(Q^T \otimes Q^T)P_c$ are orthogonal (which implies full rank), and have either length one or length $\sqrt{2}$. It then follows that the singular values are equal to the length of the column vectors, either 1 or $\sqrt{2}$. □

Lemma 4: P_{row} can be chosen such that for a given n, the smallest singular value of A is uniformly bounded below, and consequently that $\|A^{-1}\|$ is uniformly bounded.

Proof. This result follows from the theory and methods of Gu and Eisenstat [10], which guarantee that the rows of A (or equivalently the columns of A^T, as is done in [10]) can be selected from the rows of $(Q^T \otimes Q^T)P_c$ in such a way that the smallest singular value of A is larger than or equal to the smallest singular value of $(Q^T \otimes Q^T)P_c$, divided by a low order polynomial in n and r. Since n and r are given and the smallest singular value of $(Q^T \otimes Q^T)P_c$ is always larger than or equal to 1, the result follows. □

Finally, we show that η goes to zero as the GSS method converges to a stationary point.

Lemma 5: Assume that the step length expansion factor is uniformly bounded by, say, M. Then, as the step length δ go to zero, so does η.

Proof. That the step length δ goes to zero is an integral part of the convergence theory of GSS methods and is proved in e.g. [12]. η is the diameter of neighborhood of points $\mathcal{N}$. Since all the points in $\mathcal{N}$ lie within the rectangles of points used in the formula (4), it follows that η must be smaller than maximum possible distance between the first and the last corner point used for computing C. Suppose, that when the computation of C is started the step length is $\delta_{\max}$, and that the maximum possible number of step length increases before C is computed is t. Then we have

$$\eta \leq \sum_{k=0}^{t} \delta_{\max} M^{k-1}.$$

The only variable in this expression is $\delta_{\max}$, and we know it goes to zero as the method converges. Consequently, so must η. □

This allows us to state the following theorem:

Theorem 6: Assume that f is twice continuously differentiable, bounded below and that the level sets $\mathcal{L}(x)$ are compact. Then, as the method converges, C converges to the true Hessian.

The proof follows from the preceding Lemmas. This result, together with the preliminary numerical results in [6] allows us to conjecture that the method actually converges to second-order stationary points.

V. Numerical Results

For the sake of brevity, there are many common implementation details for GSS methods which have been omitted in this paper. For instance, it is possible to have individual step lengths (e.g., n step lengths, one for each positive-negative search direction pair), to compute an approximate gradient and performing Newton-like steps, have variations on how step length(s) can be increased and decreased, choose ρ in several ways, and so on. These all affect the numerical performance of the method. The purpose of the present paper is, however, to show the benefits of exploiting sparsity when computing curvature information in the context of GSS methods. For this reason, it is the relative increase in performance when exploiting sparsity that is important in our numerical experiments, which used, among other things, n individual step lengths. The results are shown in Table II. The table reads, from left to right, the function name, and the dimension n. The functions are all differentiable, so the column r indicates the number of nonzero elements in the Hessian matrix. Then follow the number of function evaluations required to reduce the objective function value from the recommended initial solution to 10^{-5}, first for the method exploiting sparsity, and finally for the method not exploiting sparsity. The functions all have an optimal objective function value $f^* = 0$.

As one can see, one sometimes can get significant savings when exploiting sparsity, for example for the extended Rosenbrock function, CRAGGLVY, MOREBV and TQUARTIC. The reason for this is that the new method is able to rotate its search directions more often, which adapts them to the local topography of the objective function.

If we look at the extended Rosenbrock function there are several advantages to exploiting sparsity. Firstly there are $3n/2$ nonzero elements in the Hessian, which means

Function	n	r	Sparsity	No sparsity
BRYBND	10	49	936	1100
	50	329	4111	3774
CHNROSNB	10	19	2103	2971
	25	49	7400	15451
	50	99	26385	52574
CRAGGLVY	4	5	118	481
DECONVU	61	767	3232	15790
DQRTIC	10	9	335	471
	50	49	2991	3774
Ext. Rosenbr.	16	24	3369	6407
	32	48	6945	16577
	64	96	13889	50635
FREUROTH	10	19	912	1226
LIARWHD	36	71	3602	5257
MOREBV	10	27	363	521
	50	147	1320	5769
SBRYBND	10	49	747	736
SPARSQUR	100	1232	2878	2988
TQUARTIC	50	99	9022	14176
TRIDIA	20	39	1065	1453
	30	59	1662	2791
	50	99	2843	5621

Table II
Number of function function evaluations required to reduce the objective function value to 10^{-5}, starting at the recommended initial solution, for selected functions from the CUTEr test set [8].

that in relative terms, C can be computed increasingly cheaply as n grows. Secondly, the Hessian is block diagonal, which implies that it has element functions which can be optimized independently. As a consequence the eigenvectors have a block structure as well, which, since there are n step lengths, actually means that the method exploiting sparsity automatically optimizes the element functions independently of each other. This is reflected in the fact that the number of function evaluations needed to obtain a solution grows more or less linearly with n, as opposed to when not exploiting sparsity, where the growth in function evaluations is almost quadratic.

If the topography is such that frequent updating of the search directions is not important, then the results are more similar for the two algorithms.

VI. Conclusion

We have presented a GSS algorithm which exploits the partial separability of the objective function. The method is provably convergent to first-order stationary points, and based on its theoretical and numerical properties we conjecture that it is convergent to second-order stationary points. Numerical results indicate that exploiting separability can lead to significant improvement in convergence, in many cases.

Acknowledgements

The authors would like to thank Marielba Rojas for suggesting the use of rank-revealing QR factorizations. The first author gratefully acknowledges partial funding from The Norwegian Research Council, Gassco and Statoil under contract 175967/S30.

References

[1] A. R. Conn, K. Scheinberg, and L. N. Vicente. *Introduction to Derivative-Free Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.

[2] I. D. Coope and C. J. Price. A direct search conjugate directions algorithm for unconstrained minimization. *The ANZIAM Journal*, 42(E):C478–C498, 2000.

[3] A. L. Custódio and L. N. Vicente. Using sampling and simplex derivatives in pattern search methods. *SIAM Journal on Optimization*, 18(2):537–555, 2007.

[4] L. Frimannslund and T. Steihaug. A generating set search method using curvature information. *Computational Optimization and Applications*, 38(1):105–121, 2007.

[5] L. Frimannslund and T. Steihaug. Sparsity of the average curvature information matrix. *PAMM, Proc. Appl. Math. Mech.*, 7:1062101–1062102, 2007.

[6] L. Frimannslund and T. Steihaug. Convergence basins for some derivative-free optimization methods on problems with saddle points. Technical Report 394, Department of Informatics, University of Bergen, Norway, 2010.

[7] G. H. Golub and C. F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 3rd edition, 1996.

[8] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr (and SifDec), a constrained and unconstrained testing environment, revisited. Technical Report RAL–TR–2002–009, Computational Science and Engineering Department, Rutherford Appleton Laboratory, 2002.

[9] A. Griewank and Ph. L. Toint. On the unconstrained optimization of partially separable functions. In M. Powell, editor, *Nonlinear Optimization 1981*, pages 301–312. 1982.

[10] M. Gu and S. C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM J. Sci. Comput.*, 17(4):848–869, 1996.

[11] R. A. Horn and C. R. Johnson. *Topics in matrix analysis*. Cambridge University Press, Cambridge, United Kingdom, 1991.

[12] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, 2003.

[13] S. Lucidi and M. Sciandrone. On the global convergence of derivative-free methods for unconstrained optimization. *SIAM Journal on Optimization*, 13(1):97–116, 2002.

[14] S. R. Pope. *Parameter Identification in Lumped Compartment Cardiorespiratory Models*. PhD thesis, North Carolina State University, Raleigh, North Carolina, USA, 2009.

[15] C. P. Price and Ph. L. Toint. Exploiting problem structure in pattern search methods for unconstrained optimization. *Optimization Methods and Software*, 21(3):479–491, 2006.

A Quasi Real-Time Parallel FE Analysis of Masonry Walls

Franco Milicchio
Dept. of Informatics and Automation
Roma Tre University
Rome, Italy
milicchio@dia.uniroma3.it

Giovanni Formica
Dept. of Studies on Structures
Roma Tre University
Rome, Italy
formica@uniroma3.it

***Abstract*—We present a parallel implementation of a non-linear finite element analysis of masonry walls. The implementation is based on a shared-memory architecture, while the mechanical simulation is inspired by a model recently developed for this type of structures. Such a model showed to be both reliable and efficient in predicting collapse mechanisms for safety assessment purposes. Its formulation, as we will explain, favors naturally a parallel implementation because the collapse mechanisms are assumed independently for each Finite Element (FE). Additionally, the non-linear response of the same Element is offered at fast computations, because it is based on average elasto-plastic stress distributions which well simulates the more significant mechanical criticisms, *i.e.*, the frictional toughness along squeezing lines.**

***Keywords*-Finite element methods; Nonlinear systems; Parallel programming.**

I. Introduction

In the field of masonry mechanics, there was an extensive production of research works across the last thirty years. However, comparing the existing literature with other structural typologies such as concrete and steel structures, we face an almost embarrassing bias, finding huge deficiencies in both theory, practice, and numerical simulations. In fact, except for reinforced masonry structures, these inadequacies are still reflected in technical codes and predictive software analysis tools, with dramatic results if we observe recent collapses, such as the Umbrian-Marchigian earthquake in 1997, or in Mexican states of Puebla and Oaxaca in 1999, the infamous Tehuacan earthquake, of magnitude 7, damaging about 1800 historic buildings, among them several temples and convents from early colonial era.

The complexity of simulating the behavior of masonry structures is evident when investigating all available predictive tools. Masonry structures manifest different inhomogeneities that require computational approaches to account for different scales, both in length and in time. Numerical simulations struggle to grasp such features, as they cannot be underestimated when extrapolating crucial information on the overall local and global structural behavior: these procedures are still far from being robust and accurate, yielding acceptable results when dealing with full three-dimensional analysis. Multiscale and algebraic multigrid approaches (see [1], [2] and [3] for a reference) recently emerged for masonry mechanics as a possible research direction, on the basis of simplified, yet complicated, similarities with composite structures. A first step on this track has been recently published in [4]; however, the unstructured nature of general mechanical problems dampen the efficiency of such these methods, unless some ad-hoc solutions is adopted, as proposed in [5], [1], and [6].

The present work presents a parallel implementation, based on a shared-memory architecture, of a non-linear finite element analysis of masonry walls. Fine-grained modeling accurately represent several mechanical features, while coarse scale ones achieve better performances in terms of computational time, while sacrificing precision. An alternative approach has been previously proposed in [4], where a fine-scale model is employed in order to generate a coarse-grained finite element formulation. Our objective is to extend the previous work in order to achieve a quasi real-time simulation, *i.e.*, within a time-frame perceived as "immediate". To the best of our knowledge, no attempt has been previously made in simulating the non-linear behavior of masonry walls; however, several works lie in the field of real-time simulation, for example [7] in the field of visco-elastic materials, and [8] in the computer graphics area.

II. Finite Element Formulation

Let us consider an equivalent continuum model, and let us represent its linear elastic behavior by means of a fine-grained modeling of the overall masonry assembly. Such identification technique is known in literature as "refined Cauchy", although several other alternative approaches have been proposed (cf. [9]).

Then, let the constitutive Cauchy law be $\sigma = E\varepsilon$, where σ and ε represent the 2nd order tensors of stress and linear strain, respectively, and E being the 4th order elastic tensor. Our reference finer scale model, initially proposed in [10], is comprised on an assembly of bricks, considered as rigid bodies of dimensions $h \times b \times s$ (*i.e.*, height, width, and thickness), connected to each other by means of a thick mortar joint, modeled as elastic springs, with normal and tangential stiffness, equal to E and G. Therefore, according to the chosen identification technique, both discrete and continuum models possess the same homogeneous strain

ISBN: 978-1-61208-101-4

patterns; the reference elementary volume will be comprised of one brick, and six mortar joints connecting the reference brick to all its neighboring ones. Additionally, as detailed in [9], the rotational field is obtained imposing the momentum balance on the reference elementary volume, thus obtaining the components of the elastic tensor E as:

$$\begin{aligned}
E_{1111} &= \frac{1}{2}\frac{b+a}{h+a}\left(\frac{Gs(b+a)}{2a}+2\frac{Es(h+a)}{a}\right)\\
E_{2222} &= \frac{Es(h+a)}{a}\\
E_{1212} &= \frac{\frac{Gs(b+a)}{2a}\left(\frac{Es(b+a)}{2a}+2\frac{Gs(h+a)}{a}\right)}{\frac{1}{2}\frac{b+a}{h+a}\left(\frac{Es(b+a)}{2a}+2\frac{Gs(h+a)}{a}\right)+\frac{Gs(h+a)}{a}}\\
E_{2121} &= E_{1212}\,,
\end{aligned}$$

with all the remaining coefficients being zero.

The FE formulation is based on the classical 5β element (see, *e.g.*, [11] and [9]), a quadrilateral assumed stress mixed-form finite element. Let us indicate with (ξ,η) the intrinsic coordinates, and with (x,y) the global ones; the discretized displacement may be therefore expressed as the following:

$$u := (u_x, u_y)^\top = N(\xi,\eta)d\,, \tag{1}$$

with

$$N(\xi,\eta) := \left[\begin{array}{ccccc} N_1 & 0 & \dots & N_4 & 0\\ 0 & N_1 & \dots & 0 & N_4 \end{array}\right]. \tag{2}$$

Functions N_i interpolate the displacement through 2×4 node parameters, assembled in the vector d, with standard bilinear interpolating functions. The masonry wall is then discretized on a regular quadrangular mesh. The FE formulation is locking free [12], its five stress parameters, collected in the vector β, interpolating the stress as follows

$$\sigma := (\sigma_x, \sigma_y, \sigma_{xy})^\top = P(\xi,\eta)\beta\,, \tag{3}$$

with

$$P(\xi,\eta) := \left[\begin{array}{ccccc} 1 & 0 & 0 & \eta & 0\\ 0 & 1 & 0 & 0 & \xi\\ 0 & 0 & 1 & 0 & 0 \end{array}\right]. \tag{4}$$

Finally, the compatibility and equilibrium conditions are expressed as

$$\begin{aligned}
H\beta - Qd &= 0 \qquad (5)\\
Q^\top\beta - p &= 0\,, \qquad (6)
\end{aligned}$$

where

$$H := \int_{\Omega_e} P^\top E^{-1} P\,, \qquad Q := \int_{\Omega_e} P^\top D N\,.$$

A. Non-linear Plasticity

The more significant non-linearity is essentially on the frictional behavior; we neglect the coupling with the damage process. More sophisticated numerical simulations, based on fine-scale models and experimental evidences, show that the frictional resistance plays an important role in the structural response under cyclic loading conditions [10], [4], [13]. Within an elastoplastic model, a Mohr-Coulomb criterion is employed in order to characterize the inelastic part of the structural response. The key idea, based on the microplane modeling, is reported in several works such as [14], [15]. The proposed model holds small-strain elastoplasticity and thermodynamical frameworks, being plastic deformation the only dissipative mechanism.

The frictional criterion is then described by the following condition:

$$|\tau_n| - c - \mu\sigma_n \leq 0 \tag{7}$$

where τ_n and σ_n are the shear and normal stress, respectively, acting on a plane with normal vector n, while c and μ are the cohesion and static friction coefficient, respectively. Since we are in a context of non-associated plasticity, we can assume the increments $\dot{\varepsilon}_p$ of plastic deformation to be only in the shear direction, *i.e.*:

$$\dot{\varepsilon}_p = \dot{\gamma}\frac{\tau_n}{|\tau_n|} \tag{8}$$

where $\dot{\gamma} \geq 0$ is the increment of the plastic multiplier.

The yield surface $f[\sigma]$ is completed by two conditions bounding normal tension and compression, thus providing the following elastic domain $\mathcal{D}_e$:

$$\mathcal{D}_e := \{\sigma \,:\, f[\sigma] \leq 0\}\,, \quad \text{with}$$

$$f[\sigma] := \begin{cases} -c - \mu\sigma_n + |\tau_n| \\ -\sigma_{yt} + \sigma_n \\ \sigma_{yc} + \sigma_n \end{cases} \tag{9}$$

where σ_{yt} and σ_{yc} are the tension and compression yield normal stresses, respectively, and we define compressive stress as negative.

Within the representation of the element stress field, and by means of the Haar-Kármán principle, we usually get the admissible stress field by controlling the stress level by (9) at some Gauss points of each element, and then numerically integrating on the same element. This is a standard way for FE formulations in elastoplasticity, we anyway tested in our numerical implementation.

We follow an alternative approach, which is less computationally expensive, yet accurate enough, as we will show. Following [16], we reformulate the elastoplastic response of the assumed stress FE by adopting a kinematic approach. Such approach defines a discrete number of possible mechanisms, corresponding to the plastic deformations that the

Element exhibits. More specifically, within each element we consider a set $\mathcal{S}_e$ of possible "bands" (*i.e.*, lines in the 2D element), along which the plastic deformation can take place. This corresponds to fix a discrete number of possible directions $\dot{\varepsilon}_{pn}$, depending on the assumed band with normal $n \in \mathcal{S}_e$. To define $\mathcal{S}_e$, we consider the possible collapse mechanisms of a generic assembly of bricks contained in an element. We choose five bands with their three associated plastic mechanisms, as depicted in Figure 1.

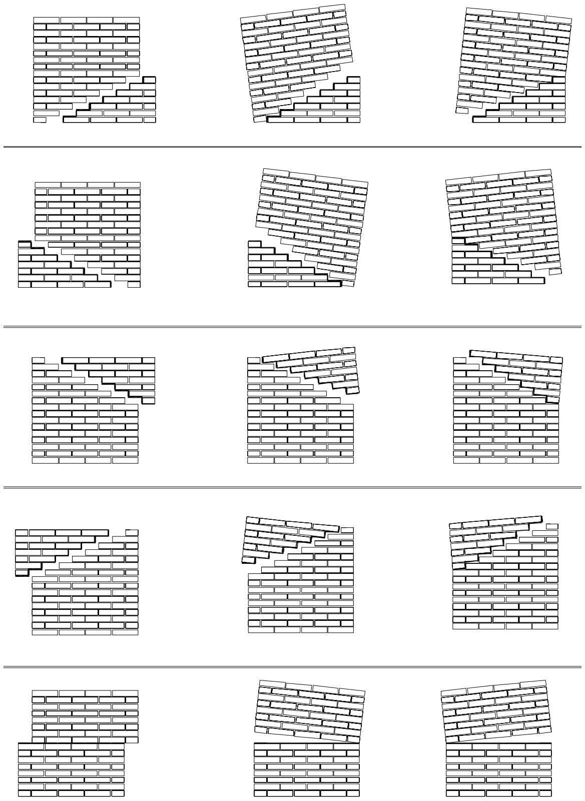


Figure 1. Possible mechanisms that a single FE represents.

Therefore, by imposing the yield conditions in (9) for each band we define for each FE an envelop of planes in the space of the discretized stress components (3).

III. Numerical Solver

A. Overview

Our solver relies on the numerical properties exhibited by non-linear elasto-plastic media. As suggested by the currently available literature, our solver employs a path-following algorithm in order to retrieve the non-linear equilibrium path. The path-following technique [17], recovers the equilibrium path, arising from a non-linear structural response s, subject to a load p varying with a scalar term λ, by means of the following alternative system:

$$\begin{cases} r(u(\xi), \lambda(\xi)) = s(u(\xi)) - \lambda(\xi) p \\ g(u(\xi), \lambda(\xi)) = \xi\,, \end{cases} \tag{10}$$

where g is a known constraining surface, and r represents the equilibrium error.

The system expressed in (10) is then solved by means of a predictor-corrector iterative scheme, starting with an initial solution ($u^0 = 0$, $\lambda^0 = 0$). The solution is attained at convergence on the k-th step in the j-th iteration, providing the ensuing equilibrium point ($u^{k+1} = u_{j*}$, $\lambda^{k+1} = \lambda_{j*}$); the predictor is a trial solution obtained by extrapolation of previous solutions, while the employed corrector is based on an iterative Newton-Raphson scheme:

$$\begin{aligned} r_j &= r(u_j, \lambda_j) = s(u_j) - \lambda_j p \\ \dot{\lambda} &= (u^\top p)^{-1} u^\top r_j \\ \dot{u} &= K^{-1} r_j + \dot{\lambda} u \\ u_{j+1} &= u_j + \dot{u} \\ \lambda_{j+1} &= \lambda_j + \dot{\lambda} \end{aligned}$$

The structural response $s(u_j)$ is evaluated by means of a predictor-corrector scheme, as detailed in [4]. This solution employs the Haar-Kármán principle, *i.e.*:

$$\Phi(\sigma) := 1/2 \int_\Omega (\bar{\sigma} - \sigma)^\top E^{-1} (\bar{\sigma} - \sigma) = \min\,, \tag{11}$$

for all equilibrated stress $\bar{\sigma}$.

B. Parallel Implementation

The solver described in the previous section was implemented on a shared-memory architecture[1], and in the following we will outline the main components along with a speed-up measurement.

First, we highlight the fact that initial and terminal operations in FE analysis, *i.e.*, the stiffness matrix assembly and the output variable update processes, are inherently parallelizable. The non-linear plastic analysis, however, is comprised of parts that are not parallel in the strict sense. A comprehensive diagram of the overall solver architecture is pictured in Figure 2.

Matrix assembly and stress update, being the latter modeled with an incremental algorithm, are carried out in parallel by means of the *reduction* operation. The non-linear structural response of the masonry wall can be easily carried out in a parallel fashion.

The structural response $s(u)$ needed to solve the system (10), is evaluated by means of the Haar-Kármán principle (11), *i.e.*, formulating it as a quadratic programming problem, as expressed in equation (11). The Haar-Kármán principle is local to each element, and may be locally solved

[1] The software was implemented in C++ following the OpenMP 3.0 specification; benchmarks were conducted on an Intel Core 2 Duo processor at 2.9 GHz, with 4 GB of RAM.

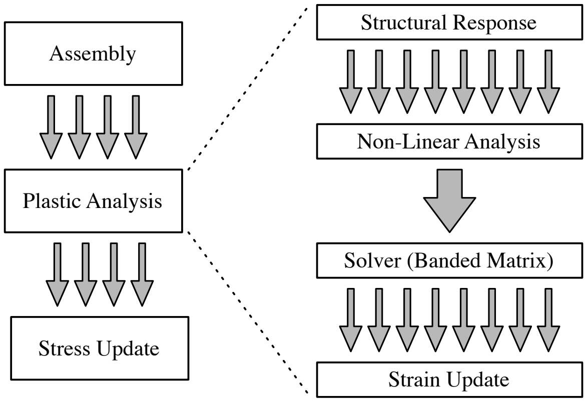


Figure 2. The numerical solver architecture (left), and a detailed plastic analysis diagram (right). Multiple arrows indicate a parallel execution.

Table I
BENCHMARKS FOR THE PROPOSED PARALLEL FE SOLVER.

Threads	1	2	4	8	16	32
Time	2.75	1.68	1.04	0.97	0.65	0.43
Speed-up	1	1.64	2.64	2.83	4.22	6.37

iteratively, adopting the Goldfarb-Idnani method [18]: this allows us to parallelize this process, up to the actual solution of the system of linear equations arising from (10). As pictured in Figure 2, the aforementioned system is solved serially, while the ensuing strain update, and the subsequent output variables, are naïvely parallelized.

IV. RESULTS

The chosen test bed for our parallel implementation is the Pavia's test, an experimental test performed in the University of Pavia [19]. The horizontal displacement employed in the test is pictured in Figure 4, with the overall time of analysis equal to 20 seconds. We mention in passing that a quasi real-time process involves updates of the outcomes with a minimal frequency of 20 Hz circa (cf., [7], [20], and [8]).

Results are reported in Table I, where we reported the number of threads employed in the test bed, the overall time for the nonlinear analysis, and the speed-up [21]; the latter quantity has been calculated as $S := T_i T_1^{-1},\ i = 1, 2, \ldots$, where i indicates the number of threads involved in the analysis.

As expected by known theoretical results (cf. [22] and [23]), we are obtaining a sub-linear speed-up, detailed in Table I, and pictured in Figure 5. In order to better analyze the results, we recall that quasi real-time requires updates at 20 Hz, *i.e.*, 0.05 seconds. Comparing the total analysis time of 20 seconds, with the actual analysis process, we obtain that our FE formulation allows us to obtain a quasi-RT update with a number of threads equal or above four.

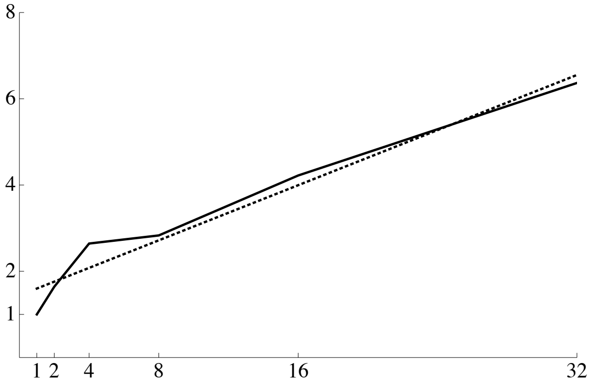


Figure 3. Graph of the speed-up value plotted against the number of processes (in solid black); graph of the linear fit (dotted).

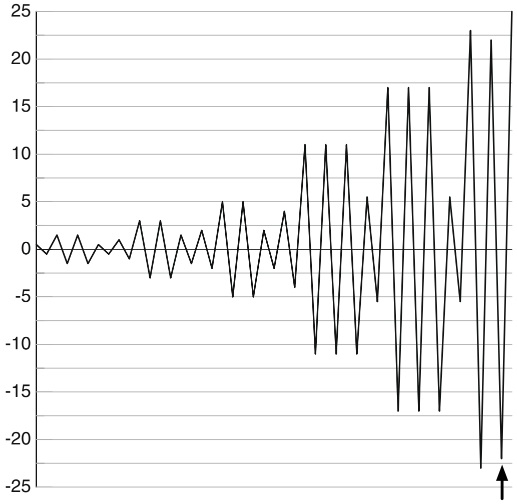


Figure 4. Graph of the horizontal displacement, measured in *mm*, with respect to time. The arrow marks the last input cycle.

V. CLOSING REMARKS

We proposed in this manuscript a parallel implementation of a FE formulation for the analysis of masonry walls involving non-linear plasticity. Such implementation, based on a shared-memory architecture, allows us to obtain results in a quasi real-time fashion.

The proposed FE formulation relies on a fine-grained approach, where a detailed model is employed in the formulation of coarser elements, grasping the non-linear mechanic behavior and obtaining considerably better performances compared to a naïve finer modeling approaches.

Coupling this novel multiscale approach to non-linear plasticity, with a parallel implementation of the analysis process, we are able to hold quasi real-time performances. A future direction of research will investigate all the possible issues affecting performances, clarifying the optimal number of threads on specific architectures, and comparing standard solvers with our custom solution.

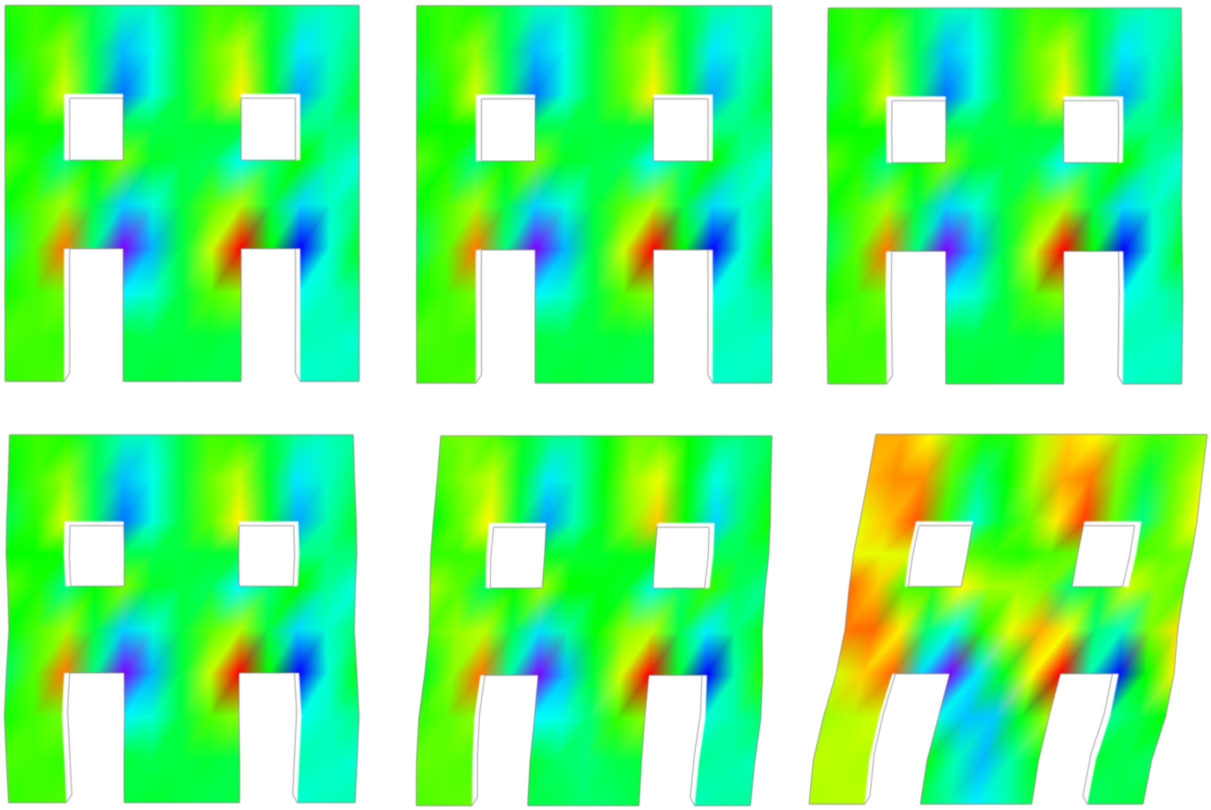


Figure 5. Rendering of the stress σ_{xy} for the last input cycle in the analysis, indicated with an arrow in Figure 4.

REFERENCES

[1] J. Fish and Q. Yu, "Multiscale damage modelling for composite materials: theory and computational framework," *Int. J. Numer. Meth. Eng.*, vol. 52, pp. 161–191, 2001.

[2] K. Stüben, "A review of algebraic multigrid," *J. Comput. Appl. Math.*, vol. 128, pp. 281–309, 2001.

[3] T. J. R. Hughes, L. Mazzei, A. A. Oberai, and A. A. Wray, "The multiscale formulation of large eddy simulation: decay of homogeneous turbulence," *Phys. Fluids*, vol. 13, pp. 505–512, 2001.

[4] S. Brasile, R. Casciaro, and G. Formica, "Multilevel approach for brick masonry walls - Part I: A numerical strategy for the nonlinear analysis," *Comput. Method. Appl. M.*, vol. 196, pp. 4934–4951, 2007.

[5] C. Miehe and C. G. Bayreuther, "On multiscale FE analyses of heterogeneous structures: From homogenization to multigrid solvers," *Int. J. Numer. Meth. Eng.*, vol. 71, pp. 1135–1180, 2007.

[6] F. Feyel, "A multilevel finite element method (FE^2) to describe the response of highly nonlinear structures using generalized continua," *Comput. Method. Appl. M.*, vol. 192, pp. 3233–3244, 2003.

[7] O. R. Astley and V. Hayward, "Real-time finite-elements simulation of general visco-elastic materials for haptic presentation," in *ROS '97, IEEE/RJS Int. Conference on Intelligent Robots and Systems*, 1997, pp. 52–57.

[8] M. Müller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler, "Stable real-time deformations," in *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*. New York, NY, USA: ACM, 2002, pp. 49–54.

[9] G. Salerno and G. deFelice, "Continuum modeling of periodic brickwork," *International Journal of Solids and Structures*, vol. 46, no. 5, pp. 1251–1267, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/B6VJS-4TX33SP-3/2/3ee7f6bbe651cf544e768b4d8c6ba2bc

[10] G. Formica, V. Sansalone, and R. Casciaro, "A mixed solution strategy for the nonlinear analysis of brick masonry walls," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 51–52, pp. 5847–5876, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/B6V29-478HYRR-1/2/ee021cbf69514b64abc145f7629b969c

[11] T. H. H. Pian and K. Sumihara, "Rational approach for assumed stress finite elements," *International Journal for Numerical Methods in Engineering*, vol. 20, no. 9, pp. 1685–1695, 1983.

[12] O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method for Solid and Structural Mechanics*, 6th ed. Butterworth-Heinemann, 2005.

[13] S. Brasile, R. Casciaro, and G. Formica, "Multilevel approach for brick masonry walls - Part II: On the use of equivalent continua," *Comput. Method. Appl. M.*, vol. 196, pp. 4801–4810, 2007.

[14] I. Carol, M. Jirásek, and Z. Bažant, "A thermodynamically consistent approach to microplane theory. part i: free energy and consistent microplane stresses," *Int. J. Solids Struct.*, vol. 38, pp. 2921–2931, 2001.

[15] E. Kuhl, P. Steinmann, and I. Carol, "A thermodynamically consistent approach to microplane theory. Part II: Dissipation and inelastic constitutive modelling," *Int. J. Solids Struct.*, vol. 38, pp. 2933–2952, 2001.

[16] S. Brasile, R. Casciaro, and G. Formica, "Finite element formulation for nonlinear analysis of masonry walls," *Comput. Struct.*, vol. 88, no. 3-4, pp. 135–143, 2010.

[17] E. Riks, "An incremental approach to the solution of snapping and buckling problems," *Internat. J. Solids and Structures*, vol. 15, no. 7, pp. 529–551, 1979.

[18] D. Goldfarb and A. Idnani, "A numerically stable dual method for solving strictly convex quadratic programs," *Mathematical Programming*, vol. 27, 1983.

[19] G. Magenes and G. Calvi, "In-plane seismic response of brick masonry walls," *Earthquake Engineering & Structural Dynamics*, vol. 26, no. 11, pp. 1091–1112, 1997.

[20] P. A. Coe, A. Mitra, S. M. Gibson, D. F. Howell, and R. B. Nickerson, "A study of geodetic grids for the continuous, quasi real time alignment of the atlas semiconductor tracker," in *Proceedings of the 7th International Workshop on Accelerator Alignment*, November 2002.

[21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Belmont, MA: Athena Scientific, 1997.

[22] X.-H. Sun and L. M. Ni, "Another view on parallel speedup," in *Supercomputing '90: Proceedings of the 1990 ACM/IEEE conference on Supercomputing*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1990, pp. 324–333.

[23] M. F. Adams, H. H. Bayraktar, T. M. Keaveny, and P. Papadopoulos, "Ultrascalable implicit finite element analyses in solid mechanics with over a half a billion degrees of freedom," in *SC '04: Proceedings of the 2004 ACM/IEEE conference on Supercomputing*. Washington, DC, USA: IEEE Computer Society, 2004, p. 34.

[24] J. Azevedo and G. Sincraian, "Modelling the seismic behaviour of monumental masonry structures," in *UNESCO International Millennium Congress (Archi 2000)*, 2000.

[25] A. Cecchi, G. Milani, and A. Tralli, "A Reissner-Mindlin limit analysis model for out-of-plane loaded running bond masonry walls," *Int. J. Solids Struct.*, vol. 44, pp. 1438–1460, 2007.

[26] A. DiCarlo and S. Quiligotti, "Growth and balance," *Mech. Res. Commun.*, vol. 29, pp. 449–456, 2002.

[27] L. Gambarotta and S. Lagomarsino, "A microcrack damage model for brittle materials," *Int. J. Solids Struct.*, vol. 30, pp. 177–198, 1993.

[28] P. B. Lourenço, G. Milani, A. Tralli, and A. Zucchini, "Analysis of masonry structures: review of and recent trends in homogenization techniques," *Can. J. Civil Eng.*, vol. 34, pp. 1443–1457, 2007.

[29] H. R. Lofti and P. B. Shing, "An appraisal of smeared crack models for masonry shear wall analysis," *Comput. Struct.*, vol. 41, pp. 413–425, 1991.

[30] T. J. Massart, R. H. J. Peerlings, and M. G. D. Geers, "An enhanced multi-scale approach for masonry wall computations with localization of damage," *Int. J. Numer. Meth. Eng.*, vol. 69, pp. 1022–1059, 2007.

[31] G. Magenes and G. M. Calvi, "In-plane seismic response of brick masonry walls," *Earthquake Eng. Struc.*, vol. 26, pp. 1091–1112, 1997.

[32] G. Magenes, G. M. Calvi, and R. Kingsley, "Seismic Testing of a full-scale, two-story masonry building: test procedure and measured experimental response," in *Experimental and Numerical Investigation on a Brick Masonry Building Prototype - Numerical Prediction of the Experiment, Report 3.0, G.N.D.T.*, 1995.

[33] U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*. Academic Press, 2001.

[34] E. Oñate, "Multiscale computational analysis in mechanics using finite calculus: an introduction," *Comput. Method. Appl. M.*, vol. 192, pp. 3043–3059, 2003.

[35] M. E. Gurtin and B. D. Reddy, "Alternative formulations of isotropic hardening for Mises materials, and associated variational inequalities," *Continuum Mechanics and Thermodynamics*, vol. 21, no. 3, pp. 237–250, 2009. [Online]. Available: http://dx.doi.org/10.1007/s00161-009-0107-3

A Novel Local Search Algorithm for Knapsack Problem

Mostafa Memariani
Department of Electrical Engineering,
Ferdowsi University of Mashhad
Mashhad, Iran
E-mail: mostafa.memariani@stu-mail.um.ac.ir

Kambiz Shojaee Ghandeshtani
Department of Electrical Engineering,
University of Tehran
Tehran, Iran
E-mail: k.shojaee@ece.ut.ac.ir

Ahmad Madadi
Department of Computer Engineering,
University of Amir Kabir
Tehran, Iran
E-mail: a.madadi@gmail.com

Mohammad Mohsen Neshati
Department of Electrical Engineering,
Ferdowsi University of Mashhad
Mashhad, Iran
E-mail: mohsen_neshati@stu-mail.um.ac.ir

Abstract— **Knapsack problem is an integer programming that is generally called "Multidimentional Knapsack". Knapsack problem is known as a NP-hard problem. This paper is an introduction to a new idea for solving one-dimentional knapsack that with defining the "Weight Value Index", "Sorting" and "Smart local search" forms a new algorithm. This algorithm is mathematically formulated and has run on 5 sample problems of one-dimentional knapsack, that in most of them the result is close to the optimum. The results show that this method by comparison with the others recently published in this field, despite of its simplicity, has enough required functionality in order to get the result on the tested items.**

Keywords-Artificial intelligence; NP-hard; Knapsack problem; Combinational optimization.

I. INTRODUCTION

Knapsack problem is an integer programming that is in general called "Multidimentional Knapsack". Knapsack problem is known as a NP-hard problem [1]. One-dimentional knapsack problem with "constant weight group" is a special form of multidimensional knapsack. For one-dimensional knapsack in comparison with multidimensional knapsack, more precise evolutionary algorithms have been studied. Most of the researches is regarding to one-dimentional knapsack problem. For further information about knapsack problem and different precise algorithms, please refer to [2]-[4].

The reason for naming this problem to "knapsack" is because of its similarity to making decision for a mountain climber to pack his knapsack. The person should decide the optimum combination in choosing his accessories for knapsack in a way that according to the knapsack capacity, he should select items with more value (profit). This kind of problems is of combinational optimization problems family.

For several past years, precise methods such as Branch and Bound have used for solving knapsack problem [22]. In recent years, and with the development of smart optimization and evolutionary algorithms, solving more difficult problems is now possible, such that in addition to reducing the time for achieving results close to the optimum, it has increased the accuracy in solving knapsack problem. Therefore evolutionary algorithms and more definitely decoder-based evolutionary algorithms are widely used in solving knapsack problem [5], [6]. Their advantage over the more traditional direct representation of the problem is their ability to always generate and therefore carry out evolution over feasible candidate solutions, and thus focus the search on a smaller more constrained search space.

Many researchers have struggled in developing evolutionary methods for knapsack problems. From them, we can name some modern evolution methods like tabu search [7], [8], genetic algorithm [9], [10] and simulated annealing [11], [12] that in most cases show good results. In recent years, genetic algorithms show that it is the best method for solving large knapsack problems and in general 0-1 integer programming problems [13], [14].

The knapsack repeatedly is used in different processing models like processor allocation in distributed systems [15], manufacturing in-sourcing [16], asset-backed securitization [17], combinatorial auctions [18], computer systems design [19], resource-allocation [20], set packing [21], cargo loading [22], project selection [23], cutting stock [24] and capital budgeting (where project has profit and consume units of resource. The goal is to determine a subset of the projects such that the total profit is maximized and all resource constraints are satisfied) [25].

Another type of knapsack is Quadratic Knapsack Problem (QKP) [26]. In the Quadratic Knapsack Problem, an object's value density is the sum of all the values associated with it divided by its weight. It can be used in finance [27], VLSI design [28] and location problems [29].

In the second part of this paper, we will describe the knapsack problem; in third part, the proposed algorithm will be introduced. In the forth part, algorithm simulation and comparison of results have been presented and we will conclude in the final part.

II. PROBLEM DESCRIPTION

Suppose that some items are available and each item has a weight of 'w_i' and a value of 'v_i'. In knapsack problem,

ISBN: 978-1-61208-101-4

weight restriction is defined in a way that the total weight of selected items should be less than knapsack capacity. The goal in this problem is finding a subset of items in a way that they have the most total value and also satisfy the knapsack capacity constraint.

For mathematically defining the mentioned concepts, we have:

$$\max\left\{\sum_{i=1}^{n} v_i x_i : \sum_{i=1}^{n} w_i x_i \le b, x_i = 0 \text{ or } 1\ , i = 1,...,n\right\} \quad (1)$$

In formula (1), 'n', 'v_i' and 'w_i' are number of items, value of item 'i' and weight of item 'i', respectively. In the above formula, 'b' is the knapsack capacity and x_i is the algorithm input array. If the element is chosen, the x_i is 1 and otherwise is 0.

As formula (1) shows, the goal is to maximize the goal function with the given conditions. In the next section, the proposed algorithm for solving the knapsack problem will be introduced.

III. PROPOSED ALGORITHM

The presented method for solving the knapsack problem is based on statistical operations on data and combining it with artificial intelligence methods. In this method we have a set of weight and value data groups that are related in pairs and each of data shows the weight and the value of an item. The goal of this method in first stage is introducing each item with a new coefficient that would be a combination of its value and weight. With the help of this new index, the chance of selecting an item will be defined. The proposed algorithm with enough experience and iteration in changing the method of selecting based on the weight-value index and in a converged evolutionary process will provide results close to optimum. The stage of process on data for achieving a real close result to optimum will be as follows:

- According to the point that the goal of knapsack problem is to take the sum of values to the maximum and satisfy the weight constrain of knapsack, for converting 2 item dependents to one dependent, we will use the general form of (Value^{p1} / Weight^{p2}) that the p1 and p2 are the power of values and weights, respectively. The best value of them will be different depending on the number of items and their dispersion that with scanning the power of values and weights in the above combinational index and calculating the sum of selected item values until satisfaction of the weight constrain, we can have the best selection for the powers of mentioned formula in the beginning of the algorithm. This value would be the "weight-value index" of items.
- Next step of solving the problem is sorting items based on their weight-value index and generating initial result that would be close to optimum. In this selection, the items with higher weight-value prioritized for selection and selection of items will continue until the knapsack capacity is full.
- Because of the used method in first stage for generating weight-value index is not precise. The probability of error in the second stage would be existent as well. It is important to know that the probability of the error in selecting items based on proposed priority that is weight-value index would increase as we get closer to final stages. The probability of such errors is in the moment that the knapsack is getting filled with lower weight-value index of items. Therefore in this stage that is the main part of algorithm, we will replace the items with similar weight-value index in the final stages of selecting items. In this stage we will gradually increase the boundary of searching. In this part of algorithm we will study different results to achieve the best one.

In this intelligence searching algorithm, in addition to previous stages, we achieved the better results by the helping of some sort of modifications and corrections. For instance we can find the minimum of the selected items by dividing the knapsack capacity to the item with the highest weight. We can get to the scope of weight-value index results or in fact, items that their probability of being among the optimum answer is very high.

The main foundation of this method has been introduced above in 3 steps and the algorithm pseudocode would be as follows.

IV. ALGORITHM FORMULA

s1- Determining optimum powers for achieving optimum weight-value index by scanning from 0 to 2 with the step of 0.1 and selecting the best powers for the proportion of value to weight of items by selecting items until the knapsack is completely full. This selection is based on a way that the weight-value index priority, selected items value should be higher than the other powers that has been scanned for the proportion of value to weight.

s2- Random search around the selected power from s1 with the Radius boundary of $\alpha = 0.5$.

s3- Sorting and selecting items based on weight-value of s2 until the completion of knapsack capacity sequence length accepting items l1 and rejected items the l2

s4- Fixing items from vector value of s3 that is higher than Mean and standard deviation values of weight vector elements as selected items and random replacement of the rest selected items from s3 and rejected items as well around the last selection of s3 with the radius of 0.1 items and l1 and l2.

s5- Studying selection rule of selecting minimum items equal to dividing the maximum capacity of knapsack to the highest weight of items value and increasing the length of sequence of accepted items (l1) until satisfying the minimum selection rule.

s6- comparing the answers and the results of the current selections with the best achieved result and replacing it with the previous if that is a better answer.

s7- α=0.5 +α

s8- reduce the radius boundary of optimum power index with a coefficient of 0.9.

s9- repeating s1 to s9 while α=1 and radius boundary has reached to boundary interval.

V. RESULTS AND COMPARISON

In this part, the result of running algorithm on set of data that was given in [32]-[33] is analyzed. Five sample problems are proposed in [30]-[32] for testing the algorithm. In [30], e2, e3 and e5 samples have been solved with the different methods.

In [31], samples e1 to e4 and in [32], samples e1 and e2 have been studied. The samples e1 to e5 have 10, 20, 50,100 and 100 objects respectively. It is obvious that the samples with a greater number of objects are more complicated than samples with less number of objects and they are more difficult and more time consuming to solve.

In Table 1, the best obtained results in the relevant papers have been compared with the results of our proposed algorithm. As it is clear in Table 1, the proposed algorithm that is called Wise Experiencing Knapsack Problem (WEKP) has resulted acceptable answers.

The algorithm that has been introduced in this paper has improved the results of greedy and simple evolutionary algorithms by rate of 0.9 and 1.9 percent to the best answer. The algorithm of [31], which is a combination of greedy and genetic algorithms, has been improved the results of e1-e4 problems by rate of 0.7 and 0.2. The algorithm mentioned in [32] is an enhanced form of ACO that the results shows 0.2 percent improvement in e1 and e2 problems as well.

The results after simulating the proposed algorithm by this paper show that the results have been improved by 0.16 percent regarding to [30], 0.05 percent to [31] and 0.9 % regarding to [32].

In Table 1, we can see that for the 3rd sample problem we have achieved a result that was never achieved in other papers up to now.

In Table 2, the best, average and the worst answers for 20 times run for every sample has been given. Also, the sequence of the best obtained results for every sample has been determined as a string containing 0 and 1, where 0 means no selection and 1 stands for selecting the i^{th} object.

As it is illustrated in Table 2, even the average of the responses is very close to the optimum response and these responses acquire in an acceptable time period.

The mean time for running every problem on a pentium4 and a processor of 1.8GHz speed and 512MB of ram with the MATLAB 7.7 software is given answers.

VI. CONCLUSION

This paper is an introduction to a novel idea for solving one-dimensional knapsack problem by defining weight-value index and sorting; as a consequence, a new algorithm was proposed. This algorithm is mathematically formulated and has run on 5 samples regarding to one-dimensional knapsack that in most of them the answers are near to optimum.

The results shows that this method in comparison with the recent works published in this field, despite of its simplicity is functional enough to achieve acceptable results in tested problems.

ACKNOWLEDGMENT

This work is supported by Nano-Age Technology Group in Mashhad, Iran (www.nanoage.ir).

REFERENCES

[1] M. Garey and D. Johnson, "Computers and intractability: a guide to the theory of NPcompleteness," San Francisco: W. H. Freeman, 1979.

[2] S. Martello and P. Toth, “Knapsack Problems: Algorithms and Computer Implementations,” Wiley, New York, 1990.

[3] S. Martello, D. Pisinger, and P. Toth, “New trends in exact algorithms for the 0–1 knapsack problem,” European Journal of Operational Research, vol. 123,No. 2, pp. 325–336, 1999.

[4] D. Pisinger, “Contributed research articles: a minimal algorithm for the bounded knapsack problem”, ORSA Journal on Computing, vol. 12, No. 1, pp. 75–84, 2000.

[5] J. Gottlieb, “Permutation-Based evolutionary algorithms for multidimensional knapsack problem,” Proc. of ACM Symp. on Applied Computing, 2000.

[6] G. R. Raidl, “An improved genetic algorithm for the multiconstrained 0-1 knapsack problem,” Proc of 1998 IEEE Congress on Evolutionary Computation, pp. 207 – 211, 1998.

[7] F. Glover and G. A. Kochenberger, "Critical event tabu search for multidimensional knapsack problems," Kluwer Academic Publishers, pp. 407–427, 1996.

[8] S. Hanafi and A. Fréville, "An efficient tabu search approach for the 0-1 multidimensional knapsack problem," European Journal of Operational Research, vol. 106, pp. 659–675, 1998.

[9] Chu and J. Beasley, "A genetic algorithm for the multiconstrained knapsack problem," Journal of Heuristics, vol. 4, pp. 63–86, 1998.

[10] G. R. Raidl, “Weight-Codings in a genetic algorithm for the multiconstraint knapsack problem,” Proc of 1999 IEEE Congress on Evolutionary Computation, pp. 596-603, 1999.

[11] C. Reeves, “Modern Heuristic Techniques for Combinatorial Problems," McGraw-Hill Book Company Europe, 1995.

[12] A. Drexl. "A simulated annealing approach to the multiconstraint zero-one knapsack problem". Computing, vol. 40, pp. 1–8, 1988.

[13] Y. Sun and Z. Wang, “The Genetic Algorithm for 0–1 Programming with Linear Constraints,” Proc. of the 1st ICEC’94, Orlando,FL, edited by D. B. Fogel, pp. 559–564, 1994.

[14] R. Hinterding, “Mapping, order-independent genes and the knapsack problem”, Proc. of the 1st IEEE International Conference on Evolutionary Computation 1994, Orlando, FL, edited by D. B. Fogel, pp. 13–17, 1994.

[15] B. Gavish and H. Pirkul, “Allocation of data bases and processors in a distributed computing system Management of Distributed Data Processing,” vol. 31, pp. 215–231, 1982.

[16] N. S. Cherbaka, R. D. Meller, and K. P. Ellis, “Multidimensional knapsack problems and their application to solving manufacturing insourcing problems,” Proc. of the Annual Industrial Engineering Research Conference, Houston,TX, May 16-19, 2004.

[17] R. Mansini and M. Speranza, “A multidimensional knapsack model for the asset-backed securitization,” Journal of Operational Research Society, vol. 53, pp. 822-832, 2002.

[18] S. DeVries, and R. Vohra, “Combinatorial Auctions: A Survey,” Northernwestern University Technical Report, Evanston, IL (2000).

[19] C. Ferreira, M. Grotschel, S. Kiefl, C. Krispenz, A. Martin, and R. Weismantel., “Some integer programs arising in the design of mainframe computers,” ZORMethods Models Operations Research, vol. 38, No. 1, pp. 77-110, 1993.

[20] E. Johnson, M. Kostreva, and U. Suhl, “Solving 0 – 1 integer programming problems arising from large scale planning models,” Operations Research, vol. 33, pp. 805-819, 1985.

[21] G. Fox and G. Scudder, “A heuristic with tie breaking for certain 0 – 1 integer programming models,” Naval Research Logistics, vol 32, No. 4, pp. 613-623, 1985.

[22] W. Shih, “A branch and bound method for the multiconstraint zero-one knapsack problems,” Journal of the Operations Research Society, vol. 30, pp. 369-378, 1979.

[23] C. Peterson, “Computational experience with variants of the balas algorithm applied to the selection of research and development projects,” Management Science, vol. 13, pp. 736-750, 1967.

[24] P. Gilmore and R. Gomery, “The theory and computation of knapsack functions,” Operations Research, vol. 14, pp. 1045-1074 1966.

[25] J. Lorie and L. Savage, “Three problems in capital rationing,” journal of business, vol. 28, pp. 229-239, 1955.

[26] B. A. Julstrom," Greedy, genetic, and greedy genetic algorithms for the quadratic knapsack problem," GECCO’05, Washington, DC, USA, pp.607-614, June 25–29, 2005.

[27] D. L. Laughhunn, "Quadratic binary programming with applications to capital budgeting problems", Operations Research, vol. 18, pp. 454–461, 1970.

[28] C. E. Ferreira, A. Martin, C. C. de Souza, R. Weismantel, and L. A. Wolsey," Formulations and valid inequalities for node capacitated graph partitioning," Mathematical Programming, vol. 74, pp. 247–266, 1996.

[29] J. Rhys, "A selection problem of shared fixed costs and network flows," Management Science, vol. 17, pp. 200–207, 1970.

[30] K. Li, Y. Jia, W. Zhang, and Y. Xie, "A new method for solving 0-1 knapsack problem based on evolutionary algorithm with schema replaced," Proceedings of the IEEE, International Conference on Automation and Logistics Qingdao, China, pp. 2569-2571, Sep. 2008.

[31] Y. Shao, H. Xu, and W. Yin, "Solve zero-one knapsack problem by greedy GA," IEEE 2009 -International Workshop on Intelligent Systems and Applications.

[32] P. Zhao, P. Zhao, and X. Zhang, "A new ant colony optimization for the knapsack problem," Computer-Aided Industrial Design and Conceptual Design, 2006, CAIDCD '06, 7th International Conference on 17-19 Nov. 2006.

TABLE 1. COMPARATIVE RESULTS BY OTHER HEURISTIC METHODS

	[30]			**[31]**			**[32]**		**WEKP**
	Greedy algorithm	***Simple evolutionary algorithm***	***evolutionary algorithm with schema replace***	***Greedy algorithm (GA)***	***Standard genetic algorithm (SGA)***	***Greedy genetic algorithm (GGA)***	***Basic ACO***	***Improved ACO***	***Proposed method (Best result)***
e1	-	-	-	295	295	295	292	295	295
e2	1023	1042	1042	1024	1037	1042	1022	1024	1042
e3	3095	3077	3103	3077	3103	3112	-	-	**3119**
e4	-	-	-	5372	5365	5375	-	-	5372
e5	26380	25848	26559	-	-	-	-	-	26553

TABLE 2. BEST RESULTS FOR FIVE SAMPLE PROBLEM

	Example 1	**Example 2**	**Example 3**	**Example 4**	**Example 5**
No. of Objects	10	20	50	100	100
Best	295	1042	3119	5372	26553
Mean (20 runs)	295	1040.5	3106	5367.4	26553
worst	295	1037	3115.1	5360	26553
Mean time (s)	7.1	22.8	13.08	23.07	45.33
Best chromosome	111000111	101111110 10111100000	110101011110100110 110111111111000010 11011000000010	111111111011111111111001110111011 000101001110111110010110101000001 00001000011001001010000011000000000	1111111111111111111111111111111111 1111111111110111111110100010110110 11111110001110111000000000000000

Geometric Error Estimation

Houman Borouchaki
Project-team GAMMA 3
UTT
Troyes, France
Email: houman.borouchaki@utt.fr

Patrick Laug
Project-team GAMMA 3
INRIA
Paris - Rocquencourt, France
Email: patrick.laug@inria.fr

***Abstract*—An essential prerequisite for the numerical finite element simulation of physical problems expressed in terms of PDEs is the construction of an adequate mesh of the domain. This first stage, which usually involves a fully automatic mesh generation method, is then followed by a computational step. One can show that the quality of the solution strongly depends on the shape quality of the mesh of the domain. At the second stage, the numerical solution obtained with the initial mesh is generally analyzed using an appropriate a posteriori error estimator which, based on the quality of the solution, indicates whether or not the solution is accurate. The quality of the solution is closely related to how well the mesh corresponds to the underlying physical phenomenon, which can be quantified by the element sizes of the mesh. An a posteriori error estimation based on the interpolation error depending on the Hessian of the solution seems to be well adapted to the purpose of adaptive meshing. In this paper, we propose a new interpolation error estimation based on the local deformation of the Cartesian surface representing the solution. This methodology is generally used in the context of surface meshing. In our example, the proposed methodology is applied to minimize the interpolation error on an image whose grey level is considered as being the solution.**

***Keywords*-a posteriori error estimation; interpolation error; mesh adaptation; surface curvature.**

I. Introduction

Different kinds of estimators are available to *a posteriori* control the error made on a finite element solution [1]. Using such an estimator, it is possible to control the mesh by *h-adaptation* so that the corresponding solution of the PDE problem has a given accuracy. Some of these estimators are based on the interpolation error (and, in this sense, are purely geometric since they ignore the nature of the operator considered). This kind of estimators has been studied by many authors [2]–[5]. However, most of these studies lie on the fact that a parameter h, representing the size of the elements, is small or tends to zero, and thereby they are asymptotic studies. The estimator is thus based on appropriate Taylor expansions, and gives in this manner some indications on the admissible size h. Nevertheless, as this size is not necessarily small, we propose a novel approach which, although closely related to the previous ones, does not assume any particular hypothesis on this parameter, and therefore is probably more justified. Our approach is besides rather similar, in its spirit, to certain solutions used in a different domain, namely the mesh generation of parametric patches [6]–[8].

Section 2 gives the mathematical formulation of the problem and reviews the related works. Section 3 introduces a new class of measures to quantify the interpolation error depending on the local deformation or curvature of the Cartesian surface corresponding to the solution. A numerical example is illustrated in Section 4 and finally, the last section provides a brief conclusion.

II. Definition of the problem and state of the art

Let Ω be a domain of R^d (with d = 1, 2 or 3) and let $\mathcal{T}$ be a simplicial mesh of Ω composed of linear simplices P^1 or quadratic simplices P^2. We suppose that, in order to solve a problem given in terms of PDEs on Ω, we have made a finite element computation on Ω using $\mathcal{T}$, and we have obtained the scalar solution $u_{\mathcal{T}}$. Denoting by u the exact solution, the problem firstly consists in evaluating the gap $e_{\mathcal{T}} = u - u_{\mathcal{T}}$ between u and $u_{\mathcal{T}}$ representing the error involved by the finite element solution, and secondly deducing (in general by bounding this gap) another mesh $\mathcal{T}'$ such that the estimated gap between u and the solution $u_{\mathcal{T}'}$ using mesh $\mathcal{T}'$ is bounded by a given threshold. Several points must be more precisely explained:

- how to quantify the gap $e_{\mathcal{T}}$ between u and $u_{\mathcal{T}}$?
- how to use the latter information for building a new mesh on which the gap between the corresponding finite element solution and the exact solution is bounded by a given threshold?

The solution $u_{\mathcal{T}}$ obtained by the finite element method is not interpolating (*i.e.* the solution obtained at the nodes of $\mathcal{T}$ does not coincide with the exact value of u at these nodes). Moreover, for each element of the mesh, it cannot be guaranteed that the solution $u_{\mathcal{T}}$ coincide with the exact value of u at one point (at least) of the element. Then, it seems difficult to explicitly quantify the gap $e_{\mathcal{T}}$. However, the direct study of this gap has been dealt in several works [9]. But, in the general case, its quantification remains an open problem. Consequently, other indirect approaches have been proposed to quantify or rather bound this gap. Let us denote by $\widetilde{u}_{\mathcal{T}}$ the function interpolating u on the mesh $\mathcal{T}$

(which is a piecewise linear or quadratic function, depending on the degree of the elements of $\mathcal{T}$) and by $\widetilde{e}_{\mathcal{T}}$ the gap $u-\widetilde{u}_{\mathcal{T}}$ between u and $\widetilde{u}_{\mathcal{T}}$, called the *interpolation error* on u along mesh $\mathcal{T}$. To be able to quantify the gap $e_{\mathcal{T}}$, we suppose the following relation holds (Céa's lemma):

$$||e_{\mathcal{T}}|| \;\leq\; C\,||\widetilde{e}_{\mathcal{T}}||$$

where $||.||$ denotes a norm and C is a constant not depending on $\mathcal{T}$. In other words, we suppose that the finite element error is bounded by the interpolation error. The original problem is then simplified by considering the following problem: given an interpolation $\widetilde{u}_{\mathcal{T}}$ of u along a mesh $\mathcal{T}$, how to build another mesh $\mathcal{T}'$ for which the interpolation error is bounded by a given threshold? As $\widetilde{u}_{\mathcal{T}}$ can be seen as a discrete representation of u, the problem now reduces to a characterization of meshes for which the interpolation error is bounded by this threshold. This problem has been the subject of several studies (see for instance [5]) and, in most of them, the examination of a "measure" of the interpolation error provides some constraints associated with the mesh elements. In the context of mesh adaptation methods, h-methods or size adaptation are particularly relevant, and the constraints are specified in terms of element sizes. In the following, some classical measures of this error are recalled, as well as resulting constraints on the mesh elements.

To quantify the interpolation error, two kinds of measures can be considered: continuous or discrete. A classical continuous measure of this error is the square of the L^2 norm of $\widetilde{e}_{\mathcal{T}}$:

$$||\widetilde{e}_{\mathcal{T}}||^2_{L^2} = \int_{\mathcal{T}} \widetilde{e}^{\,2}_{\mathcal{T}}\,d\omega = \sum_{K\in\mathcal{T}} ||\widetilde{e}_K||^2_{L^2}$$
$$\text{with}\quad ||\widetilde{e}_K||^2_{L^2} = \int_K \widetilde{e}^{\,2}_K\,d\omega\,,$$

where $\widetilde{e}_K$ is the interpolation error on each element K of $\mathcal{T}$, and $d\omega$ is an elementary volume of R^d. In two dimensions, considering linear elements and assuming that the Hessian H_u of u restricted to the elements is constant, Nadler [10] gives an analytical expression of the measure of the interpolation error $||\widetilde{e}_K||^2_{L^2}$ on K as a function of the area A of K and the quantities $d_i = \frac{1}{2}\,a_i^T\,H_u\,a_i$ (second directional derivatives along the edges) where a_i is the vector joining vertices i and $i+1$ of K:

$$\int_K \widetilde{e}^{\,2}_K\,dx\,dy = \frac{A}{180}\left(\left(\sum_i d_i\right)^2 + \sum_i d_i^2\right).$$

Berzins [11] extends this result in three dimensions (for linear elements) and shows (still assuming that the Hessian H_u of u is constant in element K) that:

$$\int_K e^2_{\mathcal{T}}\,dx\,dy\,dz = \frac{V}{420}\left(\left(\sum_i d_i\right)^2 + \sum_i d_i^2 - d_1\,d_4 - d_2\,d_5 - d_3\,d_6\right),$$

where V is the volume of K and quantities d_i are similar to the 2D case. Berzins deduces from this expression a measure of the quality of the elements, and thus characterizes the mesh. However, it is unclear to interpret this information in terms of element size. The extension of these results to the case of an arbitrary Hessian H_u remains open. An alternative measure, well suited to problem solving by the finite element method, consists in considering Sobolev norms of $\widetilde{e}_K$, in particular the H^1 norm whose square is defined by:

$$||\widetilde{e}_K||^2_{H^1} = \int_K \left(\,\widetilde{e}^{\,2}_K + ||\nabla\widetilde{e}_K||^2\,\right)\,d\omega\,,$$

where ∇ represents the gradient and $||.||$ is the usual Euclidian norm. In two dimensions and considering linear elements, Zlamal [12], as also Babuska and Aziz [2], independently propose an upper bound of $||\widetilde{e}_K||^2_{H^1}$ by the seminorm $|u|_2$ of the Sobolev space H^2 whose square is defined by:

$$|u|_2^2 = \left\|\frac{\partial^2 u}{\partial x^2}\right\|^2_{L^2} + 2\,\left\|\frac{\partial^2 u}{\partial x\partial y}\right\|^2_{L^2} + \left\|\frac{\partial^2 u}{\partial y^2}\right\|^2_{L^2}.$$

Indeed, they show that:

$$||\widetilde{e}_K||^2_{H^1} \;\leq\; \Gamma(\theta)\,|u|_2\,,$$

where $\Gamma(\theta)$ is a function depending on the diameter of K and its internal angles. An extension in three dimensions of this relation has been proposed by Krizek [13]. Again, it seems difficult to establish a constraint in terms of element size for this norm. Another measure, which is simpler, consists in considering the L^2 norm of the gradient of $\widetilde{e}_K$. It is given by:

$$||\nabla\widetilde{e}_K||^2_{L^2} = \int_K ||\nabla\widetilde{e}_K||^2\,d\omega\,.$$

An explicit expression of this error measure related to linear elements has been proposed by Bank and Smith [14] in two dimensions in the case where the Hessian H_u is constant in K. An approximation of this expression is given by:

$$||\nabla\widetilde{e}_K||^2_{L^2} \;\approx\; \frac{\sum\limits_i ||a_i||^2\;\sum\limits_i d_i^2}{48\,A}\,.$$

They use this measure for relocating the nodes of the mesh in order to minimize the error.

Among the discrete measures, one can mention the L^∞ norm of the interpolation error, which is defined by:

$$||\widetilde{e}_K||_{L^\infty} = \max_{x\in K}|\widetilde{e}_K(x)|\,,$$

where point x sweeps element K. Similarly, assuming that Hessian H_u is constant on each element, Manzi *et al.* [15] propose an approximation of the measure $||\widetilde{e}_K||_{L^\infty}$ from an expression of error e_K given by D'Azevedo and Simpson [3] for linear elements in two dimensions:

$$||\widetilde{e}_K||_{L^\infty} \approx \frac{\prod\limits_i \delta_i}{16\det(H_u)\,A^2}\,,$$

where $\delta_i = a_i^T\,|H_u|\,a_i$, $|H_u|$ being the absolute value of the Hessian of u. Using this approximation, they show that if the size h of K along all directions verifies $h^T\,|H_u|\,h \leq 3\,\varepsilon$ then $||\widetilde{e}_K||_{L^\infty} \leq \varepsilon$. This size constraint proves well-suited to h-methods and the results obtained by the authors show the simplicity and the efficiency of this method. In the context of surface triangulation by linear elements, Anglada *et al.* [7] propose, in the general case where the Hessian of u is arbitrary, an upper bound of $||\widetilde{e}_K||_{L^\infty}$ given by:

$$||\widetilde{e}_K||_{L^\infty} \leq \frac{2}{9}\sup_{x\in K}\,||\overrightarrow{pq}^T\,H_u(x)\,\overrightarrow{pq}||\,,$$

where point x sweeps element K, p is the vertex of K such that the barycentric coordinate of x in K with respect to p is maximal, and q the intersection point of the straight line $(p\,x)$ with the edge of K opposite to p. They infer that the interpolation error is bounded by a threshold if element K lies in regions defined and centered at the vertices of K. Therefore, these regions can be defined at every points of the domain and then constitute constraints for the element sizes.

According to the above description of different works on the subject (although this list is far from being exhaustive), a discrete measure (linking error bound and mesh element size) seems more appropriate in the scope of error estimation for mesh adaptation. The following section details this issue.

III. A NOVEL APPROACH BASED ON SURFACE GEOMETRY

In this section, we recall the approach proposed by [16] which considers solution u as a Cartesian surface, and we give a new error estimation in the case of anisotropic geometric surface meshing. Let Ω be the computational domain, $\mathcal{T}(\Omega)$ a mesh of Ω, and $u(\Omega)$ the physical solution obtained on Ω using the mesh $\mathcal{T}(\Omega)$. The couple $(\mathcal{T}(\Omega), u(\Omega))$ defines a Cartesian surface $\Sigma_u(\mathcal{T})$. Given $\Sigma_u(\mathcal{T})$, the problem of minimizing the interpolation error consists in defining an optimal mesh $\mathcal{T}_{opt}(\Omega)$ of Ω for which surface $\Sigma_u(\mathcal{T}_{opt})$ would be as smooth as possible. For this purpose, we propose to locally characterize the surface in the neighborhood of a vertex. Two methods are introduced: the first one, based on local deformation, can be applied for an isotropic adaptation while the second one, based on local curvature, is suitable to an anisotropic adaptation.

A. Local deformation of a surface

The main idea consists in locally characterizing the deviation (of order 0) of a surface mesh $\Sigma_u(\mathcal{T})$ in the neighborhood of a vertex with respect to a reference plane, in particular the tangent plane to the surface at this vertex. This deviation can be quantified by considering the Hessian along the normal to the surface (*i.e.* the second fundamental form of the surface).

Let P be a vertex of the solution surface $\Sigma_u(\mathcal{T})$. Locally, in the neighborhood of P, this surface admits a parametric representation $\sigma(u,v)$, (u,v) being the parameters, with $P = \sigma(0,0)$. The Taylor expansion at order 2 to σ in the neighborhood of P gives:

$$\begin{aligned}\sigma(u,v) &= \sigma(0,0) + \sigma'_u\,u + \sigma'_v\,v \\ &\quad + \frac{1}{2}\,(\sigma''_{uu}\,u^2 + 2\,\sigma''_{uv}\,u\,v + \sigma''_{vv}\,v^2) + o(u^2+v^2)\,e\,,\end{aligned}$$

where $e = (1,1,1)$. If $\nu(P)$ denotes the normal to the surface at P, then the quantity $\langle \nu(P), (\sigma(u,v) - \sigma(0,0))\rangle$ ($\langle .,.\rangle$ denoting the dot product) representing the gap between point $\sigma(u,v)$ and the tangent plane at P, expressed by:

$$\begin{aligned}&\frac{1}{2}\,(\langle\nu(P),\sigma''_{uu}\rangle\,u^2 + 2\,\langle\nu(P),\sigma''_{uv}\rangle\,u\,v + \langle\nu(P),\sigma''_{vv}\rangle\,v^2) \\ &\qquad\qquad + o(u^2+v^2)\,,\end{aligned}$$

is therefore proportional to the second fundamental form of the surface for u^2+v^2 small enough.

The local deformation of the surface at P is defined as the maximum gap between vertices adjacent to P and the tangent plane to the surface at P. If (P_i) denotes these vertices, then the local deformation $\varepsilon(P)$ of the surface at P is given by:

$$\varepsilon(P) = \max_i\,\langle\nu(P), \overrightarrow{PP_i}\rangle\,.$$

Consequently, the optimal mesh of Ω for $\Sigma_u(\mathcal{T})$ is a mesh whose size at each node p is inversely proportional to $\varepsilon(P)$ where $P = (p, u(p))$. More formally, the optimal size $h_{opt}(p)$ associated with a node p reads:

$$h_{opt} = h(p)\,\frac{\varepsilon}{\varepsilon(P)}\,,$$

where ε denotes the imposed deviation threshold and $h(p)$ the element size in the neighborhood of p in mesh $\mathcal{T}(\Omega)$.

It can be noticed that the local deformation is a very simple characterization of the local deviation of the surface, which does not require the explicit computation of the Hessian of the solution. The only disadvantage of this measure is that the resulting adaptive meshes can only be isotropic. In the same context (local deviation minimization), the notion of curvature provides a more precise and anisotropic analysis of this deviation.

B. Local curvature of a surface

The analysis of the local geometric curvature of the surface representing the solution can be used to minimize also the deviation (of order 1) between the tangent planes of the interpolating solution and those of the exact solution. Indeed, in the context of isotropic surface mesh generation, we show [8] that the two deviations of order 0 and 1 of the surface are bounded by a given threshold if, at any point of the surface, the size of the surface elements is proportional to the minimal radius of curvature. Let $P = (p, u(p))$ be a vertex of $\Sigma_u(\mathcal{T})$, let $\rho_1(P)$ and $\rho_2(P)$ with $\rho_1(P) \leq \rho_2(P)$ be the two principal radii of curvature at P, and let $(\overrightarrow{e_1}(P), \overrightarrow{e_2}(P))$ be the two unit vectors in the corresponding principal directions. The ideal size for a surface mesh element at P is [8]:

$$h^{\Sigma}_{opt}(P) = \gamma\,\rho_1(P)\,,$$

where γ is a coefficient depending on the imposed deviation threshold. This size is defined in the tangent plane to the surface at P. In the reference system $(P,\, \overrightarrow{e_1}(P),\, \overrightarrow{e_2}(P))$ of this plane, the ideal size in a given direction is a vector $\overrightarrow{v}(P) = h_1^{\Sigma}\overrightarrow{e_1}(P) + h_2^{\Sigma}\overrightarrow{e_2}(P)$ whose components h_1^{Σ} and h_2^{Σ} satisfy the following relation:

$$\begin{pmatrix} h_1^{\Sigma} & h_2^{\Sigma} \end{pmatrix} \quad \frac{\mathcal{I}_2}{\gamma^2\,\rho_1^2(P)} \quad \begin{pmatrix} h_1^{\Sigma} \\ h_2^{\Sigma} \end{pmatrix} = 1\,.$$

This expression, where $\mathcal{I}_2$ denotes the 2×2 identity matrix, represents the equation of a circle with center P and radius $\gamma\,\rho_1(P)$ in the tangent plane to the surface at P. By an orthogonal projection of this circle in the plane of Ω, the size constraint at p is obtained. If $\overrightarrow{v_1}(p)$ and $\overrightarrow{v_2}(p)$ are the respective orthogonal projections of $\overrightarrow{e_1}(P)$ and $\overrightarrow{e_2}(P)$ in the plane of Ω, then this size constraint in the reference system $(p,\, \overrightarrow{i},\, \overrightarrow{j})$ ($\overrightarrow{i} = (1, 0)$ and $\overrightarrow{j} = (0, 1)$) is given by:

$$\begin{pmatrix} h_1 & h_2 \end{pmatrix} \quad \mathcal{P}^T \quad \frac{\mathcal{I}_2}{\gamma^2\,\rho_1^2(P)} \quad \mathcal{P} \quad \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} = 1\,,$$

where $\mathcal{P} = \begin{pmatrix} \overrightarrow{v_1}(p) & \overrightarrow{v_2}(p) \end{pmatrix}^{-1}$ and (h_1, h_2) are the coordinates in the reference system $(p,\, \overrightarrow{i},\, \overrightarrow{j})$ of the projection of the ideal size vector $\overrightarrow{v}(P)$ in the plane of Ω. This relation defines, among others, a metric (generally anisotropic) at p.

This metric may produce an important number of elements owing to the isotropic feature of surface elements. To minimize this number of elements, and in the context of anisotropic geometric surface meshing, we have established [17] a relation which is similar to the isotropic case and depends on both principal radii of curvature. Now, the ideal size of the surface elements is given by a metric, called geometric, which can be expressed at a vertex P of $\Sigma_u(\mathcal{T})$:

$$\begin{pmatrix} h_1^{\Sigma} & h_2^{\Sigma} \end{pmatrix} \begin{pmatrix} \dfrac{1}{\gamma^2\,\rho_1^2(P)} & 0 \\ 0 & \dfrac{1}{\eta^2\,\rho_2^2(P)} \end{pmatrix} \begin{pmatrix} h_1^{\Sigma} \\ h_2^{\Sigma} \end{pmatrix} = 1\,,$$

where $\gamma = 2\,\sqrt{\varepsilon\,(2-\varepsilon)}$, $\eta = 2\,\sqrt{\varepsilon\,\dfrac{\rho_1(P)}{\rho_2(P)}\,(2-\varepsilon\,\dfrac{\rho_1(P)}{\rho_2(P)})}$, in which ε is the prescribed gap in direction $\overrightarrow{e_1}(P)$. This relation generally represents an ellipse in the tangent plane to the surface at P which contained a circle in the isotropic case. Again, by projecting this ellipse in the plane of Ω, the corresponding metric at p in this plane is obtained. This measures also provide a means to control the interpolation error in H^1 norm (bounding the error on the solution but also on its derivatives), and thereby seams more adequate compared to an isotropic measure.

In practice, to compute the local curvature, several steps are necessary. First, at each vertex of the surface mesh, the normal (hence the gradient) is determined by a weighted average of unit normals to the adjacent elements. Then, in the local reference system (composed of the tangent plane and the normal) associated with each vertex, a quadric centered at this vertex and approaching at best the adjacent vertices is built. Afterwards, the Hessian is locally approximated by the Hessian to this quadric. Knowing the gradient and the Hessian of the solution at the nodes of $\mathcal{T}(\Omega)$, the curvatures and principal directions at each vertex of surface $\Sigma_u(\mathcal{T})$ are obtained.

IV. Numerical Example

To illustrate the proposed method, we consider an image of 700 × 536 pixels and the field of its grey levels. Figure 1 shows the original color image, a reproduction of The Adoration of the Magi (circa 1500). Its author was the North Italian Renaissance painter Andrea Mantegna, whose early career was shaped by impressions of Florentine works. The image is firstly represented by a regular grid of 699 × 535 quadrilaterals, defining its initial mesh. The analysis of the local geometric curvature of the Cartesian surface representing the field leads to the determination of an anisotropic geometric size map associated with the initial mesh, in order to bound the interpolation error (here $\varepsilon = 0.1$). Figure 2 (general view) and 3 (close-up) show the adapted anisotropic mesh. This mesh contains 227,557 vertices and 453,265 triangles. It has been realized using the anisotropic adaptive mesh generator BL2D [18]. The resulting interpolation error is 0.085 in average.

V. Conclusion

A novel approach connecting the problem of *a posteriori* error estimation and some techniques of surface meshing has been introduced. It constitutes an alternative method to classical approaches using the Hessian of the solution.

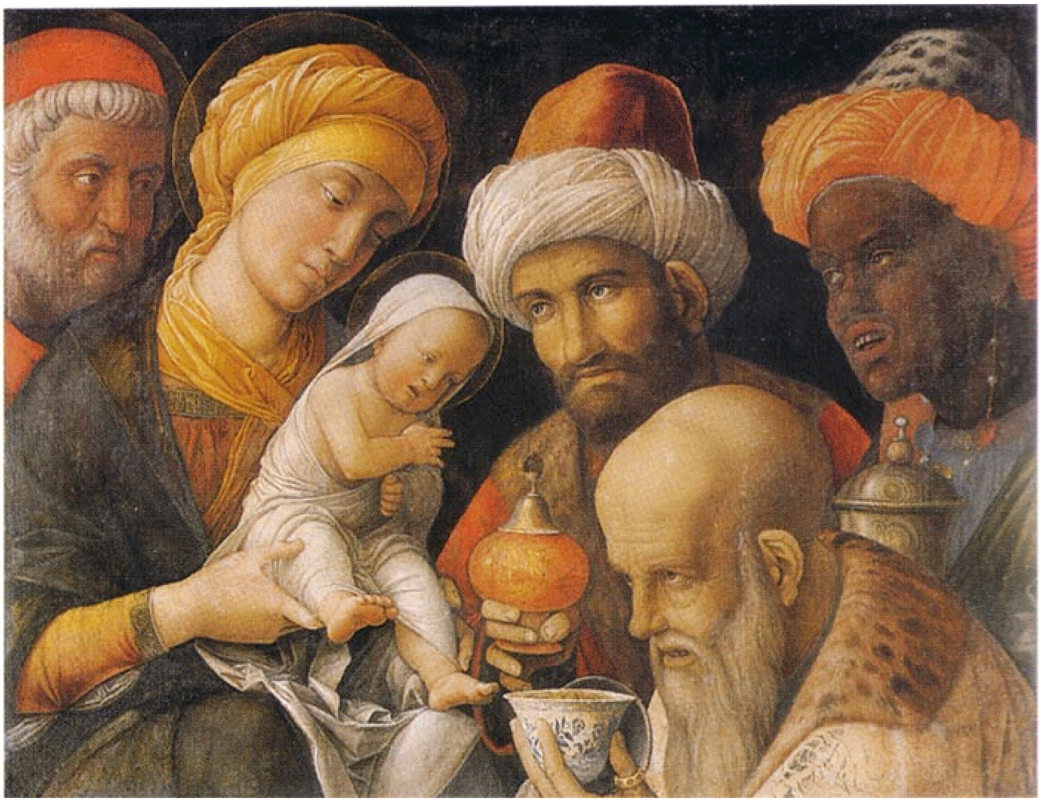


Figure 1. Original color image (painting by Andrea Mantegna, circa 1500).

To illustrate our methodology, a numerical example has been presented. The proposed *a posteriori* error estimation can be used in any computational problem where a static field must be calculated. In the case of dynamic fields, the adaptive computation is constituted by a calculation loop: at each iteration, beginning at the same global initial time and ending at a different time, a combination of the current metric and the previous metrics is applied.

REFERENCES

[1] M. Fortin, "Estimation *a posteriori* et adaptation de maillages", Rev. Europ. des Éléments Finis, vol. 9, n° 4, 2000.

[2] I. Babuska and K. Aziz, "On the angle condition in the finite element method", Siam J. Numer. Anal., vol. 13, n° 2, 1976, pp. 214–226.

[3] E.F. D'Azevedo and B. Simpson, "On optimal triangular meshes for minimizing the gradient error", Numer. Math., vol. 59, 1991, pp. 321–348.

[4] S. Rippa, "Long and thin triangles can be good for linear interpolation", Siam J. Numer. Anal., vol. 29, n° 1, 1992, pp. 257–270.

[5] M. Berzins, "Mesh Quality : a Function of Geometry, Error Estimates or Both ?", Eng. with Comp., vol. 15, 1999, pp. 236–247.

[6] X. Sheng and B.E. Hirsch, "Triangulation of trimmed surfaces in parametric space", Comp. Aid. Des., vol. 24, n° 8, 1992, pp. 437–444.

[7] M.V. Anglada, N.P. Garcia, and P.B. Crosa, "Directional adaptive surface triangulation", Comp. Aid. Des., vol. 16, 1999, pp. 107–126.

[8] H. Borouchaki, P. Laug, and P.L. George, "Parametric Surface Meshing Using a Combined Advancing-Front Generalized-Delaunay Approach", Int. Journal for Numerical Methods in Engineering, vol. 49, 2000, pp. 233–259.

[9] R. Verfürth, A Review of *A Posteriori* Error Estimation and Adaptive Mesh-Refinement Techniques, Wiley & Teubner, 1996.

[10] E.J. Nadler, "Piecewise linear best l_2 approximation on triangles, Approximation Theory V :" Proc. Fifth Inter. Symposium on Approx. Theory, Academic Press, New York, 1986, pp. 499–502.

[11] M. Berzins, "Solution-based Mesh Quality for Triangular and Tetrahedral Meshes", Proc. 6th International Meshing Roundtable, Sandia Lab., 1997, pp. 427–436.

[12] M. Zlamal, "On the finite element method", Numer. Math., vol. 12, pp. 394–409, 1968.

[13] M. Krizek, "On the maximum angle condition for linear tetrahedral elements", Siam J. Numer. Anal., vol. 29, n° 2, 1992, pp. 513–520.

[14] R.E. Bank and R.K. Smith, "Mesh smoothing using *a posteriori* error estimates", Siam J. Numer. Anal., vol. 34, n° 3, 1997, pp. 979–997.

[15] C. Manzi, F. Rapetti, and L. Formaggia, "Function approximation on triangular grids : some numerical results using adaptive techniques", Appl. Numer. Math., vol. 32, n° 4, 2000, pp. 389–399.

[16] P.J. Frey and H. Borouchaki, "Surface meshing using a geometric error estimate", Int. Journal for Numerical Methods in Engineering, vol. 58, 2003, pp. 227–245.

[17] P. Laug and H. Borouchaki, "Interpolating and Meshing 3-D Surface Grids", Int. Journal for Numerical Methods in Engineering, vol. 58, 2003, pp. 209–225.

[18] P. Laug and H. Borouchaki, The BL2D Mesh Generator – Beginner's Guide, User's and Programmer's Manual, INRIA Technical Report RT-0194, July 1996.

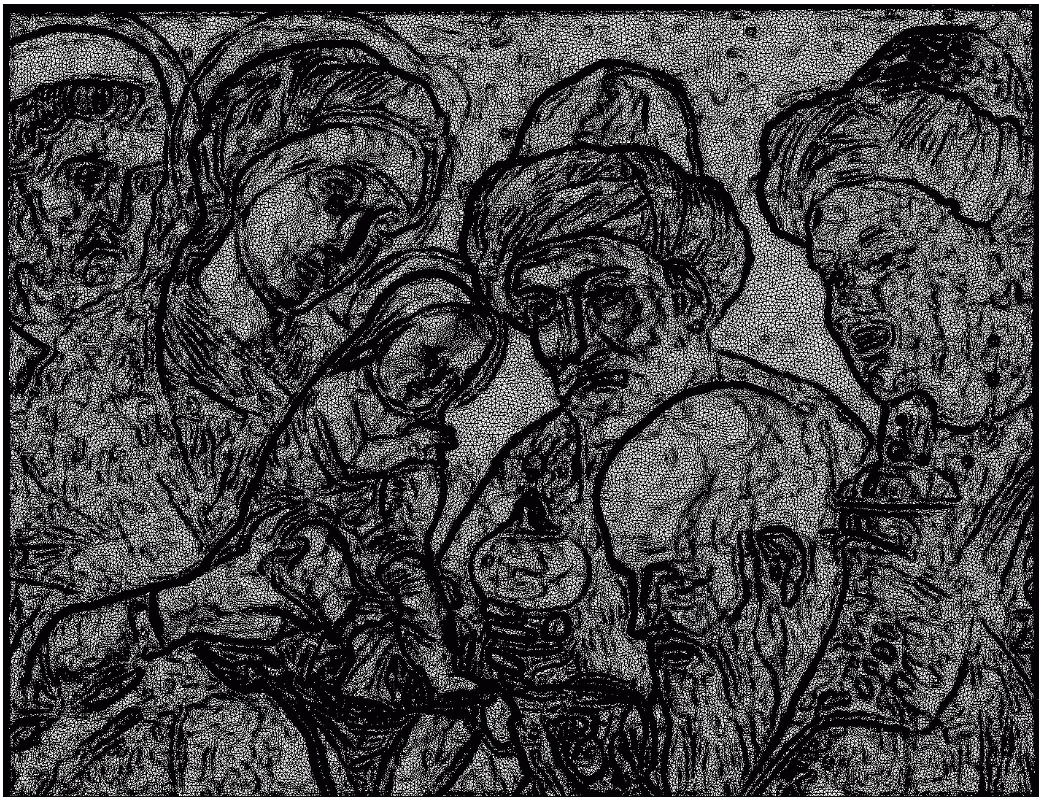


Figure 2. Adapted mesh corresponding to the interpolation error $\varepsilon = 0.1$.

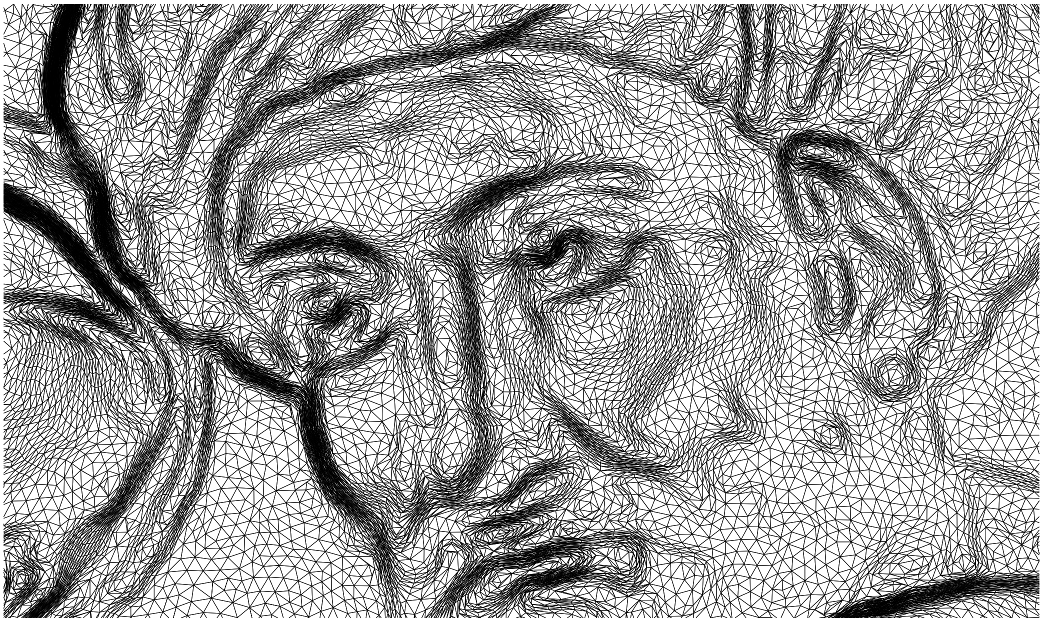


Figure 3. Enlargement of a selected region.

New Simulated Annealing Algorithm for Quadratic Assignment Problem

Kambiz Shojaee Ghandeshtani
Department of Electrical Engineering
University of Tehran
Tehran, Iran
E-mail: k.shojaee@ece.ut.ac.ir

Seyed Mohammad Hosein Seyedkashi
Dept of Mechanical Engineering
Tarbiat Modares University
Tehran, Iran
E-mail: seyedkashi@modares.ac.ir

Nima Mollai
Dept. of Electrical Engineering
Sadjad institute of higher education
Mashhad, Iran
E-mail: nimamollai@yahoo.com

Mohammad Mohsen Neshati
Department of Electrical Engineering,
Ferdowsi University of Mashhad
Mashhad, Iran
E-mail: mohsen_neshati@stu-mail.um.ac.ir

***Abstract*— In facility layout design, the problem of locating facilities with material flow between them was formulated as a Quadratic Assignment Problem (QAP), so that the total cost to move the required material between the facilities is minimized, where the cost is defined by a quadratic function. In this paper, a new definition in cooling scheduling is proposed for simulated annealing algorithm to solve the QAPs. Also a simple greedy-type algorithm is proposed to improve this method. The algorithm is implemented and tested on 40 benchmarks. In comparison with many other recently developed methods, considerable results are obtained by this approach.**

***Keywords*- QAP; Simulated annealing; Cooling Schedule; Greedy search.**

I. INTRODUCTION

Quadratic Assignment Problem or QAP is one of the most known and complicated problems in combinational optimization problems which was proposed by Koopmans and Beckmann in 1957 [1]. In 1976, Sahni and Gonzales [2] showed that QAP belongs to the class of NP-hard problems. QAP has been considered by many researchers for a long time and is capable to model many daily real problems. Among the applications of QAP, typewriter keyboard design [3], electronic components placement problems [4], campus planning [5], hospital layout [6], numerical analysis [7] and memory layout optimization in signal processors [8] can be mentioned.

In QAP, several facilities (for example n factories) are assigned to several locations (for example n cities) in such a way that the distance between any of the locations and also the flow between any of the facilities, are constant and predetermined. This assignment should be in a way that the goal function which is affected by the distance between the locations and the circulation number of the goods between the facilities, is minimized. In general, when the goal is to allocate n facilities to n locations, the number of possible situations is $n!$. That is why this problem is in the NP-complete problem category. It seems that the most achievable deterministic method to solve QAP is the branch-and-bound algorithm [5, 9, 10]. Recent researches illustrate that the accurate solving of QAP takes place with an n up to 36 [11] which in that case takes a long time [12]. So the researchers usually try to use metaheuristic methods to solve QAP. Some of these methods are Neural Networks algorithm [13], Simulated Annealing [14, 15], Threshold Accepting [16], Genetic Algorithms [17, 18], Tabu Search [19-21], Ant Colony Optimization [22, 23], Scatter Search [24].

In this paper, a new version of SA algorithm for QAP solving is presented with redefinition of Time Scheduling program parameters. In Section 2, QAP is fully explained. In Section 3, the base algorithm of simulated annealing is presented. The proposed simulation algorithm for QAP solving is discussed in the next section. In Section 5, the results of simulation is presented and compared with other metaheuristic methods.

II. QAP DESCRIPTION

In the mathematical definition of QAP, there exist n locations with specific coordinates. So an $n \times n$ matrix representing the distance between each pair of the locations will be generated $\left(D = \lfloor d_{ij} \rfloor_{n\times n}\right)$. The other $n \times n$ matrix generated includes the flow of each pair of facilities $\left(F = \lfloor f_{ij} \rfloor_{n\times n}\right)$. Considering the distance between locations and the flow between the facilities, the goal is to find the minimum cost for assigning the facilities. Mathematically, if $S(n)$ is assumed as a set of all possible permutations for a set of {0,1, …, n}, the goal is to find a permutation such as $p \in S(n)$ which is able to minimize a cost function defined as Equation 1.

$$Z(p) = \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij} f_{p(i)p(j)} \qquad (1)$$

ISBN: 978-1-61208-101-4

In fact, the *p* permutation shows the sequence of facilities placement in locations. More description is showed with an example in Figure 1.

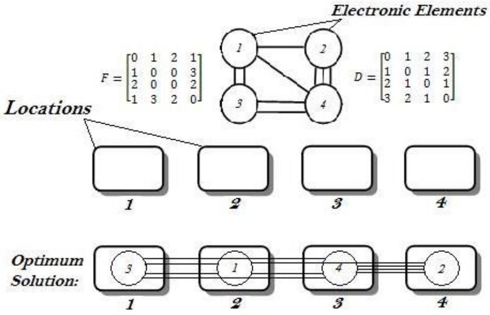

Electronic Elements
F = [0 1 2 1; 1 0 0 3; 2 0 0 2; 1 3 2 0]
D = [0 1 2 3; 1 0 1 2; 2 1 0 1; 3 2 1 0]
1
2
3
4
Locations
1
2
3
4
Optimum Solution:
3
1
4
2
1
2
3
4

Figure 1. A QAP sample for assignment of 4 facilities to 4 locations

In this figure, it is decided to place 4 electronic elements in 4 specific locations. Matrix D shows the distance between each pair of the locations and Matrix F shows the number of required connections between two parts. As shown in Figure 1, considering the length of consumed wires, the optimum answer for this problem is *p*=(3,1,4,2). In other words, the optimum result is obtained when the element #3 is placed in location #1, element #1 in location #2, element #4 in location #3 and element #2 in location #4.

III. Base Simulated Annealing Algorithm

The idea of simulated annealing algorithm was first proposed as the modified Monte Carlo method by Metropolis in 1953 who was working in the publishing industry [25]. He resembled paper to the material which is obtained after cooling of a molten material. SA for combinational optimization applications such as Traveling Salesman Problem was first developed by Kirkpatrick in 1983 inspired by Metropolis algorithm [26]. This algorithm is an adoption of cooling process in which metal is heated to its melting point and then slowly cools. This reduction in temperature is in such a way that the system will approximately be in thermodynamic equilibrium. During the gradual temperature reduction, the system becomes more ordered and approaches to the steady state with minimum energy. The main plan in the determination of temperature and the initial energy state of thermodynamic system is that if the energy changes are negative, the new structure (energy and temperature) will be accepted but if the changes are positive, the acceptance is dependent on Boltzmann distribution function:

$$P\{accept\} = \begin{cases} 1 & ,\Delta f \leq 0 \\ e^{-\Delta f / CT} & ,\Delta f > 0 \end{cases} \tag{2}$$

In which Δf is the change value of cost function, T is temperature parameter in simulated annealing process, P is the acceptance probability of the next point and C is a control parameter known as Boltzmann's constant with positive value [27]. The whole process will be repeated while the energy is minimized and the system reaches to the steady state. This algorithm is suitable for mixed discrete problems and complicated nonlinearity problems. In the SA algorithm, cooling schedule parameters control the process in search algorithm. Cooling schedule consists of three factors:

1) Initial temperature (T_0).

2) Convergence criterion or Freezing temperature (T_f).

3) Cooling function.

In this algorithm, when the initial and freezing temperatures are defined properly and the rate of temperature reduction is less than the slope of $T_k=T_0/(1+\log(k))$, then the SA algorithm will be converged to the absolute minimum when the number of tries (k) tends to infinity. But, According to the slope of this curve, temperature reduction makes the solving time very longer, therefore, faster temperature reduction functions are usually used such as $T_{k+1} = \alpha * T_k$ in which $0.8<\alpha<1$ or $T_k=T_{k-1}/(1+\log(k))$. Using these functions, the number of temperature steps from melting point to freezing point has a considerable reduction and hence the probability of passing through an effective temperature range for optimal search also decreases. So by the definition of numerous iterations in the inner search loop including new generation, assessment and decision making in each temperature, it is tried to give enough search time in optimum temperature range for the algorithm. In this algorithm the number of iterations in a specific temperature is called Markov chain length. SA pseudo-code algorithm is as following:

```
1) Randomly initialize the solution $V = V_{Start}$ .
2) Set the initial temperature $T = T_0$.
3) Until stop criterion is reached, do:
       Generate a new solution V' from the neighborhood of V.
       Let E and E' be the values of the cost function at S and S', respectively.
       If (E' < E), accept new solution V = V'.
       Else if (exp(-(E'-E)/T)) > a random number Є [0,1], accept new solution V = V'.
4) If freezing condition (convergence criterion) is valid, stop.
5) Reduce the temperature by cooling function.
6) Go to 3).
```

The way of producing new generation, based on the current generation, makes SA algorithm distinct in continuous or discrete problems. In continuous problems, some definitions such as neighborhood radius are used for producing a new generation in neighborhood of the current generation. In the SA process, the neighborhood radius is reduced according to the temperature in order to increase the convergence speed. This new generation in discrete problems is performed by some operators which implicitly generate the next generation in neighborhood of the current generation. These operators are also called Move Set. In QAP, each possible answer for the problem is corresponding to a permutation of 1 to *n*. several effective operators in discrete problems such as QAP are as follows:

1. Switching Or Swap Operator:

Randomly selects two locations from permutation and replaces with each other.

2. Translation Operator:

Randomly selects a portion of permutation and replaces in another random location in permutation.

3. k-Opt Operator:

In k-Opt move, the tour is broken into k parts, then the k parts reconnect in the other possible way. Inversion is the case of k-Opt in which $k = 2$.

In fact, these operators are used as local search approaches, in the global search approaches, such as, Tabu search, Simulated Annealing or Genetic algorithm.

IV. Proposed Algorithm

A. Proper Determination of Initial Temperature

Kirkpatrick [26] defined the initial temperature as essentially all proposed circuit flips are accepted but in quantitative definition of this qualitative significance has sufficed to presentation of a constant value (10) for his problem, so only with the justification that the initial temperature of an algorithm is sufficient to start, the probability of high initial temperature and hence non optimum operation of the algorithm is neglected. Percy [28] has assumed the initial temperature from 100000 to 4000000 according to the dimensions of the problem. Andrew [29] has changed the initial temperature from 0.001 to 100 and discussed the effect of this important parameter in temperature reduction.

In this paper, the initial temperature is defined in such a way that the proportion of, the accepted cases to the whole studied cases (γ), in Markov chain has the value of 0.2-0.5 according to different problems.

B. New Definition for Markov Chain Lenght

Since the Markov chain length is in fact giving enough time for search to the algorithm, it may be reconsidered according to the working temperature in order to optimum use of effective temperature. Constant definition of the number of iterations for search loop in a constant temperature condition (Markov chain length) is the definition of the same need of algorithm try in different temperatures for search. Kirkpatrick [26] has mentioned effective temperature range in search process of the algorithm which declares the effectiveness of search in this range. So the constant definition of inner search chain is not optimum. In this paper, the number of iterations for a specific temperature is proportional to the number of acceptances in an inner loop instead of the number of tries for generation and evaluation. The number of acceptances required for search in the inner loop of the algorithm decreases according to the temperature reduction.

C. Proper Definition of the Freezing Temperature

Definition of the freezing temperature or convergence condition is very important in increasing the speed and accuracy in the search process. If the stop condition of the algorithm is not defined effectively, the algorithm will be stopped sooner which as a result reduces the accuracy or the convergence of the algorithm will be announced by delay which results in the speed reduction.

In different papers the way of determining the convergence conditions for algorithm is explained in different methods and various criteria are discussed. For example Kirkpatrick [26] defined that:

“If the desired number of acceptances is not achieved at three successive temperatures, the system is considered "freeze " and annealing stops.”

Percy [28] has considered the approach to the optimum response or to the specific number of tries (500 iterations) as the stop condition of the algorithm.

In this paper, the stop condition is defined when in two sequential searches in Markov chain, there is no change in the best obtained result.

D. Improving the Results by a Simple Greedy Algorithm

In this paper, a kind of a greedy algorithm is used for local search at the end of the simulated annealing algorithm. This algorithm gives the final response of the algorithm. So considering [30], we can say that when Matrices of D and F are symmetric, if permutation of $p^{'}$ is created by replacement of the s_{th} and t_{th} elements in permutation of p, the cost function is calculated by Equation 4.

$$\Delta(p,s,t) = Z(p') - Z(p) = -2 \sum_{\substack{k=1 \\ k \neq s,t}}^{n} \left(f_{p(s)p(k)} - f_{p(t)p(k)}\right)(d_{sk} - d_{tk}) \qquad (3)$$

Based on this equation, a matrix is defined as $\Delta(p) = [\Delta_{st}]_{n \times n}$ in which Δ_{st} shows the difference in cost function because of displacement of the s_{th} and t_{th} elements in permutation of p. In this simple algorithm, the elements which produce the negative element in the matrix are moved until there will be no negative number in the matrix. For the determination of the negative elements' displacement priority, the lowest negative element is selected greedily.

V. Simulation and Comparison

The proposed algorithm is executed for a sample problem presented in QAPLIB site [11] and obtained results are compared with other algorithms. Considering [19], the standard problems discussed in QAPLIB can be classified in 4 categories.

I. Unstructured, randomly generated instances:

They are the problems in which distance and flow matrices are generated randomly with uniform distribution. These problems are usually more complicated than other QAPs. For example *taixxa* is in this category. (Each *x* is an integer)

II. Instances with grid-based distance matrix:

They are the problems in which distance matrix is created inspiring some points in Manhattan Island and flow matrix is randomly created. For example *Nugxx* and *Skoxx* are in this category.

III. Real-life instances:

These problems are derived from real applications of QAP. For instance, hospital layout is discussed in *Kra30x* category and typewriter keyboard design is discussed in *Bur26x* category. Their flow matrix has usually more zero in comparison with other categories and the input distribution of their flow matrix is not uniform.

IV. Real-life like instances:

Since the size of real-life instances are not so big, E. Taillard discussed Taixxb problems [19], which are like real-life instances with the same distribution to compensate this lack.

Considering the difference in QAP problems, in order to gain a better result, some parameters must be changed. The proposed algorithm starts with the initial temperature proportional to 20% of acceptances to the whole situations, (γ = 0.2) for categories (I) and (II), proportional to 50% of acceptances to the whole situations, (γ = 0.5) for category (III) and proportional to 40% of acceptances to the whole situations, (γ = 0.4) for category (IV).

This ratio has been obtained by the trial and error method in various problems. In the case that at the beginning of the algorithm, searching for the initial temperature performs with proportionate steps starting from the initial value of zero and this temperature increases until the ratio of the number of accepted cases to the total studied cases (γ) in a single markov chain length, reaches the determined ratio, we will reach a temperature equivalent to the melting point.

Finding the above ratio will be very important in the definition of the optimized initial temperature, in order to have a high speed in addition to maintaining the accuracy of the search process.

It has to be said that the performed operation in the existing loop in defining the initial temperature, is exactly the same operation used in the search engine.

This means that it starts with a random variable and after applying the switching operator, the acceptance terms of the algorithm will be checked. In the case of acceptance, the previous generation will be replaced by a new one and the operation will be continued until the markov chain ends and at the end, it is the ratio of accepted, to the total states that represents the desired ratio. If this ratio is sufficient, the loop will stop, but otherwise, a new ratio will be calculated for the increased temperature by the stepped increase of the temperature and repetition of the above stages. This temperature increase will be continued until the ratio of the accepted states to the total states in a single markov chain, reaches the ratio defined at the beginning of the algorithm.

The iterations in the first inner loop will finish when a specific number of acceptances occur related to A_0=2000n/(1-γ). In each temperature reduction the number of acceptances reduces with the equation $A_k = 0.8\sqrt[k]{5}A_0$. Therefore, more tries are performed in effective temperatures in the search process. The repetition rate reduction has been resulted by trial and error.

In the inner loop for producing the new generation, the Switching operator is used. It seems that this operator can find the best possible result.

Simulation and optimization of process is performed in Matlab7.7 by a Core2Due computer with a 2.66 GHZ CPU and 4GB of RAM. The effectiveness of the simulated annealing algorithm combined with the new cooling schedule mentioned in Section IV, has been studied by solving some of the complicated QAPs reported in the literature available in the QAPLIB. The criterion considered for evaluating the performance is The Average Percent Deviation (APD) of the solution quality from the Best-Known Solution (BKS) from the literature. APD is determined as follows:

APD = 100*(*C* – *BKS*) / *BKS*,

where, C and BKS are, calculated cost by the proposed algorithm and the best-known solution, respectively.

Table I provides a comparison between all the variants of QAP categories. The APD and the average time to completion obtained by the new approaches are compared with the other results given in recent novel researches. The first method chosen for comparison is the iterated fast local search algorithm by using order crossover with random sliding mutation named as IFLS / OXSM [31].The second one is a new iterated fast local search (NIFLS) algorithm by recombination of crossover with sliding mutation (RCSM) scheme that is referred as NIFLS / RCSM proposed in [32]. Finally, the results are compared with [12], for its new diversification TS variants for the QAP named as DivTS.

VI. Conclusion and Results

As the compared results in Table I shows, the new proposed definitions of cooling schedule in SA algorithm indicate the performance improvement in comparison with other algorithms. As it is obvious, the result of proposed method in APD criterion and average time of completion are 0.40 and 78% respectively, which are both better than the results in [31]. Also 0.50 and 87% improvement in APD and running time in comparison with [32] are obtained. Eventually by comparing the results with [12], it's shown that the average time to completion has been improved 63% but APD criterion has been weakened about 0.28.

Acknowledgment

This work is supported by Nano-Age Technology Group in Mashhad, Iran (www.nanoage.ir).

References

[1] T.C. Koopmans and M.J. Beckmann, "Assignment problems and the location of economic activities," Econometrica, vol. 25, pp. 53–76, 1957.

[2] S. Sahni and T. Gonzales, "P-complete approximation problems," Journal of the Association for Computing Machinery, vol. 23, pp. 555–565, 1976.

[3] M.A. Pollatschek, N. Gershoni, and Y.T. Radday, "Optimization of the typewriter keyboard by simulation," Angewandte Informatik, vol. 17, pp. 438-439, 1976.

[4] G. Miranda, H.P.L. Luna, G.R. Mateus, and R.P.M. Ferreira, "A performance guarantee heuristic for electronic components placement problems including thermal effects," Computers and Operations Research, vol. 32, pp. 2937–2957, 2005.

TABLE I: PERFORMANCE OF THE PROPOSED SA FOR THE SELECTED QAP INSTANCES FROM QAPLIB IN COMPARISON WITH BEST KNOWN SOLUTIONS AND OTHER METHODS.

Instances					Proposed SA		IFLS / OXSM		NIFLS / RCSM		DivTS	
No.	Category	*problem Name*	*N*	BKS	*APD*	*CPU Time (Sec)*	*APD*	*CPU Time (Sec)*	*APD*	*CPU Time (Sec)*	*APD*	*CPU Time (Sec)*
1	Unstructured, randomly generated	Lipa30a	30	13178	0.000	12.40	0.000	119.72	0.000	81.57	0.000	72
2		Lipa40a	40	31538	0.000	51.23	0.000	489.91	1.060	346.37	0.000	189.6
3		Lipa50a	50	62093	0.083	188.02	1.020	1556.28	0.740	1061.26	0.000	394.2
4		Rou15	15	354210	0.000	0.74	0.000	2.95	0.000	2.89	0.000	54
5		Rou20	20	725522	0.115	19.38	0.020	11.73	0.000	11.40	0.000	14.4
6		Tai30a	30	1818146	0.680	72.64	1.110	83.06	1.510	82.81	0.000	78.6
7		Tai40a	40	3139370	1.349	233.20	1.850	354.38	1.870	346.93	0.222	309.6
8		Tai50a	50	4941410	1.803	462.58	2.250	1104.03	2.130	1076.12	0.725	613.8
9		Tai60a	60	7208572	1.930	948.61	2.750	2739.83	1.610	2701.20	0.718	1541.4
10		Tai80a	80	13557864	1.487	1242.47	2.340	11332.95	2.160	11584.81	0.753	3164.4
11	Grid-based distance matrix	Nug20	20	2570	0.000	6.21	0.000	16.06	0.000	11.16	0.000	13.8
12		Nug24	24	3488	0.000	8.81	0.000	39.75	0.000	27.29	0.000	24
13		Nug27	27	5234	0.000	16.94	0.000	80.56	0.000	53.49	0.000	34.8
14		Scr12	12	31410	0.000	0.12	0.000	1.11	0.000	11.09	0.000	24
15		Scr15	15	51140	0.000	0.35	0.000	3.09	0.000	3.11	0.000	54
16		Scr20	20	110030	0.001	7.90	0.000	12.69	0.000	12.34	0.000	13.8
17		Sko56	56	34458	0.144	522.73	0.470	2612.69	0.270	1828.30	0.002	789.6
18		Sko72	72	66256	0.188	1304.26	0.730	8663.36	0.540	6169.37	0.006	2278.8
19		Sko81	81	90998	0.070	2004.03	0.430	16959.59	0.510	11729.39	0.016	3381.6
20		Sko100a	100	152002	0.099	4211.41	1.300	308.66	0.320	33616.39	0.027	7753.2
21		Tho30	30	149936	0.073	51.82	0.290	118.94	0.350	78.12	0.000	72
22		Wil50	50	48816	0.068	336.72	0.280	1498.69	0.240	1039.73	0.000	475.2
23	Real-life	Bur26h	26	7098658	0.008	23.37	0.000	57.47	0.000	37.44	0.000	31.2
24		Chr12c	12	11156	0.000	2.38	0.000	1.02	0.000	1.01	0.000	24
25		Chr15a	15	9896	0.827	6.79	0.000	2.97	1.150	2.95	0.000	54
26		Esc16j	16	8	0.000	0.05	0.000	2.91	0.000	2.03	0.000	6.6
27		Esc32h	32	438	0.000	24.19	0.000	85.75	0.000	54.90	0.000	82.2
28		Esc64a	64	116	0.000	1.09	0.000	1521.70	0.000	1059.92	0.000	1041.6
29		Esc128	128	64	0.000	71.80	-	-	0.000	23370.06	0.000	9781.2
30		Had12	12	1652	0.000	1.01	0.000	0.97	0.000	0.96	0.000	24
31		Had14	14	2724	0.000	2.21	0.000	1.97	0.730	2.05	0.000	42
32		Had20	20	6922	0.062	11.12	0.000	10.58	0.000	10.18	0.000	13.8
33		Kra30b	30	91420	0.172	68.94	0.130	101.83	0.190	68.67	0.000	72.6
34		Ste36a	36	9526	0.277	48.30	0.000	204.36	3.000	202.17	0.000	138
35	Real-life like	Tai12b	12	39464925	0.000	1.69	0.000	1.03	0.000	1.03	-	-
36		Tai15b	15	51765268	0.004	3.35	0.000	3.05	0.000	3.07	-	-
37		Tai25b	25	344355646	0.552	19.42	5.590	34.52	5.590	34.26	0.000	27.6
38		Tai30b	30	637117113	1.374	26.74	2.220	80.55	1.400	83.15	0.000	78.6
39		Tai35b	35	283315445	1.103	53.34	3.540	186.42	5.080	190.14	0.000	143.4
40		Tai80b	80	818415043	0.717	1147.28	2.790	10532.89	2.610	10942.90	0.006	3494.4
Average					0.330	330.39	0.746	1562.56	0.827	2698.55	0.065	958.2
Proposed SA Average					-	-	0.338	337.02	0.330	330.39	0.347	347.4

[5] J.W. Dickey and J.W. Hopkins, “Campus building arrangement using Topaz,” Transportation Research, vol. 6, pp. 59–68, 1972.

[6] A.N. Elshafei, “Hospital layout as a quadratic assignment problem,” Operations Research Quarterly, vol. 28, pp. 167–179, 1977.

[7] M.J. Brusco and S. Stahl, “Using quadratic assignment methods to generate initial permutations for least squares unidimensional scaling of symmetric proximity matrices,” Journal of Classification, vol. 17, pp. 197–223, 2000.

[8] B. Wess and T. Zeitlhofer, “On the phase coupling problem between data memory layout generation and address pointer assignment,” Lecture Notes in Computer Science, vol. 3199, pp. 152–166, 2004.

[9] E. L. Lawler, “The quadratic assignment problem,” Manage. Sci., vol. 9, pp. 586–599, Jul. 1963.

[10] A. A. Assad and W. Xu, “On lower bounds for a class of quadratic 0, 1 programs,” Oper. Res. Lett., vol. 4, pp. 175–180, Dec. 1985.

[11] R. E. Bedkard, S. E. Karisch, and F. Rendl, “QAPLIB-A Quadratic assignment problem library”, http://www.opt.math.tu-graz.ac.at/qaplib

[12] Tabitha James, César Rego, and Fred Glover, “Multistart tabu search and diversification strategies for the quadratic assignment problem,” IEEE Transactions on systems, man, and cybernetics-Part A: systems and humans, vol. 39, pp. 579-596, May 2009.

[13] C. Bousono-Calzon and M. R.W. Manning, “The hopfield neural network applied to the quadratic assignment problem,” Neural Comput. Appl., vol. 3, pp. 64–72, Jun. 1995.

[14] D. T. Connolly, “An improved annealing scheme for the QAP,” Eur. J. Oper. Res., vol. 46, pp. 93–100, May 1990.

ISBN: 978-1-61208-101-4

[15] M. R. Wilhelm and T. L. Ward, “Solving quadratic assignment problems by simulated annealing,” IIE Trans., vol. 19, pp. 107–119, 1987.

[16] V. Nissen, “Solving the quadratic assignment problem with clues from nature,” IEEE Trans. Neural Netw., vol. 5, pp. 66–72, Jan. 1994.

[17] J. P. Cohoon and W. D. Paris, “Genetic placement,” IEEE Trans. Comput.- Aided Design Integr. Circuits Syst., vol. 6, pp. 956–964, Nov. 1987.

[18] D. M. Tate and A. E. Smith, “A genetic approach to the quadratic assignment problem,” Comput. Oper. Res., vol. 22, pp. 73–83, Jan. 1995.

[19] E. Taillard, “Robust taboo search for the quadratic assignment problem,” Parallel Comput., vol. 17, pp. 443–455, 1991.

[20] J. Skorin-Kapov, “Tabu search applied to the quadratic assignment problem,” ORSA J. Comput., vol. 2, pp. 33–45, 1990.

[21] A. Misevicius, “A tabu search algorithm for the quadratic assignment problem”, Comput. Optim. Appl., vol. 30, pp. 95–111, Jan. 2005.

[22] V. Maniezzo and A. Colorni, “The ant system applied to the quadratic assignment problem,” IEEE Trans. Knowl. Data Eng., vol. 11, pp. 769–778, Sep/Oct. 1999.

[23] L. M. Gambardella, E. D. Taillard, and M. Dorigo, “Ant colonies for the quadratic assignment problem,” J. Oper. Res. Soc., vol. 50, pp. 167–176, Feb. 1999.

[24] V. D. Cung, T. Mautor, P. Michelon, and A. Tavares, “Scatter search for the quadratic assignment problem,” in Proc. IEEE Int. Conf. Evol. Comput, pp. 165–169, 1996.

[25] N. Metropolics, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of State Calculations by Fast Computing Machines,” J. Chem. Phy, vol. 21, pp. 1087-1092, 1953.

[26] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, “Optimization by Simulated Annealing,” SCIENCE, vol. 220, pp. 671-680, May 1983.

[27] V. Cerny, “Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm,” J. Opt. Theory Appl, vol. 45, pp. 41-51, 1985.

[28] Percy P. C. Yip and Yoh-Han Pao, “Combinatorial a guided evolutionary simulated annealing approach to the quadratic assignment problem", IEEE transactions on systems, man, and cybernetics, vol. 24, September 1994.

[29] S. Andrew, “Parallel N-ary speculative computation of simulated annealing", IEEE transactions on parallel and distributed systems, vol. 6, pp. 997-1005, October 1995.

[30] R. E. Burkard and F. Rendl, "A thermodynamically motivated simulation procedure for combinatorial optimization problems," European journal operational research, vol. 17, pp. 169-174, 1984.

[31] A.S. Ramkumar, S.G. Ponnambalam, N. Jawahar, and R.K. Suresh “Iterated fast local search algorithm for solving quadratic assignment problems,” Robotics and Computer-Integrated Manufacturing, vol. 24 pp. 392–401, 2008.

[32] A.S. Ramkumar, S.G. Ponnambalam, and N. Jawahar “A new iterated fast local search heuristic for solving QAP formulation in facility layout design,” Robotics and Computer-Integrated Manufacturing, vol. 25 pp. 620– 629, 2009.

A Grid-based Approach to Continuous Clustering of Moving Objects

Tongyu Zhu, Yuan Zhang, Weifeng Lv, Fei Wang
State Key Laboratory of Software Development Environment
Beihang University, Beijing, China
{zhutongyu, yuanzhang, lwf, wangfei }@nlsde.buaa.edu.cn

***Abstract*—With the rapid advances in wireless devices and positioning technologies, tracking and clustering of moving objects has drawn increasing attention. Previous methods of clustering moving objects merge clusters by searching all the existing clusters, which have an obvious decline in efficiency as the number of clusters increases. This paper proposes a grid-based approach to continuous clustering of moving objects. We first employ dynamic grid to narrow the searching area when merging clusters, and then develop an efficient split algorithm to handle the split of clusters, which avoids multiple splits of one cluster during a period of time. At last, a comprehensive experimental evaluation has been conducted to validate our approach, and the results indicate the efficiency and effectiveness of our algorithm, especially for large data set.**

Keywords- Moving object; clustering; grid; data mining

I. INTRODUCTION

Due to the growing popularity of wireless devices (e.g., PDAs, mobile phones, navigation devices) and the rapid advances in wireless communication and positioning technologies (e.g., GPS), tracking the behaviors and movements of individuals becomes increasingly available, which boosts various kinds of services exploiting knowledge of object movement.

Clustering analysis aims to group similar data into the same group and different data into distinct groups, which provides a summary of data distribution patterns and discovers data correlations in dataset. Early clustering techniques mainly focused on analyzing static datasets [1], [2], [3]. Recently, clustering moving objects has attracted increasing attention [4], [5], [6], [7] which has various kinds of applications, including mobile computing, targeted sales, traffic jam prediction, weather forecast and animal migration analysis.

As the positions of moving objects continuously change, treating moving objects as static ones and periodically clustering them with the methods for static datasets is a brute-force approach which does not consider the information of the movement. Some incremental clustering schemes have been proposed [4], [6], [7], and they mainly focus on dynamically maintaining a small set of *moving micro-clusters* (MMCs) [4]. The concept of MMC is a group of objects that are not only close to each other at current time, but also likely to move together for a while.

The *split* and *merge* operations are central parts of the schemes for incremental clustering of moving objects. As has been observed in the literature, there are two kinds of methods to deal with the split of a MMC. One is to delete the *extreme* object [4] or the farthest object from the center of MMC [7]. The other is to divide the MMC into two MMCs [6]. Neither of them considers the situation that a MMC may continuously split during a period of time, and when this situation occurs, it will take a long time to handle the multiple splits of a MMC. Moreover, when checking whether a MMC can be merged with other MMCs, previous schemes search all the existing MMCs, which have an obvious decline in efficiency as the number of MMCs increasing.

In this paper, we present a grid-based approach to continuous clustering of moving objects. First, we develop an efficient split algorithm to handle the split of a MMC, which avoids multiple splits of one MMC during a period of time. Then we employ hierarchical grids similar to that used in [8] in order to narrow the searching area during *merge* operation. The spatial area is divided into square grids, and each grid will be dynamically divided or combined according to the number of MMCs belonging to it.

Our contributions can be summarized as follows: we develop an efficient split scheme which can avoid multiple splits of one MMC during a period of time. We employ hierarchical grids to narrow the searching area during *merge* operation. We present the algorithms of maintaining the dynamic grids and applying the dynamic grids to MMCs.

II. RELATED WORK

As one of the most important analysis techniques, clustering has been an active area of research in the field of data mining. A lot of clustering techniques have been proposed for static data sets [1], [2], [3], [8], [9], [10], [11]. They can be classified into the partitioning, hierarchical, density-based, grid-based and model-based method. The k-means algorithm [1] is representative of partitioning method, which aims at dividing the objects into *k* clusters in order to minimize the metric relative to the centroids of the clusters. The Birch algorithm [2] is a comprehensive hierarchical method, which originally proposed the concept of *micro-clustering* and the notion of *clustering feature* (CF) and *CF tree*. The STING algorithm [8] is a grid-based method, which divides the spatial area into rectancle cells and employs a hierarchical stucture. It has high efficency especially for large data set. To deal with the moving objects, our approach extends the grid in STING to a dynamic one,

ISBN: 978-1-61208-101-4

which will be dynamically divided or combined according to the number of MMCs belonging to it.

Recently, clustering of moving objects has drawn increasing attention. Har-Peled [12] aims to show that moving objects can be clustered once and the resulting clusters are competitive at any future time during the motion. Li et al. [4] first addressed clustering of moving objects by applying *micro-clustering* and dynamically maintaining bounding boxes of clusters. However, the bounding boxes of the cluster are likely to be exceeded frequently, which makes the number of maintenance events dominate the overall runtimes of the algorithms.

Zhang and Lin [5] proposed a histogram technique based on the clustering paradigm. A histogram must be reconstructed if too many updates occur, and there are usually a large number of updates at each timestamp, which makes histogram maintenance lack of efficiency.

Kalnis et al. [13] presented three algorithms to discover moving clusters from historical trajectories of moving objects. They treat a moving cluster as a sequence of spatial clusters that appear in consecutive snapshots of the object movements. Such moving clusters can be identified by comparing clusters at consecutive snapshots, the cost of which can be very high.

Jensen et al. [6] proposed a scheme capable of incrementally clustering moving objects in two-dimensional Euclidean space, which extended the concepts of CF and CF tree in Birch and employed a notion of object dissimilarity considering object movement across a period of time. Their experiments show this scheme performs significantly faster than traditional methods which frequently rebuild clusters.

Lai and Heuer [7] developed an approach dynamically maintaining a small set of MMCs, and they obtain global clusters by clustering these representative MMCs with traditional clustering algorithms for static data sets. Rosswog and Ghose [14] consider the situation that the moving objects intersect the space occupied by objects from another cluster and extend the distance measure to a function of the position history of the objects so as to improve the accuracy of traditional data clustering algorithms on spatio-temporal data sets.

III. Preliminaries

In this section, we first introduce the model of MMC and some concepts about it, and then we describe the structure of the dynamic grid used in our approach and define some notions associated with it.

A. Model of MMC

In this paper, we consider moving objects in two-dimensional (2D) Euclidean space, which can be easily extended to higher dimensions. We define the *minimum update interval* (*minUI*), which denotes that the velocity of all the moving objects can be treated as constant during this interval. We assume that each moving object can transmit its new position and velocity to the server at the beginning of each *minUI*.

Each moving object can be represented as (*oid*, *p*, *v*, *t*), where *oid* is the unique ID of this object, *p* is the position of this object at time *t* and *v* is the velocity at time *t*. Both *p* and *v* are 2D vectors. During each *minUI*, the position of a object is a linear function of time and at time *t1*, it can be computed as *p(t1) = p + v(t1 - t)*, where *t1>t.*

Definition 1. The center of a MMC including *N* objects is of the form (*P*, *V*), where $P=(\sum_{i=1}^{N}p_i)/N$ and $V=(\sum_{i=1}^{N}v_i)/N$. *P* and *V* are position and velocity of the center respectively.

A MMC can be represented as (*cid*, *objn*, *objid*, *center*, *cf*, *t*), where *cid* is the ID of this MMC, *objn* is the number of moving objects in this MMC, *objid* is a list which contains the ID of objects in this MMC, *cf* is the clustering feature (CF) of this MMC and *center* is the center at time *t*. The CF of a MMC is defined as follows.

Definition 2. The CF of a MMC including *N* objects is of the form (*SP*, *SP*^{2}, *SV*, *SV*^{2}, *SPV*), where $SP=\sum_{i=1}^{N}p_i$, $SP^2=\sum_{i=1}^{N}p_i^2$, $SV=\sum_{i=1}^{N}v_i$, $SV^2=\sum_{i=1}^{N}v_i^2$ and $SPV=\sum_{i=1}^{N}p_iv_i$.

Claim 1. The CF at time t_{now} ($t_{now}>t$) can be maintained incrementally as follows [6]:
$CF'=(SP+SV(t_{now}-t), SP^2+2SPV(t_{now}-t)+SV^2(t_{now}-t)^2, SV, SV^2, SPV+SV^2(t_{now}-t))$.

Claim 2. When an object (*oid*, *p*, *v*, *t*) is inserted or deleted from the MMC, its CF can be modified as
$CF'=(SP\pm p, SP^2\pm p^2, SV\pm v, SV^2\pm v^2, SPV\pm pv)$.

Definition 3. The average radius *R*(*t*) of a MMC is the average Euclidean distance (ED) between its member objects and its center. It can be computed as

$$R(t)=\sqrt{\frac{1}{objn}\sum_{i=1}^{objn}D^2(p_i(t),p_c(t))}=\sqrt{\frac{1}{objn}(SP^2-\frac{(SP)^2}{objn})} \tag{1}$$

where *D* (p_i (*t*), p_c (*t*)) is the ED between object *i* and the center.

According to [6], the average radius of a MMC at time *t1* can be updated based on the CF given at time *t* (*t1*>*t*) as

$$R(t1)=\sqrt{(a\Delta t^2+b\Delta t+c)/objn}, \tag{2}$$

Where $a=SV^2-(SV)^2/objn$, $b=2(SPV-SPSV/objn)$, $c=SP^2-(SP)^2/objn$ and $\Delta t=t1-t$.

B. Structure of Dynamic Grid

We employ the dynamic grid (DG) similar to that used in [8]. The spatial area is divided into square grids and the grids have a hierarchical structure. The first level of DG is the root and denotes the whole area, and the grids at the bottom level of DG are leaves. Each grid has four children at its lower level and each child corresponds to one quadrant of the parent grid. The children are numbered from 0 to 3. Fig. 1 illustrates the first three levels of a DG and Fig. 1(b) indicates the numbered children of a grid. Actually, the structure of DG is a quadtree, in which each tree node corresponds to a grid in DG and each tree level corresponding to a level in DG. In the following context, we no longer distinguish between grid and quadtree node.

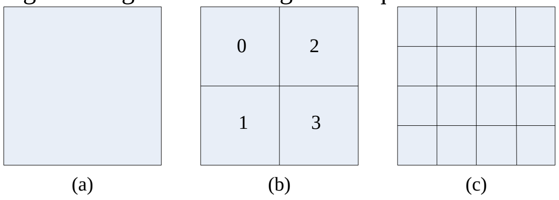


Figure 1. The first three levels of a DG

(a) the first level; (b) the second level; (c) the third level

we can use a 6-tuple (*Rcoord*, *height*, *cnum*, *cidlist*, *children*, *neighblist*) to represent a grid, where *Rcoord* is the relative coordinate of the grid, *height* is the height of the grid, *cnum* is the number of MMCs in this grid, *cidlist* is the list which contains the ID of MMCs in this grid, *children* is a list containing the pointers of the grid's children and *neighblist* is a list consisting of the pointers of the grid's neighbors.

Definition 4. The relative coordinate (RC) of a grid in DG can be represented by the form (*x*, *y*, *l*), where (*x*, *y*) is the RC of the upper left corner of the grid, and *l* is the length of this grid which is not larger than the threshold D_m (D_m will be defined in later section). The RC of the root is (0, 0, *L*), where *L* is the length of the whole area. If the RC of a grid is (*x*, *y*, *l*), then the RCs of its four children can be computed as (2*x*, 2*y*, *l*/2), (2*x*, 2*y*+1, *l*/2), (2*x*+1, 2*y*, *l*/2) and (2*x*+1, 2*y*+1, *l*/2) in order of children numbers.

Definition 5. The neighbors of a grid in DG mean grids adjacent to it. As shown in Fig. 2, a grid *G* has at most 8 neighbors which are numbered from 0 to 7 (e.g. Fig. 2(a)). The root of DG has no neighbor and other grids have at least 3 neighbors (e.g. Fig. 2(b)).

By employing RC, given the RC of a grid, we can easily compute the RCs of its neighbors. For example, in Fig. 2(a), if the RC of *G* is (*x*, *y*, *l*), then the RC of its neighbor 3 is (*x*-1, *y*, *l*), and the RC of its neighbor 7 is (*x*+1, *y*+1, *l*).

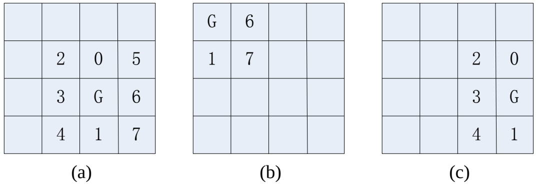


Figure 2. Neighbours of grid G

Definition 6. Each grid in DG has two possible states: expanded and unexpanded. If the subtree of a grid contains no MMCs, this grid is unexpanded. Otherwise, it is expanded.

Definition 7. Each level of DG has two possible states: expanded and unexpanded. As long as a grid at this level is expanded, this level is expanded. Otherwise, it is unexpanded.

Definition 8. The height of a grid is equal to the number of expanded levels in its subtree except for itself. Thus the height of an unexpanded grid is 0 and the height of an expanded grid is equal to the maximum height of its children plus 1.

IV. MAINTENANCE OF DYNAMIC GRID

The DG in our approach is dynamically maintained since MMCs may continue leaving a grid and then entering another grid. This section introduces some important algorithm about DG, including initialization, insertion, deletion, division and combination.

A. *Initialization*

We construct the initial DG with *h* levels (an initial quadtree with *h* levels), where *h* can be estimated from the capability of each grid *c* and the total number of MMCs *n*, that is, *h* satisfies $4^{h-1}c \geqslant n$. For each grid in DG, we initialize its *cnum* and *height* with 0, set its *cidlist* empty, compute its *Rcoord* as Definition 4 and initialize its *neighblist* according to Definition 5.

We employ a hash table *gridhash* which maps each MMC ID to the pointer of the grid it belongs to, so that given the ID of a MMC, we can fast locate the grid it belongs to. Moreover, we store the pointer of each grid into a table, in which each grid pointer can be quickly accessed by the level it belongs to and the RC of the grid.

B. *Insertion and Division*

We decide which grid a MMC belongs to by checking which "minimum" expanded grid its center is in. The minimum expanded grids denote the expanded grids with the minimum length and the minimum length *minl* can be computed as $minl = maxl / 2^{maxh}$, where *maxl* and *maxh* are the length and the height of the root in DG respectively.

After finding the belonging grid, the MMC will be inserted into this grid. To insert a MMC with ID *cid* into grid *G*, we modify *cnum* of *G* and all the ancestors of *G*, add *cid* to *cidlist* of *G*, modify the hash table *gridhash* and check whether *G* needs to be divided.

```
Divide(G)
Input: G (Rcoord, height, cnum, cidlist, children, neighblist)
       is a grid to be divided
   G.height++
   modify the height of all the ancestors of G if neccesary
   for each MMC with the ID cid in G
       decide cid belongs to G.children[ch] via cid.p
       add cid into G.children[ch].cidlist
       G.children[ch].cnum++
       modify the hash table gridhash
   clear G.clidlist
   for each child ch of G
       if G.children[ch].cnum exceeds the grid capacity
           Divide(G.children[ch])
end Divide.
```

Figure 3. Division algorithm for DG

In order to limit the number of MMCs in one grid, we set the grid capability *c*. If the number of MMCs in a grid exceeds *c*, this grid will be "divided", which means the MMCs in this grid will be redistributed to its children (at its lower level). The division algorithm is shown in Fig. 3. To divide a *G*, we first modify the height of *G* and all the ancestors of *G* according to Definition 8, and then for each MMC in *G*, we decide which child of *G* it belongs to and add its ID to the *cidlist* of this child. Meanwhile, we modify *cnum* of this child and *gridhash*. When the redistribution is over, we check whether the children of G should be divided.

C. Deletion and Combination

When a MMC leaves a grid G, we should delete it from G. In order to delete the MMC *cid* from grid G, we modify *cnum* of G and all the ancestors of G, delete *cid* from *G.cidlist*. If G does not contain any MMC now, we proceed to check the parent of G p. If p contains no MMC, we combine the children of p.

It's not complex to combine the children of p. We just need set the height of p to 0 and modify the height of all the ancestors of p. Then, if the parent of p (if existed) contains no MMC now, the combination will repeat and probably so on up to the root.

V. CLUSTERING BASED ON DYNAMIC GRID

This section will present our clustering scheme which is based on DG. We first introduce the distance metric and data structures used in our approach, and then we describe the details of the main algorithms.

A. Distance Metric

We utilize the distance metric with weight value proposed in [6] and as the *minUI* is short, we simplify this distance metric that we just consider three timestamps and our distance metric can be defined as

$$DM(O_1, O_2) = \sum_{i=1}^{3} w_i D^2(p_1(t_i), p_2(t_i)) \qquad (3)$$

where t_1 is the current time t, $t_2 = t + minUI / 2$, $t_3 = t + minUI$, D ($p_1(t_i)$, $p_2(t_i)$) is the ED between objects O_1 and O_2 at time t_i, and w_i ($0 < w_i < 1$) is the weight value at time t_i which satisfies $w_1 \geqslant w_2 \geqslant w_3$ and $w_1 + w_2 + w_3 = 1$.

Accordingly, the distance metric applying to an object O and a MMC C is

$$DM(O, C) = \sum_{i=1}^{3} w_i D^2(p_O(t_i), p_C(t_i)) \qquad (4)$$

where p_O is the position of O and p_C is the center position of C. Also, the distance metric can be applied to two MMCs C_1 and C_2 as follows

$$DM(C_1, C_2) = \sum_{i=1}^{3} w_i D^2(p_1(t_i), p_2(t_i)) \qquad (5)$$

where p_1 and p_2 are the center positions of C_1 and C_2 respectively.

B. Data Structures and Initialization

Two data structures are needed: the event queue Q and the hash table *MMChash*. Q stores future split events <t, *cid*> in ascending order of t, where t is the split time and *cid* is the ID of the MMC. *MMChash* maps each object ID to the MMC it belongs to, so that given the ID of an object, we can fast locate the MMC it belongs to.

During the initialization, we set the MMC capability C, which represents the maximum number of objects a MMC can contain, and we define the threshold R_m to represent the maximum average radius of a MMC. Also, we set the threshold D_m to denote that if the distance between an object and a MMC according to (4) exceeds D_m, this object cannot be inserted into the MMC. Then we construct the initial MMCs by the algorithm of object insertion introduced later.

C. Object Insertion and Deletion

Since the length of each grid is not larger than D_m, to insert an object O, we just need search MMCs in G and the neighbors of G instead of searching all the preexisting MMCs. As shown in Fig. 4, we first find the grid G O belongs to, and then search MMCs that are not full in G and the neighbors of G to find the nearest MMC to O according to (4). If the distance between the nearest MMC and O is larger than D_m, we create a new MMC for O. Otherwise, we try to insert O into this MMC. We compute the virtual CF of this MMC after absorbing O according to Claim 2 and then compute the virtual radius. If the virtual radius is larger than R_m, which indicates the insertion will lead this MMC into split, the insertion is failed and we create a new MMC for O. Otherwise, we update this MMC, modify *MMChash*, check whether this MMC changes its belonging grid, update the split event about this MMC in Q and the insertion is successful.

```
InsertObj(O)
Input: object O
find the grid G object O belongs to
search MMCs which are not full in G and G.neighblist
if the distance between O and the MMC cid nearest to it
is larger than D_m
   create a new MMC for O
else
  compute the virtual CF and virtual radius vr of cid
  absorbing O
  if vr ≥ R_m
     create a new MMC for O
  else
     cid.objn++
     update cid.cf and cid.center
     add O into cid.objid
     modify the hash table MMChash
     if cid changes the grid it belongs to
        insert cid into the new grid
        delete cid from the old grid
     update the split event about cid
end InsertObj.
```

Figure 4. The algorithm of object insertion

To delete an object O from a MMC is a reverse process of the object insertion. We just need remove O from *objid* of the MMC and update *objn*, *center*, *cf* of this MMC.

D. Split

The split of a MMC occurs when its average radius exceeds R_m. According to (2), the split time is when $R(t) = R_m$. For simplicity, we consider $R^2(t) = R_m^2$, that is, $(a\Delta t^2 + b\Delta t + c) / objn = R_m^2$. It's a quadratic equation about Δt and the solution is

$$\Delta t = \begin{cases} (-b + \sqrt{b^2 - 4a(c - R_m^2 objn)}) / (2a), & a \neq 0 \\ (R_m^2 objn - c) / b, & a = 0 \end{cases} \qquad (6)$$

According to [6], from current time *t* to *t* + *minUI*, the split time of a MMC during *minUI* can be determined in the process: if $R^2(t) > R_m^2$, the split time is the current time. Otherwise, if $R^2(t + minUI) \le R_m^2$, there is no need to split during *minUI*, and if not, the split time can be computed according to (6).

As a MMC often splits many times during *minUI*, after its split, we need know whether it will split again during *minUI*, so we compare its average radius at current time and at the end *of minUI* with R_m. This algorithm is shown in Fig. 5.

```
willSplit(cid, endtime)
Input: MMC cid and  the end time of the interval endtime
Output: true if split will happen, otherwise false
if the average radius of cid at current time is
   larger than $R_m$
    return true
else
    compute the average radius of cid at endtime
    if it is larger than $R_m$
        return false
    else
        return true
end willSplit.
```

Figure 5. The algorithm deciding whether the split will happen

```
Split(cid, starttime,endtime)
Input: MMC cid, split time starttime and
       the end time of the interval endtime
Output: A list of the deleted objects ID idlist
update cid.cf, cid.center to starttime
while willSplit(cid, endtime) is true
    for each object O in cid
        compute DM(O, cid) and record the ID of the
        object with the maximum DM
    delete the object with the maximum DM from cid
    add the ID of this object into idlist
return idlist
end Split.
```

Figure 6. Split algorithm

To avoid multiple splits of a MMC during *minUI*, we propose a new approach to handle the split event. When a MMC with the ID *cid* splits at time *t*, we first compute the distance between each object in *cid* and the center of *cid* according to (4). Then we delete the object with the maximum distance from *cid* and check whether *cid* will split again during *minUI*. If it will, we repeat the process above. Otherwise, the split ends and we check whether *cid* changes its belonging grid. The algorithm is shown in Fig. 6.

After the split, we check whether *cid* can be merged. Then we build a new MMC *newcid* for the deleted objects. If the average radius of *newcid* is less than R_m, we check whether it can be merged with other MMC except *cid*. Otherwise we just add the split event about *newcid* into *Q*.

E. Merge

The merge operation first searches for a MMC for merging. The search process is similar to that in the object insertion algorithm. To find a MMC that can be merged with MMC *cid*, we get the grid *G* containing *cid* via the hash table. Then we search the MMCs that have enough space to absorb *cid* in *G* and the neighbors of *G*. If the MMC that can absorb *cid* without split during *minUI* exists, it's what we want. Otherwise, we choose the MMC which have the latest split time after absorbing *cid*. The details of this algorithm are shown in Fig. 7.

```
FindMMC(cid, endtime)
Input: MMC cid and the end time of the interval endtime
Output: MMC $cid_1$ to be merged with cid
G = gridhash(cid)   //G is the belonging grid of cid
for each MMC $cid_1$ except cid in G and G.neighblist
    if $cid_1$.objn + cid.objn≤C   //C is MMC capacity
        vcf = $cid_1$.cf + cid.cf
        compute the split time during the interval according to vcf
        if the merged MMC will not split during the interval
            return $cid_1$
        else
            record $cid_1$ with the latest split time
return $cid_1$
end FindMMC.
```

Figure 7. The algorithm of finding a MMC for merging

After finding the MMC *cid*$_1$, we merge it with *cid*. We first add all the object IDs which are in *cid*$_1$*.objid* into *cid.objid* and modify the hash table *MMChash*. Then we update *objn*, *cf* and *center* of cid. At last, we update the split event about *cid* in *Q* and remove the MMC *cid*$_1$.

VI. EXPERIMENTS

This section presents the results of our experiments. We first introduce the experiment settings and data, and then compare our approach with other algorithms.

A. Experiment Settings and Data Preparation

All the experiments are conducted on an Intel Core 2 Quad 2.66 GHz PC with 3.25 GB RAM. We use synthetic data sets generated by our data generator. The whole space is a square space of size 32768×32768 units. Objects start at a random position in the space with random velocity from 0 to 5. We set *minUI* to 10 seconds and at the end of each *minUI*, we randomly choose parts of the objects and randomly change their velocity within 0 to 5. The total continuous clustering time is set to 1000 seconds.

We compare our grid-based approach (GridCMO) with other three approaches not based-on grid: the algorithm that handles split by removing the farthest objects (RECMO), the one using the split algorithm in [6] (DVCMO) and another one using the same split algorithm as our approach but not based-on grid (CMO). We conduct the experiments on 7 data sets with different size and run each algorithm 50 times on each data set.

B. Clustering Time

We compare the clustering speed of the four methods. The average clustering speed of the four algorithms is shown in Fig. 8.

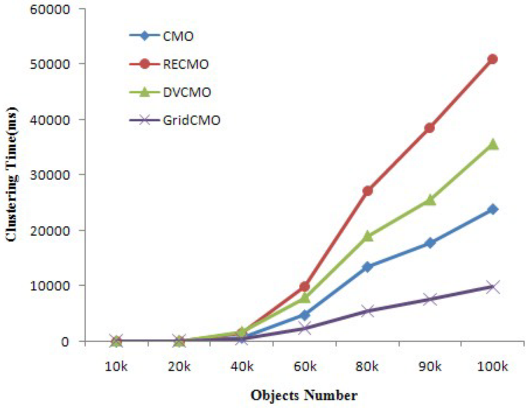


Figure 8. Average clustering speed

As can be seen from Fig. 8, GridCMO and CMO are faster than RECMO and DVCMO, which validates the efficiency of our split algorithm. GridCMO is the fastest one. This suggests that our dynamic grids accelerate the clustering. Moreover, the gaps between GridCMO and the other three methods are increasing as the data size increases, which indicates the dynamic grids employed in our approach are efficient especially for large data sets.

C. *Average Radius*

We proceed to compare the average radius of the MMCs obtained by the four methods, which can measure the conpactness of MMCs generated by these algorithms. The results are shown in Fig. 9.

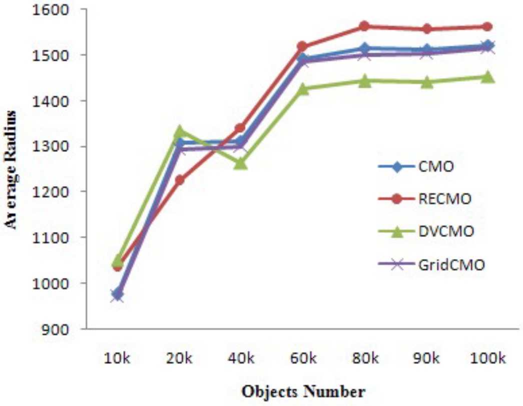


Figure 9. Avarage radius

As can be seen from Fig. 9, DVCMO has the minimum average radius because it handles the split by dividing the MMC into two parts with smaller radiuses, while the other three algorithms handle the split by deleting some “extreme” objects and aim to maintain a MMC as long as possible. GridCMO and CMO have the similar average radiuses which are close to that of DVCMO. This indicates that our split algorithm can keep the compactness of MMCs. RECMO has the maximum average radius because it only removes one object from the MMC each time it handles the split.

VII. CONCLUSION

This paper proposes an efficient grid-based approach for continuous clustering of moving objects. We develop an efficient split algorithm to handle the split of clusters, which avoids multiple splits of one cluster during a period of time. Also, we employ dynamic grids to narrow the searching area when merging clusters. The experimental evaluation has been conducted and validates that both our split algorithm and the dynamic grids accelerate the clustering as well as keep the compactness of the clusters. Our future work aims at applying the clustering scheme in the real world.

ACKNOWLEDGMENT

This work was supported by China 973 Fundamental R&D Program (No.2005CB321903) and China 863 High-tech Program (No.2009AA043305).

REFERENCES

[1] J. Macqueen, “Some Methods for Classification and Analysis of Multivariate Observations,” Proc. Fifth Berkeley Symp. Math. Statistics and Probability, pp. 281-297, 1967.

[2] T. Zhang, R. Ramakrishnan, and M. Livny, “BIRCH: An Efficient Data Clustering Method for Very Large Databases,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’96), pp. 103-114, 1996.

[3] M. Ankerst, M. Breunig, H.P. Kriegel, and J. Sander, “OPTICS: Ordering Points to Identify the Clustering Structure,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’99), pp. 49-60, 1999.

[4] Y. Li, J. Han, and J. Yang, “Clustering Moving Objects,” Proc. 10th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD ’04), pp. 617-622, 2004.

[5] Q. Zhang and X. Lin, “Clustering Moving Objects for Spatio-Temporal Selectivity Estimation,” Proc. 15th Australasian Database Conf. (ADC ’04), pp. 123-130, 2004.

[6] C.S. Jensen, Dan Lin, and Beng Chin Ooi, “Continuous Clustering of Moving Objects,” IEEE Transl. on Knowledge and Data Engineering, vol. 19, pp. 1161-1174, 2007.

[7] Chih Lai and E.A.Heuer, “Efficiently maintaining moving micro clusters for clustering moving objects,” Proc. 3rd IEEE Int’l Conf. on System of Systems Engineering (SoSE '08), pp. 1-6, 2008.

[8] W. Wang, J. Yang, and R. Muntz, “Sting: A Statistical Information Grid Approach to Spatial Data Mining,” Proc. 23rd Int’l Conf. Very Large Data Bases (VLDB ’97), pp. 186-195, 1997.

[9] R. Ng and J. Han, “Efficient and Effective Clustering Method for Spatial Data Mining,” Proc. 20th Int’l Conf. Very Large Data Bases (VLDB ’94), pp. 144-155, 1994.

[10] S. Guha, R. Rastogi, and K. Shim, “CURE: An Efficient Clustering Algorithm for Large Databases,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’98), pp. 73-84, 1998.

[11] G. Karypis, E.-H. Han, and V. Kumar, “Chameleon: Hierarchical Clustering Algorithm Using Dynamic Modeling,” Computer, vol. 32, no. 8, pp. 68-75, Aug. 1999.

[12] S. Har-Peled, “Clustering Motion,” Discrete and Computational Geometry, vol. 31, no. 4, pp. 545-565, 2003.

[13] P. Kalnis, N. Mamoulis, and S. Bakiras, “On Discovering Moving Clusters in Spatio-Temporal Data,” Proc. Ninth Int’l Symp. Spatial and Temporal Databases (SSTD ’05), pp. 364-381, 2005.

[14] J. Rosswog and K. Ghose, “Accurately clustering moving objects with adaptive history filtering,” Proc. 24th Int’l Symp. Computer and Information Sciences (ISCIS 2009), pp. 657-662, 2009.

Submesh Allocation in 2D-Mesh Multicomputers: Partitioning at the Longest Dimension of Requests

Sulieman Bani-Ahmad
Department of Information Technology
Al-Balqa Applied University
Al-Salt, Jordan
sulieman@case.edu

Abstract*-- Two adaptive noncontiguous allocation strategies for 2D-mesh multicomputers are proposed in this paper. The first is first-fit-based and the second is best-fit-based. That is; for a given request, the proposed first-fit-based approach tries to find a free submesh using the well-known first-fit strategy, if it fails, the request at hand is partitioned into two *sub-requests* that are allocated using the first-fit approach. Partitioning is performed at the longest dimension of the request. That is, for a given request of size αxβ and assuming β>α, the two partition-sizes are αx(β-1) and αx1 after removing one from the longest dimension of the request. The two new sub-requests are then allocated using the first-fit strategy. This procedure continues recursively until the request is fulfilled. The second approach is also based on *PArtitioning at the Longest Dimension* (PALD) of requests but a best-fit approach is used to allocate requests and sub-requests. The partitioning mechanism aims at (i) lifting the condition of contiguity, and (ii) at the same time maintaining *good* level of contiguity. Removing one from the longest dimension of a request is expected to produce two sub-requests one of which is relatively big and as close as possible to the square-shape and, thus; reducing communication latency caused by non-contiguity. Using extensive simulations, we evaluated the proposed strategies and compared them with previous contiguous and non-contiguous strategies. Simulation outcomes clearly show the proposed PALD-based schemes produce the best *Average Response Time* (ART), the *Average System Utilization* (ASU) and also produce relatively low communication overhead.

Keywords- Multicomputer; 2D mesh; Non-contiguous Allocation; Request Partitioning.

I. INTRODUCTION

In parallel systems, processors are connected through interconnection network; one of the most widely used architectures is the 2D and 3D mesh-connected architectures. This is because mesh architecture is simple, regular and scalable [4, 14]. Several recent commercial and experimental parallel computers have been built based these architectures such as the IBM BlueGene/L and the Intel Paragon [4].

Processor allocation in 2D-Mesh multicomputer is a major issue as it significantly affects the performance of any parallel system [4]. Processor allocation is concerned with the way for allocation submesh to a job request. Many processor allocation strategies in literature try to allocate a submesh, i.e., a contiguous set of processing units, of the same size and shape of request [1, 3, 4, 5, 7, 12, 21]. This, however, may produce low level of system utilization and cause either internal or external fragmentation or both [2, 18]. Internal fragmentation occurs when the number of processors allocated to a job is more than that it requested [16]. External fragmentation, on the other hand, occurs when enough number of idle processors is available in the system but cannot be assigned to the scheduled job because of the requirement of contiguity [2]. Several studies have attempted to reduce or solve external fragmentation [2, 9, 6, 14, 16, 18], one of the proposed solutions is to use non-contiguous allocation.

In non-contiguous allocation the contiguity condition is relaxed [2]; therefore, a job can execute on multiple disjoint smaller sub-meshes rather than always waiting until a single sub-mesh of the requested size and shape is available [2, 9, 14, 18]. Studies show that non-contiguous allocation of requests may solve the drawbacks of contiguous allocation; non-contiguous allocation strategies produce relatively high system utilization and eliminate fragmentation. However, since communication between processors running the same job can be indirect due to non-contiguity [16], communication latency is usually high. However, the introduction of wormhole routing [17] has lead researchers to consider noncontiguous allocation on multicomputers with a long communication distances, such as the 2D mesh [2, 14, 18]. One of main advantages of wormhole routing over earlier communication schemes, e.g., store-and-forward, is that message latency is less dependent on the distance traversed by the message from source to destination [2, 17]. Thus, non-contiguous allocation has recently received attention of researchers.

Partitioning allocation requests in existing non-contiguous allocation schemes can be performed in multiple ways. For example, allocation requests are subdivided into two equal partitions in [2]. The sub-partitions are recursively subdivided into further smaller sub-requests if allocation fails for any of them. In the study of [18], a promising strategy (MBS) expresses the allocation request as a base-4 number, and bases allocation on this expression.

In this paper, two *adaptive* noncontiguous allocation strategies for 2D-mesh multicomputers are proposed and evaluated through simulation. The first is a first-fit-based approach that tries to find a contiguous set of processing units of the same shape and size to the request at hand using the well-known first-fit approach. If it fails, the request at hand is divided into two sub-requests after removing one from the longest dimension of the request. That is, for a given request of size αxβ and assuming β>α, the two partition-sizes are αx(β-1) and αx1 after removing one from the longest dimension of the request. The two new sub-requests are then allocated using the first-fit approach again. This procedure continues recursively until the request is fulfilled. This approach is referred to a PALD-FF for ***PA***rtitioning at the ***L***ongest ***D***imension with First-Fit.

The second approach is also PALD-based. However, the best-fit (BF) allocation strategy is used to allocate requests and sub-requests. The used partitioning mechanism aims at (i) lifting the condition of contiguity, and (ii) at the same time maintaining *good* level of contiguity. Removing one from the longest dimension of a request is expected to produce two sub-requests one of which is relatively big and as close as possible to be square-shaped and, thus; reducing communication latency caused by non-contiguity.

Using extensive simulations, we evaluated the proposed strategies and compared them with previous promising strategies. Simulation outcomes clearly show the proposed PALD-based schemes produces the best *Average Response Time* (ART), the *Average System Utilization* (ASU) and produce relatively low communication overhead. The performance of PALD-FF and PALD-BF is compared against the performance of the MBS non-contiguous allocation strategy. This strategy is selected as it has been shown to perform well in [18]. Furthermore, proposed approaches are also compared against the contiguous First-Fit and Best-Fit strategies as this has been used in several previous related studies [2, 3, 18]. The proposed approaches are tested under two job scheduling strategies, namely; first-Come-First-Served (FCFS) and Shortest-Service-Demand-First (SSD). In FCFS, the allocation request that arrived first is scheduled for allocation first. In SSD, the job with the shortest service demand is scheduled first [11]. The FCFS scheduling strategy is chosen as it is fair and it is widely used in other similar studies [2, 3, 4, 6, 14], while the SSD scheduling strategy is used to avoid performance loss due to blocking [11].

II. RELATED WORK

In this section, we provide an overview of some existing contiguous and non-contiguous allocation strategies.

A. Non-Contiguous Allocation Strategies

The ***First Fit (FF)*** strategy is a *contiguous allocation strategy*. This scheme start search at the lowest leftmost node in mesh, and put a virtual grid that's equal size request, and then shifts by one column to the right until first large enough free submesh is found [13]. The ***Best fit (BF)*** is also a contiguous allocation strategy. This scheme is the same as first fit scheme, but it reserves a submesh after consider all large enough free submeshes and chooses the closest requests, i.e., the submesh with minimal leftovers is selected [13]. We use both strategies to search for free submeshes for the partitioned requests as should be shortly illustrated more.

B. Non-Contiguous Allocation Strategies

The introduction of wormhole routing [17] has made communication latency less sensitive to the distance traversed by between communicating entities [2]. This has made allocating a job to non-contiguous processors reasonable, in terms of performance, in networks characterized by a relatively long-diameter, such the 2D mesh. Non-contiguous alleviates the contiguity and thus allowing jobs to be executed without waiting for contiguous set of idle nodes [2, 14].

In the Paging strategy, for instance [18], the entire 2D mesh is virtually sub-divided into pages or sub-meshes of equal sides' length of 2^i where *i* is a positive integer number that represents the index parameter of the paging approach. The pages are indexed according to several indexing schemes.

In the Multiple Buddy System (MBS) strategy, the mesh of the system at hand is divided into non-overlapping square sub-meshes with side lengths that are powers of 2. The number of processors, p, requested by a job is factorized into a base-4 block. If a required block is unavailable, MBS recursively searches for a larger block and repeatedly breaks it down into *four* buddies until it produces blocks of the desired size. If that fails, the requested block is further broken into four sub-requests until the job is allocated [18].

In the Adaptive Non-Contiguous Allocation (ANCA) strategy work differently. ANCA first attempts to allocate the job at hand contiguously. If the allocation attempt fails, it partitions the request into two equi-sized sub-requests. These sub-frames are then allocated to available locations, if possible; otherwise, each of these sub-requests is recursively further partitioned into two sub-requests, and then ANCA tries to map these sub-requests to available locations [2].

Maintaining a *good* level of contiguity can prove useful in non-contiguous allocation. In Paging, there is some degree of contiguity because of the indexing schemes used. Contiguity can also be increased by increasing the index parameter. However, this may produce internal processor fragmentation for large index sizes [18]. In MBS, contiguous allocation is explicitly sought only for requests with sizes of the form 2^{2n}, where *n* is a positive integer.

An issue with the ANCA strategy is that it can disperse the allocated sub-meshes more than it is necessary through *over partitioning*. Over-partitioning may cause skipping over the possibility of identifying and thus allocating larger free sub-meshes for a large part of the request at hand which has been shown to maintain a higher level of contiguity [15]. Thus the communication overhead can be reduced by adaptively and gradually partitioning allocation requests into as large as possible contiguous sub-meshes.

III. THE PROPOSED ALLOCATION STRATEGY

The target system is a W × L two-dimensional mesh, where W and L are the width and the length of the mesh, respectively. Every processor is denoted by a pair of coordinates, namely; x and y, where $0 \leq x < W$ and $0 \leq y < L$ [14]. Each processor is connected by bidirectional communication links to its neighbor processors.

In this paper, two *adaptive* noncontiguous allocation strategies for 2D-mesh multicomputers are proposed and evaluated. The first is a first-fit-based approach that tries to find a contiguous set of processing units of the same shape and size to the request at hand using the well-known first-fit approach. If it fails, the request at hand is divided into two sub-requests after removing one from the longest dimension of the request. That is, for a given request of size αxβ and assuming β>α, the two partition-sizes are αx(β-1) and αx1 after removing one from the longest dimension of the request. The two new sub-requests are then allocated using the first-fit approach again. This procedure continues recursively until the request is fulfilled. This approach is referred to a PALD-FF for ***PA***rtitioning at the ***L***ongest ***D***imension with First-Fit.

```
Procedure PALD-FF(a, b):
Begin
        JobSize = a × b
        If (number of free processors < JobSize) return failure
        List AllocatedPIDs={}; // the list of PIDs allocate to the current job
        Return PALD-FFAllocate(a, b, AllocatedPIDs);
End

Procedure PALD-FFAllocate (a, b, AllocatedPIDs)
Begin
        S(x, y) = FIND_FF (S(a, b); // FIND_BF for PALD-BF allocation
        If (S(x, y)  != null)
          Add the PIDs of S to the list AllocatedPIDs;
        Else
        {
                If(a>=b)
                        α1= a-1;  β1=b; α2= 1;  β2=b;
                else
                        α1= a;  β1=b-1; α2= a;  β2=1;
                PALD-FFAllocate (α1, β1, AllocatedPIDs);
                PALD-FFAllocate (α2, β2, AllocatedPIDs);
        }
End
```

Figure 1: Pseudo code for the PALD-FF allocation strategy

The second approach is also PALD-based. However, the best-fit (BF) strategy is used to allocate requests and sub-requests. The used partitioning mechanism aims at (i) lifting the condition of contiguity, and (ii) at the same time maintaining *good* level of contiguity. Removing one from the longest dimension of a request is expected to produce two sub-requests one of which is relatively big and as close as possible to be square-shaped and, thus; reducing communication latency caused by non-contiguity.

The proposed PALD-based approach combines the desirable features of both contiguous and non-contiguous allocation. The well-known first-fit (FF) and best-fit (BF) strategies are used here to search for available submeshes. A 13eudo code for the allocation procedure of PALD-FF strategy is shown in Figure 1. The PALD-BF is similar except that the Find_BF() is called instead of Find_FF() to allocate partitions of the parallel job at hand. Notice that allocation always succeeds as long as enough free processors are available in the mesh. The key idea in the proposed PALD approach is to try allocating the largest submeshes possible.

IV. EXPERIMENTAL SETUP AND SIMULATION OUTPUT

The current study is simulation-based with the ProcSimity simulator is to be used. The simulated multicomputer system consists of 256 multicomputers connected through a 2-dimensional mesh network of dimensions W and L W=L=16. The routing mechanism to be used is the wormhole routing [10, 20] with packet size of 8 units and a buffer of size 1 unit and a routing delay of 3 units. The router uses XY routing to direct messages from their source to destination. Message sizes are considered to be of length 8 units. Job size conforms the exponential distribution with mean width and length being W/2 (or L/2). The execution times of jobs conforms the uniform distribution.

To maintain good levels of accuracy, each simulation experiments is repeated 10 times with a total of 1000 jobs are to be simulated in each time. The readings are 95% accurate with a maximum percentage error of 5%. The scheduling mechanisms considered in our experiments are (i) First-Come-First-Serve, or FCFS and (ii) the Shortest Service Demand first, or SSD mechanisms. The simulation outputs are:

(i) Average Response Time (ART): The response time is the time from the submission of request until the first real response produced for jobs. (ii) ***Average system utilization (ASU)***: The average of keeping the processors within a system as busy as possible, this value between 0 and 1. (iii) ***Average Packet Blocking Time (APBT)***: The average amount of time the head of the message is blocked at each station while routing the message over the path from source to destination. (iv) ***Average Packet Latency (APL)***: The average of the time that all packets within job will be sent between processors.

V. EXPERIMENTAL RESULTS AND OBSERVATIONS

In this section, the results from simulations that have been carried out to evaluate the performance of the proposed algorithm are presented and compared against those of MBS, BF and FF. The proposed allocation algorithm is implemented and later integrated with the ProcSimity simulation tool [8, 13]. Each simulation run consists of 1000 completed jobs. Simulation results are averaged over enough independent runs so that the confidence level is 95% and the relative errors do not exceed 5%.

Next we present our experimental results and observations. Parallel jobs usually communicate with each other using one-to-all or all-to-all communication patterns [9, 17, 18]. We did our experiments using both pattern but focused more on the all-to-all communication pattern as it produces message collision than the one-to-all communication pattern and is known to be a weak point for non-contiguous allocation algorithms [9]. The independent variable in the simulation is the system load. The notation *<allocation strategy>(<scheduling strategy>)* is used to represent the strategies in the performance figures, as in [15]. For example, PALD-FF(FCFS) refers to the PALD-FF processor allocation strategy under the scheduling strategy FCFS.

A. Mean response time criteria

In Figures 2 through 4, the mean job response time of jobs is plotted against the system load for the one-to-all and all-to-all communication patterns under the FCFS and SSD scheduling mechanisms. The figures reveal that PALD-based allocation strategies produce less response times and, thus, perform better than all other strategies. This is more clear under the SSD scheduling mechanism. PALD-FF is substantially superior to the FF and PALD-BF is also superior to BF. For all-to-all communication pattern both tested PALD-based allocation strategies outperformed contiguous allocation strategies.

Figure 5 shows the four allocation strategies compared together in terms of response time. Considering the same system settings figure 5 shows that PALD-based approaches outperform non-PALD-based ones.

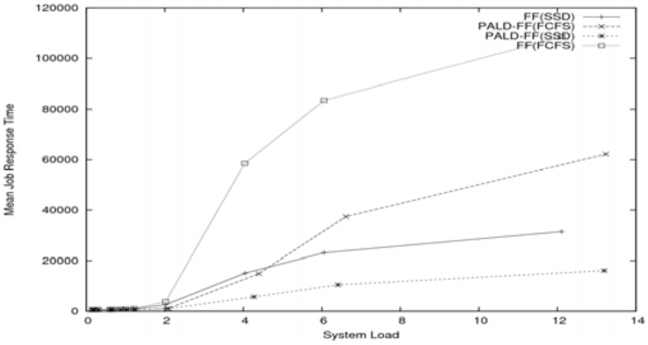


Figure 2: Mean response time in FF and PALD-FF strategies under the FCFS and the SSD scheduling mechanisms and one-to-all communication pattern.

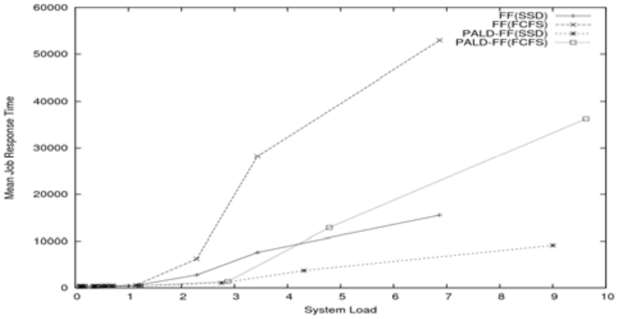


Figure 3: Mean response time in FF and PALD-FF strategies under the FCFS and the SSD scheduling mechanisms and all-to-all communication pattern.

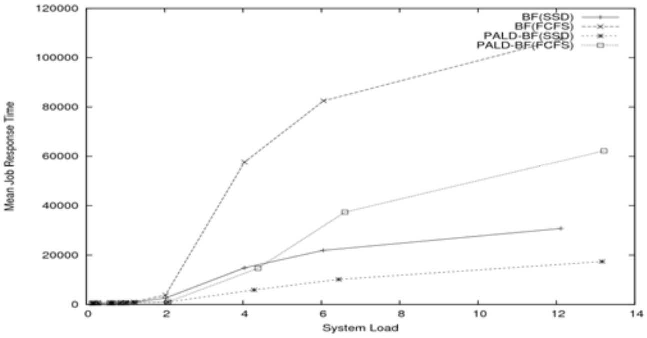


Figure 4: Mean response time in BF and PALD-BF strategies under the FCFS and the SSD scheduling mechanisms and one-to-all communication pattern.

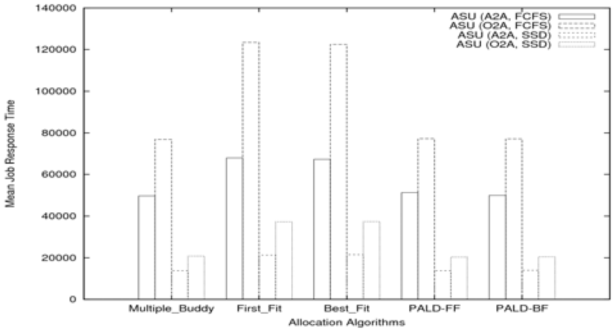


Figure 5: Mean response time in MBS, FF, BF, PALD-FF and PALD-BF strategies under both scheduling mechanisms, both communication patterns.

B. Percent system utilization criteria

Figures 6 through 9 depict the mean system utilization of the tested allocation strategies, namely; FF, BF, PALD-FF, PALD-BF and MBS, for the two communication patterns considered and under the FCFS and SSD scheduling mechanisms. Figures 6 and 7 depict the percent system utilization in FF and PALD-FF allocation strategies under the FCFS and the SSD scheduling mechanisms and one-to-all and all-to-all communication patterns. Similarly, Figures 8 and 9 depict the percent system utilization in BF and PALD-BF allocation strategies under the FCFS and the SSD scheduling mechanisms and both communication patterns.

Figures 6 through 9 reveal that the PALD-based strategies produce higher system utilization and. This is more clear under the SSD scheduling mechanism. PALD-FF and PALD-BF showed around 70% higher system utilization than the FF and BF approaches at the points where the system is heavily loaded, respectively. This observation applied for both communication patterns. This observation can be explained as follows, contiguous allocation produces high external fragmentation, which means that allocation is less likely to succeed. Consequently, system utilization becomes low. The proposed approaches have the ability to eliminate both internal and external processor fragmentation, and thus, produce higher system utilization.

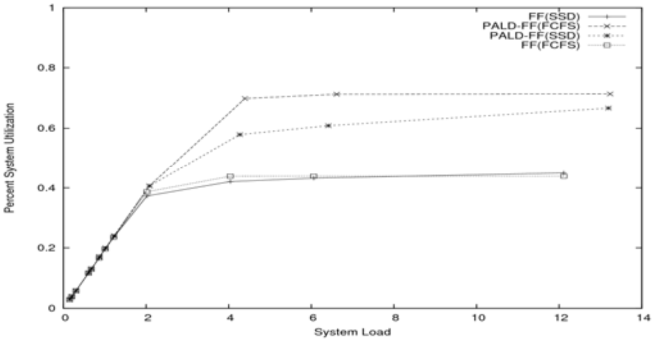


Figure 6: System utilization in FF and PALD-FF strategies under the FCFS and the SSD scheduling mechanisms and one-to-all communication pattern.

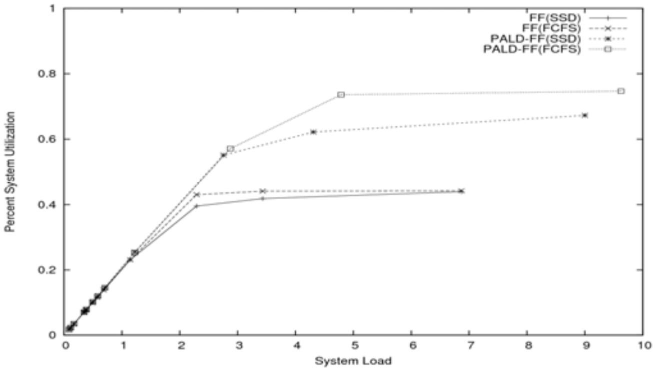


Figure 7: System utilization in FF and PALD-FF strategies under the FCFS and the SSD scheduling mechanisms and all-to-all communication pattern.

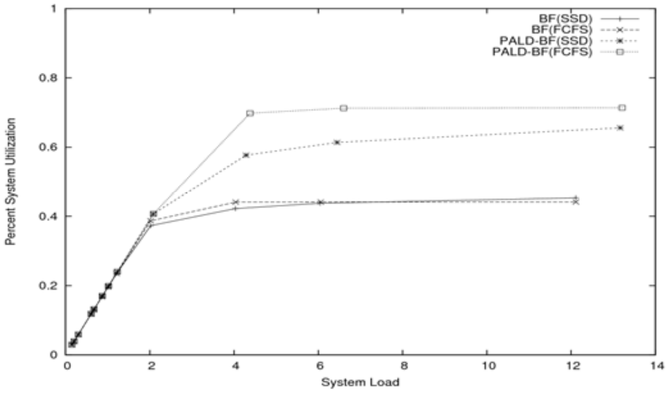


Figure 8: System utilization in BF and PALD-BF strategies under the FCFS and the SSD scheduling mechanisms and one-to-all communication pattern.

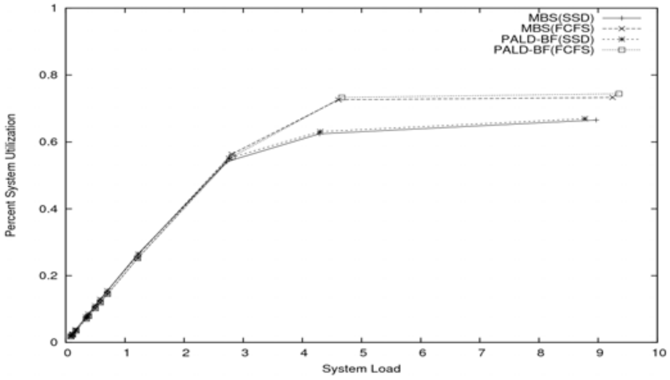


Figure 9: System utilization in MBS and PALD-BF strategies under the FCFS and the SSD scheduling mechanisms and all-to-all communication pattern.

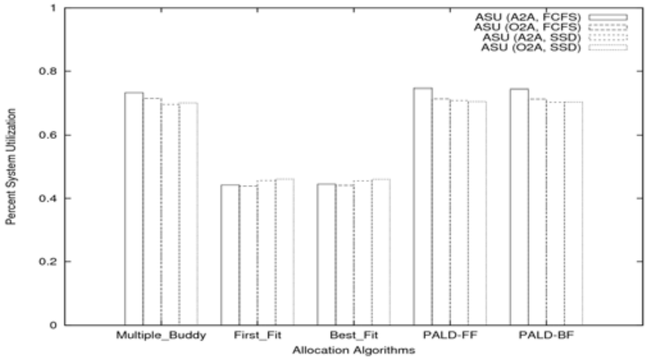


Figure 10: System utilization in MBS, FF, BF, PALD-FF and PALD-BF strategies under both scheduling mechanisms, both communication patterns.

C. *Communication Overhead*

We have measured other performance criteria for the non-contiguous allocation strategies. These are the mean packet latency (MPL) and the mean packet blocking time (MPBT). Figure 11 shows that the MPL for the tested allocation strategies for all-to-all communication pattern and under the two considered scheduling mechanisms. It can be seen that PALD-FF and PALD-BF strategies have lower MPL values than MBS strategy under the two scheduling strategies FCFS and SSD for the all-to-all communication pattern. This conclusion is compatible with the values of the mean turnaround time shown above.

To summarize, the above performance results demonstrate that PALD-FF and PALD-BF strategies are superior to all other strategies considered in this paper; including the case when contention is heavy (the communication pattern is all-to-all). Figure 12 shows that the MPBT for the tested allocation strategies under the two considered scheduling mechanisms is less than that of MBS strategy.

One concern in PALD-based allocation strategies is that requests may get over-partitioned. This results in allocating dispersed multicomputers to parallel jobs. To test that, we repeated our experiments and allowed for giving a control over the maximum number of blocks allowed to any allocated job (MBPJ). Figure 13 illustrates the observed relationship between MBPJ (the x-axis) and the average system utilization (the y-axis). At an MBPJ value of 14, we found that the system utilization reaches a maximum saturation value of around 0.92. Thus, placing this limit helps in (i) preventing over-partitioning and (ii) keeping the allocation time complexity of PALD allocation strategies to be the same as that of the contiguous allocation strategy used (the FF or BF).

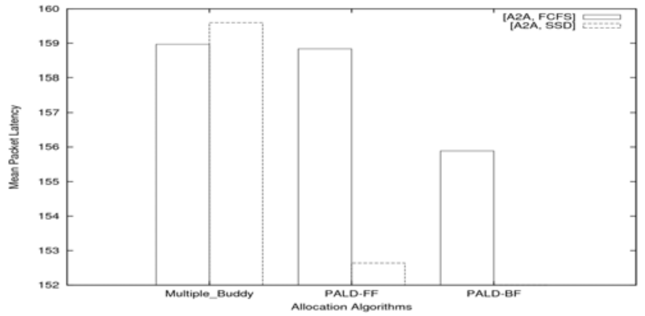


Figure 11: Mean packet latency in MBS, PALD-FF and PALD-BF allocation strategies under the FCFS and the SSD scheduling mechanisms, all-to-all communication patterns.

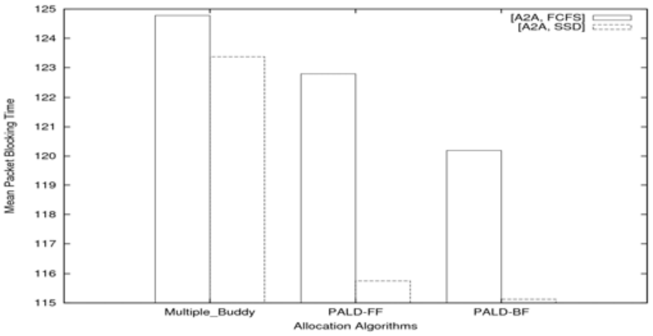


Figure 12: Mean packet blocking time in MBS, PALD-FF and PALD-BF allocation strategies under the FCFS and the SSD scheduling mechanisms, all-to-all communication patterns.

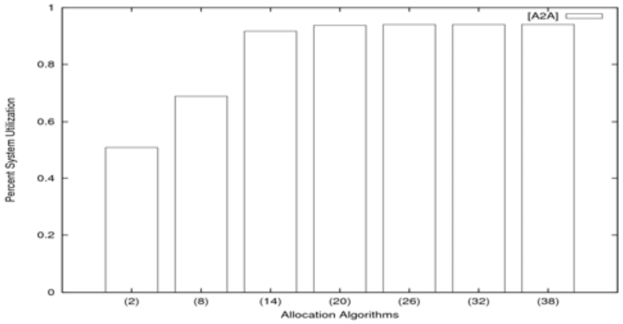


Figure 13: System utilization vs partitioning limit for PALD-BF allocation strategy under the FCFS scheduling mechanism, all-to-all communication patterns.

VI. CONCLUSIONS

Two adaptive noncontiguous allocation strategies are proposed in this paper. The first is first-fit-based and the second is best-fit-based. That is; for a given request, the proposed first-fit-based approach tries to find a free submesh using the well-known first-fit strategy, if it fails, the request at hand is partitioned into two *sub-requests* that are allocated using the first-fit approach. Partitioning is performed at the longest dimension of the request (removing one from the longest dimension of the request at hand). The two new sub-requests are then allocated using the first-fit or the best-fit approaches. This procedure continues recursively until the request is fulfilled. The second approach is also based on ***PA***rtitioning at ***L***ongest ***D***imension (PALD) of requests but a best-fit approach is used to allocate requests and sub-requests.

The partitioning mechanism aims at (i) lifting the condition of contiguity, and (ii) at the same time maintaining *good* level of contiguity. Removing one from the longest dimension of a request is expected to produce two sub-requests one of which is relatively big and as close as possible to the square-shape and, thus; reducing communication latency caused by non-contiguity. Using extensive simulations, we evaluated the proposed strategies and compared them with previous contiguous and non-contiguous strategies. Simulation outcomes clearly show the proposed PALD-based schemes produce the best *Average Response Time* (ART), the *Average System Utilization* (ASU) and produce relatively low communication overhead.

REFERENCES

[1] B. S. Yoo and C. R. Das, “A Fast and Efficient Processor Allocation Scheme for Mesh-Connected Multicomputers”, IEEE Transactions on Parallel & Distributed Systems, vol. 51, no. 1, IEEE Computer Society, Washington, USA, January 2002, pp. 46-60.

[2] C. Y. Chang and P. Mohapatra, “Performance improvement of allocation schemes for mesh-connected computers”, Journal of Parallel and Distributed Computing, vol. 52, no. 1, Academic Press, Inc. Orlando, FL, USA, July 1998, pp. 40-68.

[3] G.-M. Chiu and S.-K. Chen, “An efficient submesh allocation scheme for two-dimensional meshes with little overhead”, IEEE Transactions on Parallel & Distributed Systems, vol. 10, no. 5, IEEE Press, Piscataway, NJ, USA, May 1999, pp. 471-486.

[4] I. Ababneh, “An efficient free-list submesh allocation scheme for two-dimensional mesh-connected multicomputers”, Journal of Systems and Software, vol. 79, no. 8, Elsevier Science Inc., New York, NY, USA, August 2006, pp. 1168-1179.

[5] I. Ismail and J. Davis, “Program-based static allocation policies for highly parallel computers”, Proc. IPCCC 95, IEEE Computer Society Press, Scottsdale, AZ, USA, 28-31 Mar 1995, pp. 61-68.

[6] K. H. Seo, “Fragmentation-Efficient Node Allocation Algorithm in 2D Mesh-Connected Systems”, Proceedings of the 8th International Symposium on Parallel Architecture, Algorithms and Networks (ISPAN’05), IEEE Computer Society Press, Washington, DC, USA, 7-9 December, 2005, pp. 318-323.

[7] K. Li and K. H. Cheng, “A Two-Dimensional Buddy System for Dynamic Resource Allocation in a Partitionable Mesh Connected System”, Journal of Parallel and Distributed Computing, vol. 12, no. 1, Elsevier Science, CA, USA, May 1991, pp. 79-83.

[8] K. Windisch, J. V. Miller, and V. Lo, “ProcSimity: an experimental tool for processor allocation and scheduling in highly parallel systems”, Proceedings of the Fifth Symposium on the Frontiers of Massively Parallel Computation (Frontiers'95), IEEE Computer Society Press, Washington, USA, 6-9 Feb 1995, pp. 414-421.

[9] K. Suzaki, H. Tanuma, S. Hirano, Y. Ichisugi, C. Connelly, and M. Tsukamoto, “Multi-tasking Method on Parallel Computers which Combines a Contiguous and Non-contiguous Processor Partitioning Algorithm”, Proceedings of the Third International Workshop on Applied Parallel Computing, Industrial Computation and Optimization, Springer-Verlag, UK, 1996, pp. 641-650.

[10] L. M. Ni and P. K. McKinley. A Survey of Wormhole Routing Techniques in Direct Networks. Computer 26, 2 (Feb. 1993), pp 62-76. DOI= http://dx.doi.org/10.1109/2.191995.

[11] P. Krueger, T. Lai, and V. A. Radiya, “Job scheduling is more important than processor allocation for hypercube computers”, IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 5, IEEE Press, Piscataway, NJ, USA, May 1994, pp. 488-497.

[12] P. J. Chuang and N.-F. Tzeng, “Allocating precise submeshes in mesh connected systems”, IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 2, IEEE Press, USA, February 1994, pp. 211-217.

[13] ProcSimity V4.3 User’s Manual, University of Oregon, 1997.

[14] S. Bani-Mohammad, M. Ould-Khaoua, I. Ababneh, and L. Machenzie, “Non-contiguous Processor Allocation Strategy for 2D Mesh Connected Multicomputers Based on Sub-meshes Available for Allocation”, Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS’06), vol. 2, IEEE Computer Society Press, USA, 2006, pp. 41-48.

[15] S. Bani-Mohammad, M. Ould-Khaoua, I. Ababneh, and L. Machenzie, “A Fast and Efficient Processor Allocation Strategy which Combines a Contiguous and Non-contiguous Processor Allocation Algorithms”, Technical Report; TR-2007-229, DCS Technical Report Series, Department of Computing Science, University of Glasgow, January 2007.

[16] T. Srinivasan, J. Seshadri, A. Chandrasekhar, and J. Jonathan, “A Minimal Fragmentation Algorithm for Task Allocation in Mesh-Connected Multicomputers”, Proceedings of IEEE International Conference on Advances in Intelligent Systems – Theory and Applications – AISTA 2004 in conjunction with IEEE Computer Society, ISBN 2-9599-7768-8, IEEE Press, Luxembourg, Western Europe, 15-18 Nov 2004.

[17] V. Kumar, A. Grama, A. Gupta, and G. Karypis, Introduction To Parallel Computing, The Benjamin/Cummings publishing Company, Inc., Redwood City, California, 2003.

[18] V. Lo, K. Windisch, W. Liu, and B. Nitzberg, “Non-contiguous processor allocation algorithms for mesh-connected multicomputers”, IEEE Transactions on Parallel and Distributed Systems, vol. 8, no. 7, IEEE Press, Piscataway, NJ, USA, July 1997, pp. 712-726.

[19] W. Mao, J. Chen, and W. Watson, “Efficient Subtorus Processor Allocation in a Multi-Dimensional Torus”, Proceedings of the 8th International Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA’05), IEEE Computer Society, Washington, DC, USA, 30 November -3 December, 2005, pp. 53-60.

[20] X. Lin, P. Mckinly, and A. Esfahanina, 1993. Adaptive Multicast wormhole Routing in 2D-mesh multicomputers. Proceeding of Parallel Architecture and Language conference (PARLE), pp 228-241.

[21] Y. Zhu, “Efficient processor allocation strategies for mesh-connected parallel computers”, Journal of Parallel and Distributed Computing, vol. 16, no. 4, Elsevier, San Diego, CA, 1992, pp. 328-337.

Separation of Noise and Signals by Independent Component Analysis

Sigeru Omatu, Masao Fujimura,
Osaka Institute of Tecnology,
Asahi-ku, Osaka, 535-8585, Osaka, Japan,
{*omatu@rsh., Fujimura@elc.*}*oit.ac.jp,*

Toshihisa Kosaka
Glory Ltd
Himeji, Hyogo, Japan
kosaka@tec.glory.co.jp

***Abstract*—A separation problem of acoustic signals and noise by using the independent component analysis (ICA) with band-pass filters is proposed. The frequency distribution of a recorded acoustic signal of the operating mechanical device can be divided into three fields, the low-frequency field, which corresponds to the frequency characteristics of the gear, the medium-frequency field, which is mixed with the frequency characteristics of the gear and the motor, and the high-frequency field, which corresponds to the frequency characteristics of the motor. Since only the medium-frequency components are the mixture of acoustic signals of gears and motors, the ICA with band-pass filters is expected to separate the acoustic signals of motors and gears more accurately than the conventional ICA. The simulation and experimental results show that the proposed method can separate the acoustic signals of motors and gears of mechanical devices successfully.**

***Keywords*-signal separation, independent component analysis, neural networks**

I. Introduction

In the quality evaluation of mechanical devices, it is important to separate the acoustic signals of motors and gears in order to identify the causes of failures. The ICA method, which is developed to solve the cocktail-party problem, can separate two independent acoustic signals from their mixtures by using the information measure of statistically independent properties [1],[2], and [3]. However, many applications in practice denote that the ICA does not perform well in separation by using the observed acoustic signals directly [4]-[5]. In order to separate the independent acoustic signals correctly, additional data processing is necessary before applying the ICA. By applying the fast Fourier transform (FFT) to a recorded acoustic signal of the operating mechanical device, we observe that its frequency distribution can be divided into three fields, the low-frequency field, which corresponds to the frequency characteristics of the gear, the medium-frequency field, which is mixed with the frequency characteristics of the gear and the motor, and the high-frequency field, which corresponds to the frequency characteristics of the motor. Since the frequencies of a motor may be harmonics of the fundamental frequencies of a gear, which causes the independence assumption of the sources to fail and affects the separation accuracy. Therefore, the mixed acoustic signals with less frequency components are expected to be separated more accurately. In this paper, the ICA with band-pass filters is used to separate the acoustic signals of gears and motors. We first record the acoustic signals of the operating mechanical devices. By applying the band-pass filters, the respective components of low- frequency, medium-frequency and high-frequency can be obtained. Then the medium-frequency components are given to the ICA. After separation, the acoustic signals of gears and motors are recovered by adding the low-frequency and high-frequency components to the separated results, respectively. In this paper, the mixtures of two independent signals are also designed to simulate the separation process of acoustic signals of a gear and a motor. Both the simulation results and the experimental results show that the better separation results can be obtained by using the mixed medium-frequency field than using the whole frequency field.

II. Principle of ICA

Let us summarize the principle of the ICA and state some problem when we use the ICA directly. For simplicity, we assume two source signals denoted by s_1 and s_1 which are statistically independent each other. Furthermore, we assume that those signals are observed by two microphones and denoted by x_1 and x_2. The relation is given by the following equations:

$$x_1 = a_{11}s_1 + a_{12}s_2x_2 = a_{21}s_1 + a_{22}s_2 \tag{1}$$

where $a_{ij}, i, j = 1, 2$ are unknown constant. Using the vector notation, we have

$$x = As, \quad x = (y_1, y_2)^t, s = (s_1, s_2)^t \tag{2}$$

where an unknown matrix A is given by

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}.$$

From the observed vector x, we estimate the original signal vector s by the following linear transform

$$\hat{s} = wx \tag{3}$$

where

$$w = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix}$$

such that the following J can be minimized

$$J = \int p(\hat{s}) log \frac{p(\hat{s})}{p(\hat{s_1})p(\hat{s_2})} d\hat{s}. \tag{4}$$

There are some algorithms to minimize J, we adopt the fast ICA developed by Hyvarinen A. *et al* [1]

ISBN: 978-1-61208-101-4

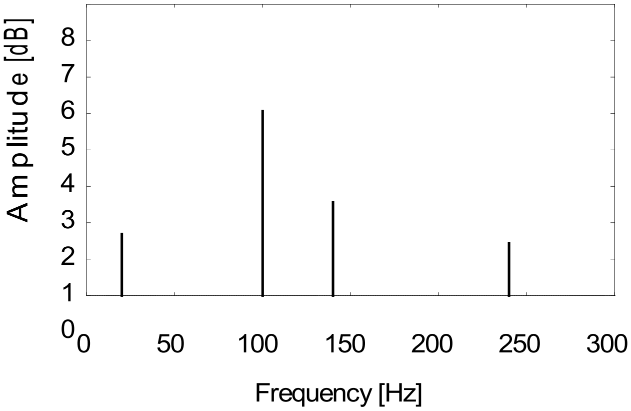


Fig. 1. Frequency characteristic of $x_1(t)$.

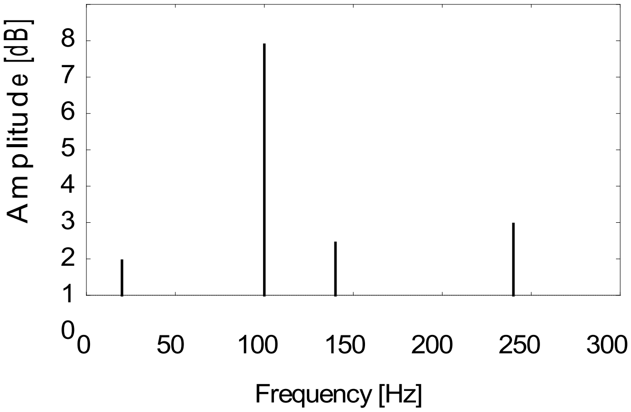


Fig. 2. Frequency characteristic of $x_2(t)$.

III. Simulation Results

We have tried the signal separation using the fast ICA algorithm in case of $f_1 = 20$Hz,$f_2 = 140$Hz,$f_3 = 100$Hz, and $f_4 = 240$Hz and $a_{11} = 2, a_{12} = 1.2, a_{21} = 1$, and $a_{22} = 1.3$. The spectra of two independent signals $x_1(t)$ and $x_2(t)$ are illustrated in Figs. 1 and 2, respectively. The corresponding spectra of recovered signals are illustrated in Figs. 3 and 4.

From these results, signal separations for these data are not good since frequencies except for the original signals are remained.

However, if we select several frequencies, some combinations of them could recover the original signals perfectly. To see this effect, we have tried the following simulation in case where f_1, f_2 and f_4 are fixed and f_3 is changed.

If we use two microphones to record the acoustic signals, we have two observed signals given by

$$x_1(t) = a_{11}s_1(t) + a_{12}s_2(t)$$
$$x_2(t) = a_{21}s_1(t) + a_{22}s_2(t).$$

We use the ICA to separate the two independent acoustic signals $s_1(t)$ and $s_2(t)$ from the observed signals $x_1(t)$ and $x_2(t)$. Table I shows the separation results where ”Y” denotes that the independent signals $s_1(t)$ and $s_2(t)$ can be separated correctly and ”N” denotes that they cannot be separated

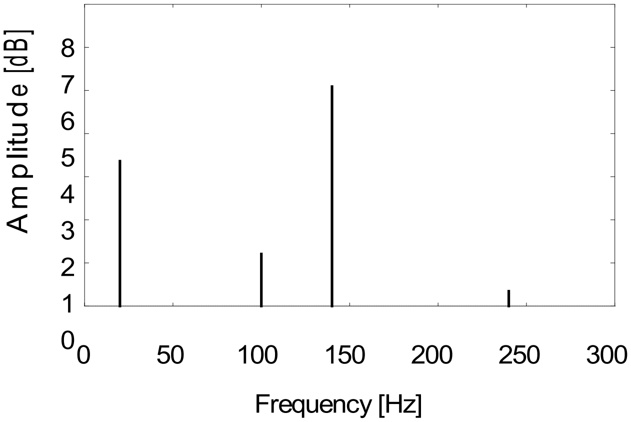


Fig. 3. Frequency characteristic of $\hat{s}_1(t)$.

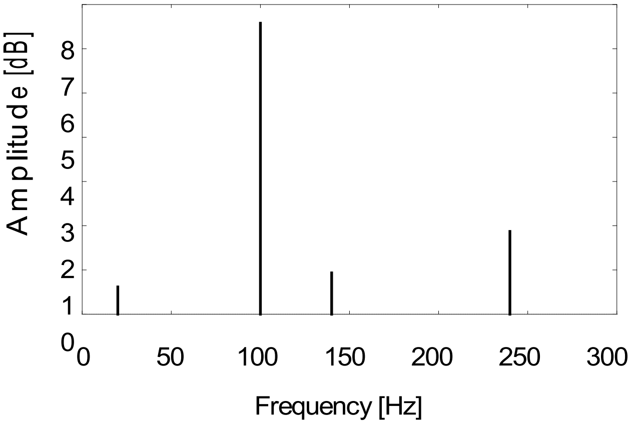


Fig. 4. Frequency characteristic of $\hat{s}_2(t)$.

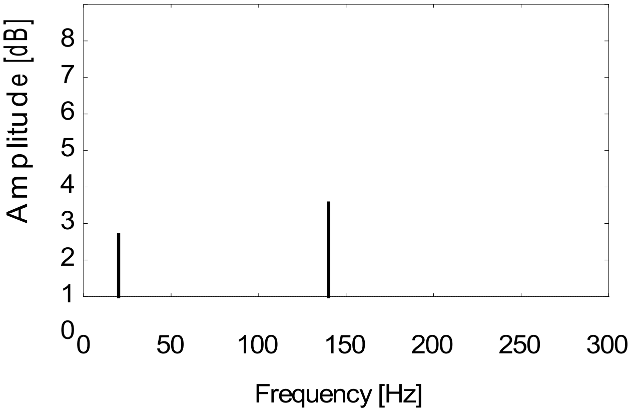


Fig. 5. Frequency characteristic of s_1.

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-101-4

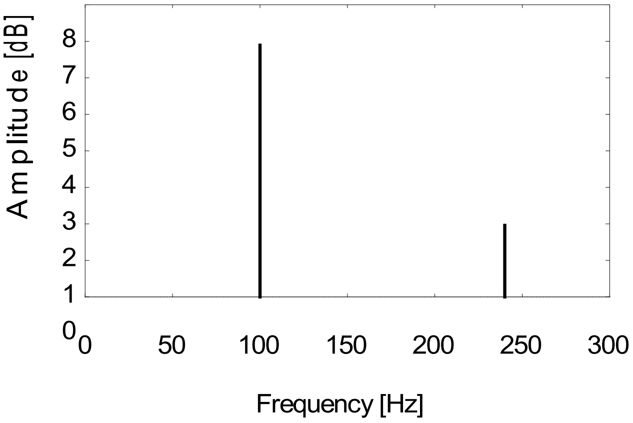


Fig. 6. Frequency characteristic of s_2.

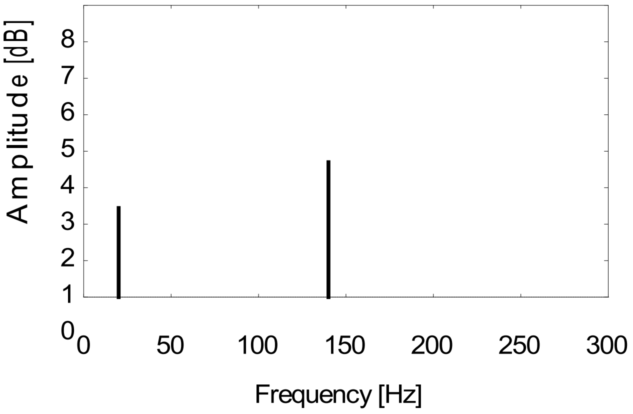


Fig. 7. Frequency characteristic of separated signal $\hat{s}_1(t)$ with band-pass filters.

correctly. From Table I, it can be seen that sometimes we fail in separating the acoustic signals $s_1(t)$ and $s_2(t)$ by using the observed signals $x_1(t)$ and $x_2(t)$ directly.

However, after filtering the frequency components f_1 and f_4 with a band-pass filter, the frequency components f_2 and f_3 can be separated successfully by using the ICA. Thus, the original acoustic signals $s_1(t)$ and $s_2(t)$ can be obtained by adding the frequency components f_1 and f_4 to the separation results of the ICA, respectively. As an example, Figs. 7 and 8 show the frequency characteristics of separated signals $\hat{s}_1(t)$ and $\hat{s}_2(t)$ by using the ICA with band-pass filters, respectively where $f_1 = 100$Hz. From these figures, it can be seen that the two acoustic signals of $s_1(t)$ and $s_2(t)$ are recovered correctly.

Similarly, other unsuccessful separation experiments of Table I are redone by using the ICA with band-pass filters. The simulation results show that all the signals are separated successfully. And the separation experiments of mixed acoustic signals with multi-frequencies also show that the ICA with band-pass filters performs better than the conventional ICA in acoustic signals separation.

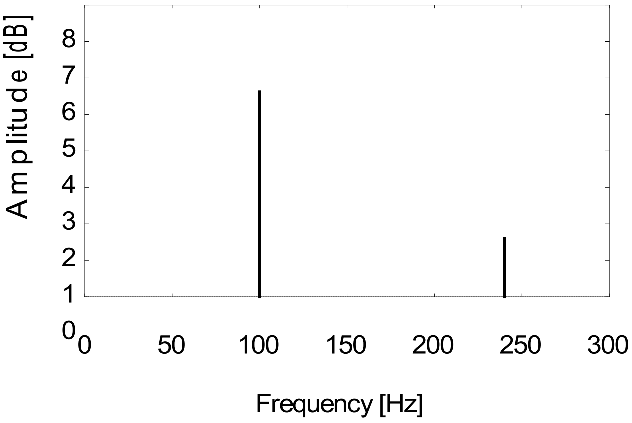


Fig. 8. Frequency characteristic of separated signal $\hat{s}_2(t)$ with band-pass filters.

TABLE I
SEPARATION RESULTS OF OBSERVED SIGNALS (UNIT: HZ) WHERE $f_1 = 20, f_2 = 140, f_4 = 240$.

f_3	30	40	50	60	70	80
Y or N	Y	Y	Y	N	Y	Y
f_3	90	100	110	120	130	150
Y or N	Y	N	Y	N	Y	Y
f_3	160	170	180	190	210	230
Y or N	Y	Y	N	Y	Y	Y

IV. EXPERIMENTAL RESULTS

According to the above simulation results, we separate the acoustic signals of motors and gears of mechanical devices by using the ICA with band-pass filters. The acoustic signals recording system is shown in Fig.5. Two microphones, which are held in different locations, are used to record the acoustic signals of operating mechanical devices. By applying the band-pass filters, we obtain the respective components of low-frequency, medium-frequency and high-frequency. Since only the medium-frequency components are the mixture of acoustic signals of gears and motors, we input the medium-frequency components to the ICA. Then the acoustic signals of gears and motors can be recovered by adding the low-frequency and high-frequency components to the separation results of the ICA, respectively.

An example of acoustic signals recorded by microphones L and R are shown in Figs. 10 and 11, respectively where the sampling rate is 8,000. Their frequency characteristics are shown in Figs. 12 and 13. Since the rotational speed of the motor is 3600 rpm and the rotor has 12 poles, the fundamental frequency of the motor is about 360 Hz. Similarly, since the gear ratio is 30:1, the fundamental frequency of the gear is about 12 Hz. Thus, it can be considered that the medium-frequency is the range of 300 to 2,000 Hz and the relevant band-pass filters are designed.

In Figs. 14 and 15, the medium-frequency fields of acoustic signals of left and right microphones with the band-pass filter are given respectively. The filtered signals are used as the input of the ICA. The spectra of the separated acoustic signals are

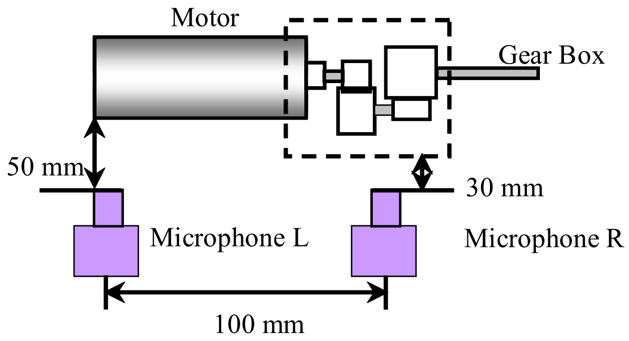


Fig. 9. The acoustic signals recording system.

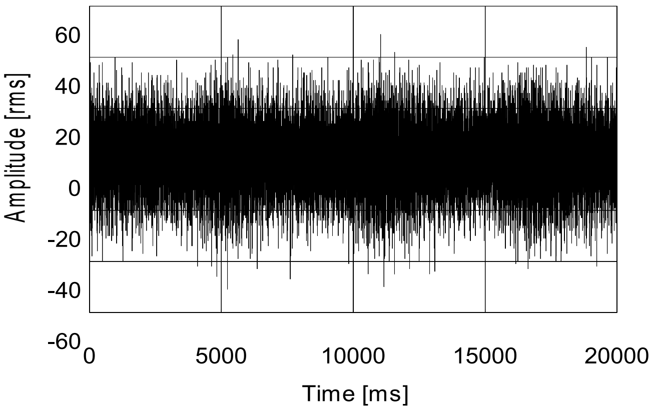


Fig. 10. Acoustic signal recorded by the left microphone.

shown in Figs. 16 and 17. Since a peak of amplitude nearby 1,000 Hz, which is about 3 times of the fundamental frequency of the motor, can be observed in Fig. 12, it is regarded that Figs. 16 and 17 show the medium-frequency fields of acoustic signals of the motor and the gear, respectively.

To verify the effectiveness of our proposed method, we also give the separation results by applying the recorded acoustic signals of mechanical devices to the ICA directly.

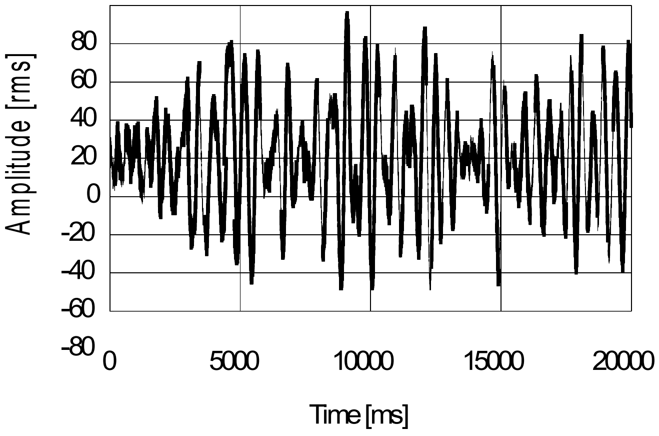


Fig. 11. Acoustic signal recorded by the right microphone.

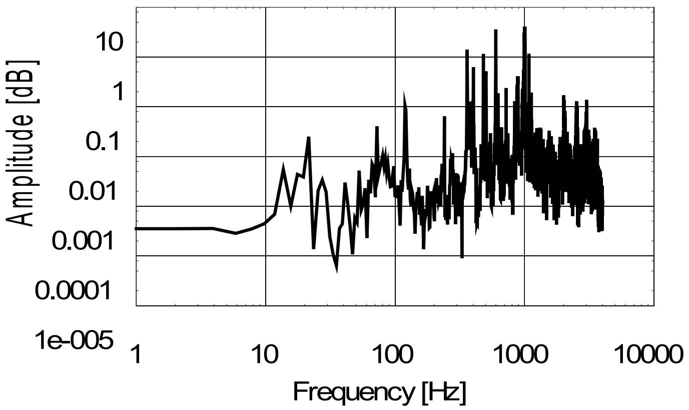


Fig. 12. Spectrum of acoustic signal of left microphone.

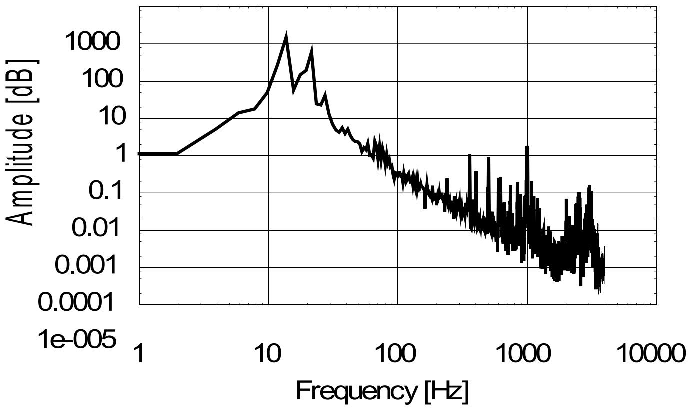


Fig. 13. Spectrum of acoustic signal of right microphone.

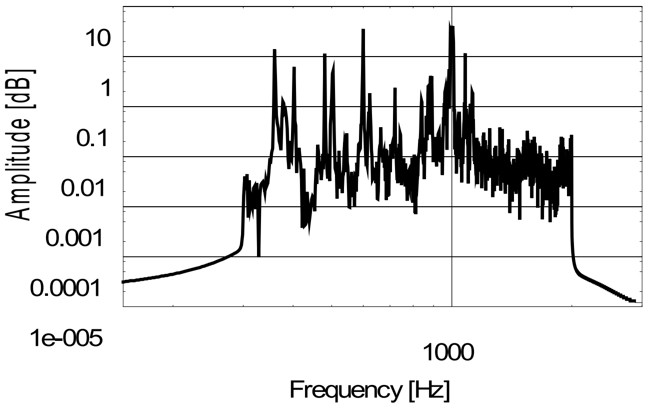


Fig. 14. Spectrum of Fig. 8 with a band-pass filter.

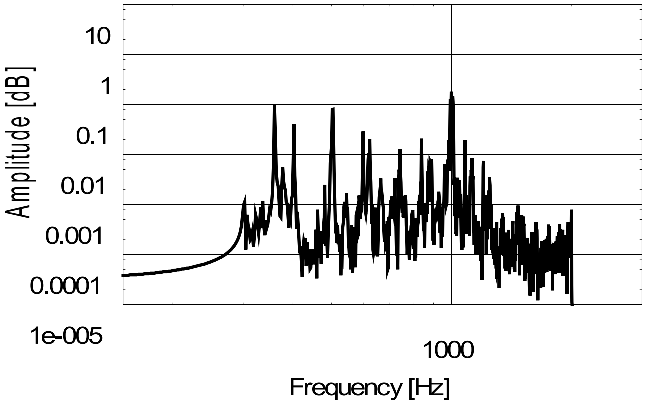


Fig. 15. Spectrum of Fig. 13 with a band-pass filter.

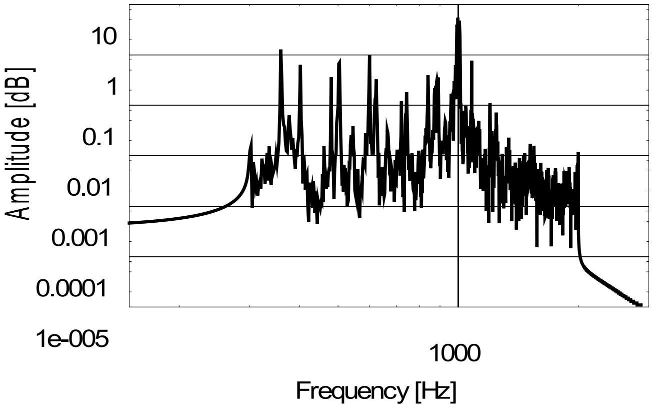


Fig. 16. Spectrum of the separated acoustic signal by using the ICA with a band-pass filter (motor).

The frequency characteristics of the separated acoustic signal are shown in Figs. 18 and 20, and the medium-frequency characteristics are shown in Figs. 19 and 21. Comparing with Figs. 12 and 13, it can be concluded that Figs. 18-20 show the frequency characteristics of the motor and the gear, respectively.

From the above figures, it can be seen that the ICA with band-pass filters performs better than the conventional ICA in acoustic signals separation. The spectrum of Fig. 19 is similar with the one of Fig. 21, especially the peaks of amplitudes appeared in both figures, which are located in the multiple of fundamental frequency of the motor, denote that the separation results of acoustic signals of the motor and the gear are not good.

The acoustic signals of the gear and the motor are recovered by adding the low-frequency and high- frequency components to the separation results of Figs. 16 and 17, respectively. The spectra of recovered acoustic signals of the gear and the motor are shown in Figs. 22 and 23 where the amplitudes of medium-frequency are adjusted according to the amplitudes of low-frequency and high-frequency, respectively. Comparing with the above figures, it can be concluded that the separation results are reasonable. The separated acoustic signals of the

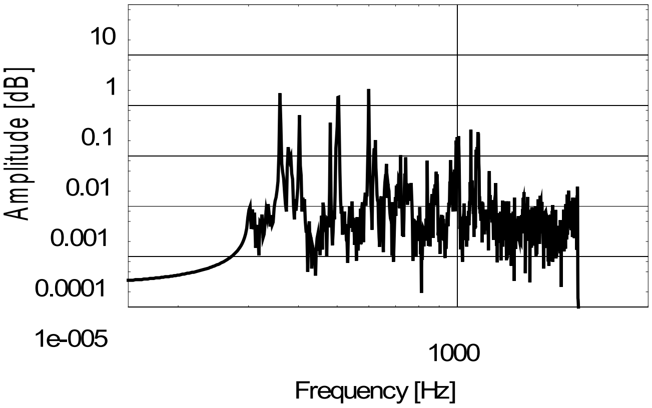


Fig. 17. Spectrum of the separated acoustic signal by using the ICA with a band filter-pass (gear).

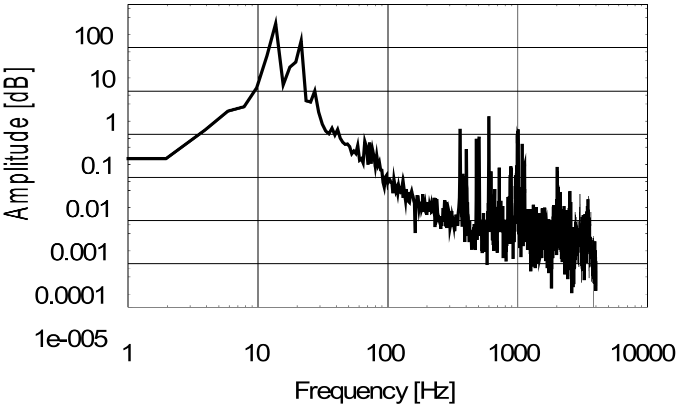


Fig. 18. Frequency characteristics of the separated acoustic signal by using the ICA (motor).

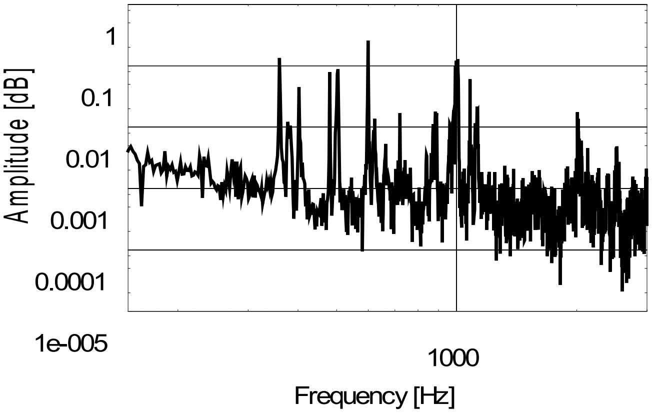


Fig. 19. Medium-frequency characteristics of the separated acoustic signal by using the ICA (motor).

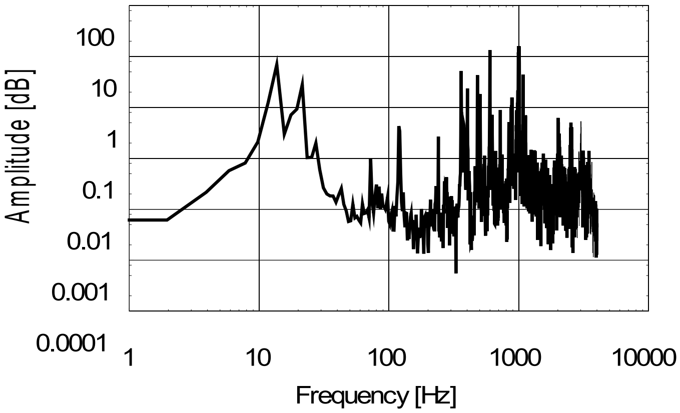


Fig. 20. Frequency characteristics of the separated acoustic signal by using the ICA (gear).

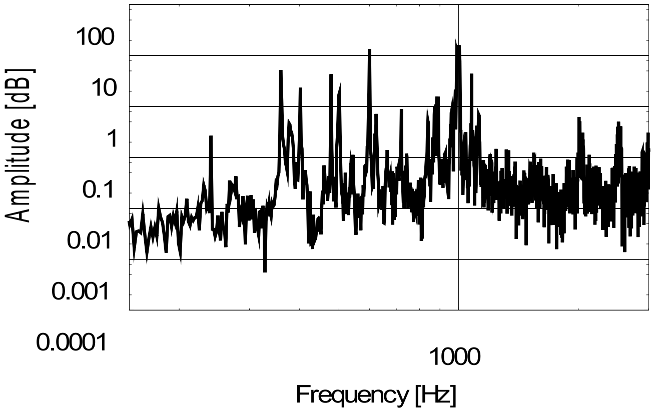


Fig. 21. Medium-frequency characteristics of the separated acoustic signal by using the ICA (gear).

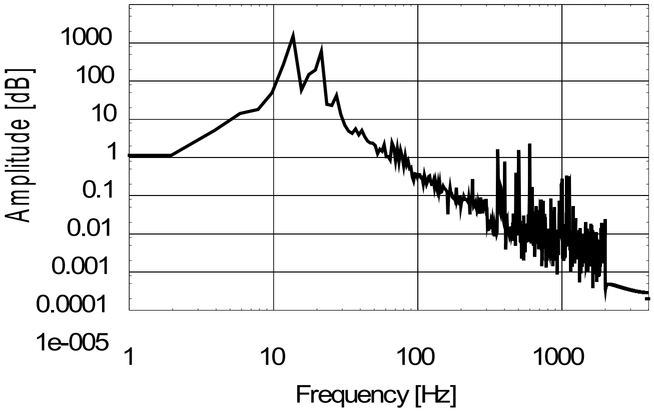


Fig. 22. Spectrum of recovered acoustic signal of the gear.

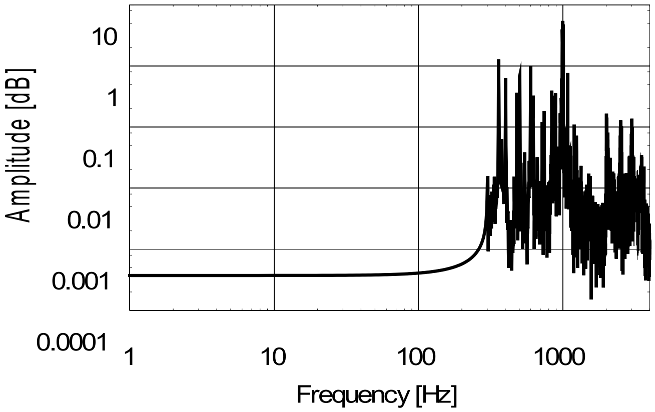


Fig. 23. Spectrum of recovered acoustic signal of the motor.

gear and the motor are also checked by a technician, the sounds of the motor and the gear denote that the acoustic signals of the gear and the motor are separated successfully by using the ICA with band-pass filters.

V. Conclusions

In this paper, a method of separating the acoustic signals of gears and motors of mechanical devices by using the ICA with band-pass filter is proposed. The simulation results denote that the mixed acoustic signals with less frequency components can achieve better separation performance by using the ICA. Therefore, for those independent signals which are mixed only in medium-frequency field, the ICA with band-pass filters can separate the independent original signals more accurately than the conventional ICA. Using the proposed method, we have solved the acoustic signals separation problem of gears and motors of mechanical devices successfully.

Acknowledgment

This work was supported by JSPS KAKENHI Grant-in-Aid for Scientific Research (B) (20360178). The authors would like to thank JSPS to support this research work.

References

[1] Hyvarinen A. , Karhumen J. , and Oja E. , Independent Component Analysis, Wiley Publisher, New York, (2001)

[2] Bingham E. , and Hyvarinen A. , A fast fixed-point algorithm for independent component analysis of complex valued signals, International Journal of Neural Systems, 10–1, 1–8 (2000)

[3] Hochreiter S. , and Schmidhuber J. , LOCOCODE performs nonlinear ICA without knowing the number of sources, Proceedings of the First International Workshop on ICA and Signal Separation, Aussois, France, 149–154 (1999)

[4] Lee T. W. , *Independent Component Analysis: Theory and Applications*, Kluwer Academic Publishers, Boston (1998)

[5] Cichocki A. , and Amari S. , *Adaptive Blind Signal and Image Processing: Learning Alogarithms and Applications*, Wiley Publisher, New York (2002)

A Taxonomy of the Future Internet Accounting Process

Igor Ruiz-Agundez, Yoseba K. Penya and Pablo Garcia Bringas
DeustoTech, Deusto Institute of Technology
University of Deusto
Bilbao, Basque Country
{*igor.ira,yoseba.penya,pablo.garcia.bringas*}*@deusto.es*

***Abstract*—Accounting is an old term that defines the activity of keeping records of the money. However, accounting in the Internet implies not only economic principles, but also engineering aspects. Accounting has been used for studying the impact on usage quotas, for dimensioning a provider infrastructure or for registering the data flow, among others. Each evolutionary step of the Internet has its implications in how the accounting process is performed. The new challenges of the Future Internet and the Next-Generation Networks (NGN) reveal the need of a revision of the accounting process. Against this background, we present a taxonomy of the accounting process of the Internet. This taxonomy classifies all the functions involved in accounting in a hierarchical structure, representing their behaviour. The resulting taxonomy helps defining the terminology, requirements and working framework of all the accounting-related studies. Further, it helps through the learning, teaching and assessing process in the area of accounting.**

***Keywords*-Accounting; Internet; Taxonomy.**

I. Introduction

Accounting is *"the art of recording, classifying, and summarizing in a significant manner and in terms of money, transactions and events which are, in part at least, of financial character, and interpreting the results thereof"* [1]. This activity is important for business and commercial purposes but also for the Internet, when a service provider charges a client with a fee for the usage of a certain service. These services are as varied as voice, data, multimedia, e-market places or any other emerging service.

In this scenario, the accounting process implies considering both engineering and economic aspects. It is essential to enforce policies such as usage quotas, to dimension the provider infrastructure, or to make statistical analysis of the usages, among others [2]. Resource accounting has also an important role in infrastructure congestion control due to the usage fees applied to the consumers [3].

Nowadays, we can distinguish two main accounting paradigms: the telecommunications world and the Internet world [4]. In telecommunications, the accounting process consist of apportioning the charges between the home environment, the serving network and the user. On the other hand, in the Internet, the accounting process is defined as the set of functions that manages the data regarding the use of the resources. In addition, traditionally, Internet accounting was limited to transport accounting. Customers paid for the use of the network resources of certain access providers [5] and not for any other type of service. Nevertheless and despite their differences, these worlds are converging, and more accurate definitions of the accounting process are emerging.

So far, in this converging model, accounting is understood as the process of collecting the resource usage for capacity and trend analysis, cost allocation, auditing and billing [6]. However, the evolution of the Internet denotes that this definition is not enough. It needs to refer to a broader concept, considering all the possible functions and related concepts [7]. This broader concept needs to meet the new requirements of the Future Internet, helping on accomplishing the new challenges that it implies.

Against this background, we introduce a new definition of the accounting process. This definition meets the requirements of the evolution of the Internet, understood as in the Future Internet [8] [9]. We also present a taxonomy of the full accounting process, from the resource usage to the financial clearing of its use.

Originally taxonomies were used only to classify organisms. Nowadays, taxonomies are used to classify things and concepts of any indole. There are economic, biological and even military taxonomies, each one specifying its domain area. Furthermore, taxonomies can have a tree, network or linear structure. All of them have proved to be useful for learning, teaching and assessing [10] and a central part of most conceptual models (ordering the elements into a model) [11]. They are specially useful presenting limited views of a model for human interpretation, and play an essential role in reuse and integration tasks.

The remainder of the paper is structured as follows. Section II introduces the related work on taxonomies about accounting. Section III presents an integrated vision of the accounting process describing all the involved functions. Finally, Section IV concludes and glimpses the future work.

II. Related work

Accounting of service usage is one of the main tasks of service providers in their operation and management processes, providing the necessary information for the subsequent functions. Although accounting requirements are studied in many books and articles, they rarely define a clear taxonomy of the full accounting process.

Some authors refer to the terms pricing, charging, or billing to represent the complete process of detecting the specific usage of a service [12] [13]. There is a need of disambiguating the accounting process employing a precise terminology and splitting clearly all the functions involved. In this paper, we refer to the accounting process as a meta-concept that includes all the aforementioned functions.

On the other hand, we considered the application area of each author. Different areas imply different terminology and semantics. For instance, accounting on packet-switched networks [14], micro-payments [5], grid services [15], mobile networks [16], VoIP services [17] or Wi-Fi connections [18].

Other researches tried to standardise the accounting process on the Internet [19] [20]. Nevertheless, to our knowledge, there is none that has performed a full taxonomy of the accounting process for the Future Internet accounting requirements making the learning, teaching and assessing process much harder [10].

The present taxonomy was performed following the directives to develop a taxonomy [21]. We started this method determining the requirements and identifying the concepts involved in the area of Internet accounting. After, a first draft of the taxonomy was deployed. This draft was reviewed with the users and the experts in the field providing the authors with feedback for a refining process. Once a final version was defined we started a maintaining process.

III. An integrated vision of the accounting process

The terminology regarding the billing process has always been diffuse sometimes involving contradictory semantics. The origin of this problem is not new and it dates back to the evolution of the accounting through the years and the influence of the different application areas in which it has been applied.

As terminology of the accounting process is evolving and is not standard [13], we studied the work done by other authors. Each contribution gives a different vision of the accounting process, creating a set of mixed concepts. However, after analysing the most relevant accounting process paradigms, we found out that they share some common characteristic that can be re-factored in order to have an integrated accounting process.

This integrated vision is represented in Fig. 1. The process starts with a resource usage which is registered by the metering function through the metering records. Afterwards, the mediation function intercedes by generating the accounting records for the accounting function. This function creates session records, which are sent both to the pricing and to the charging functions. The pricing function generates a formula defining how to price the session records that is used by the charging function. The flow continues with the charging, which generates charge records for the billing function. There, the final bill is sent to the financial clearing function.

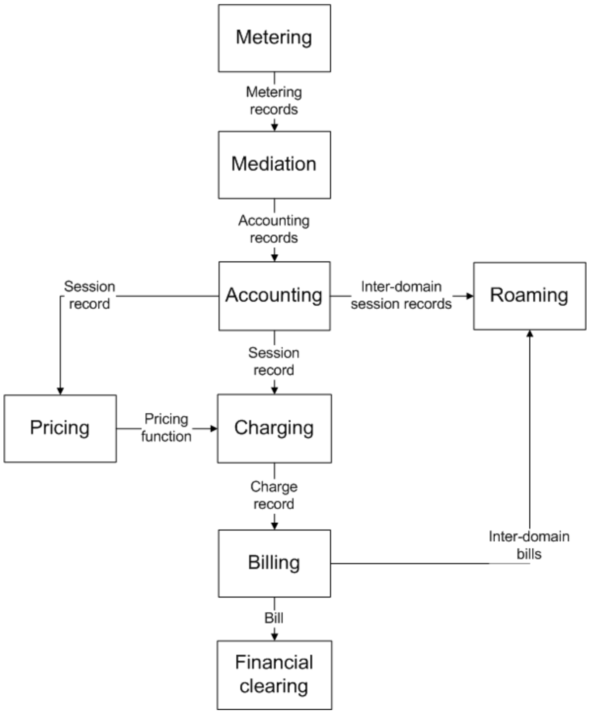


Figure 1. An integrated vision of the accounting process

Throughout the accounting function, inter-domain exchanges between organizations could be performed. These interchanges could happen at the accounting and billing steps, enabling roaming capabilities and inter-organization collaboration.

A. Metering

Metering is the function that collects the information flow regarding the resource usage of a certain service by a consumer and its usage. This measurement data is formed by service usage metrics provided by the monitoring function. Fig. 2 shows an overview of the metering function.

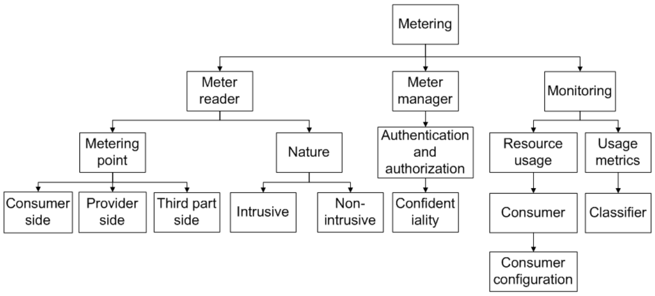


Figure 2. Metering function overview

This information is technical and is expressed in measurable quantities of consumer resources [5]. Examples of this measurable quantities are the number of data sent and received within an Internet connection, the seconds of a telephone call or the number of watts consumed.

This information is the starting point of the accounting process and will be used in the entire process. It determines the particular usage of resources within end-systems or intermediate systems on a technical level, including Quality-of-Service (QoS), management and networking parameters [22].

This function is normally implemented in a meter reader at a certain infrastructure point where the resource usage data is accumulated as long as the memory is able to. This point is known as metering point [23]. In addition, a meter reader can be classified as a consumer side meter, in which the meter reader is allocated with the consumer; or as a provider side meter, in which the meter reader is allocated in the providers infrastructure. In certain cases, meter readers are between the consumer and the provider, allocated in a third part or in a neutral infrastructure that both the consumer and the provider trust.

The meter readers can also be classified by their nature as intrusive (when there is an interface with the resource) or non-intrusive (when there is not an interface with the resource).

Additionally, the metering function is managed by a meter manager. There is a number of parameters that must be set for correct resource usage measurement. These parameters will depend on the resource itself. However, the general working procedure persists among the different resources. The manager is responsible for the authentication and authorization of the metering records, it must ensure the confidentiality of the data.

Monitoring is the function that collects the information of a resource usage as raw data and provides usage metrics to the metering function. The usage metrics reflect the use of a resource by a consumer (human, machine or other service) of a certain resource in measurable quantities. These metrics define the rules that the monitoring device apply in a classifier by defining the filtering of usage data [24].

The monitoring function can be conditioned by the consumers' configuration. That is, different consumers may have different usage metrics monitored (also known as data-points). The consumer configuration refers to the function that configures a service for its use. Normally, this configuration is set after the user is authenticated in the service provider infrastructure [25].

B. Mediation

The metering records generated by the metering function are usually stored in a homogeneous data format (accounting records). Fig. 3 shows an overview of the mediation function.

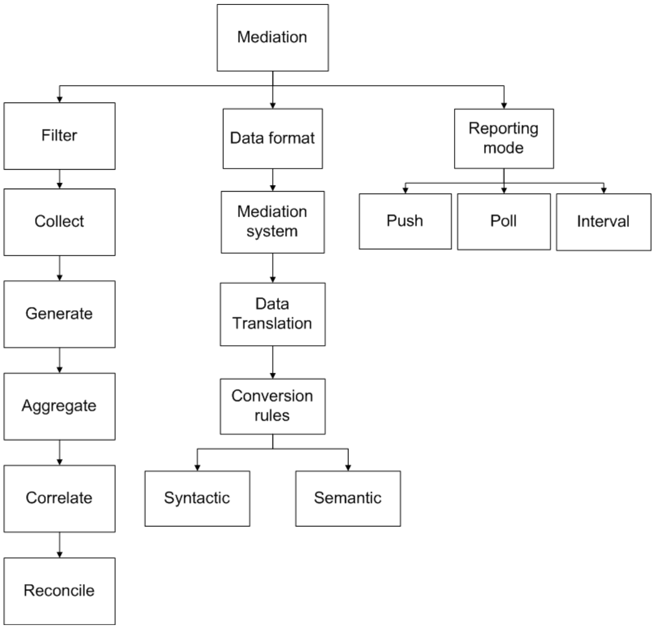


Figure 3. Mediation function overview

Mediation is intended to filter, collect, generate, aggregate, correlate, and reconcile raw technical data by transforming these metering records into a data format that can be used for storing and further processing [22] [26]. In this way, data processing is easier and the different functions of the accounting process require less mash-ups and conversions, resulting in a better performance [27].

In case different data formats are used, translation of data is necessary in order to have all the information in a homogeneous format as soon as possible. Conversion rules, both syntactic and semantic, are required in order to guarantee the integrity of the transformed data. This set of rules is also known as mediation systems [28] and is very common in the telecommunications world.

Further, the mediation can report to the accounting function in three different ways: push mode, poll mode or interval mode [19] [26]. In the push mode, the mediation function report the accounting function with accounting records as soon as it receives them. On the other hand, in the poll mode, the accounting function has to ask for the accounting records to the mediation function. Finally, in the interval mode the mediation function report to the accounting function each certain interval.

C. Accounting

Other taxonomies [6] include resource usage measurement, rating, charging, billing, and invoicing in this function. Nevertheless, we decided to split these functions in order to have a more representative organization. We also need to stress the difference between the accounting process, which implies the full process described in Fig. 1, and the accounting function, which we are defining now. Fig. 4 shows an overview of the accounting function.

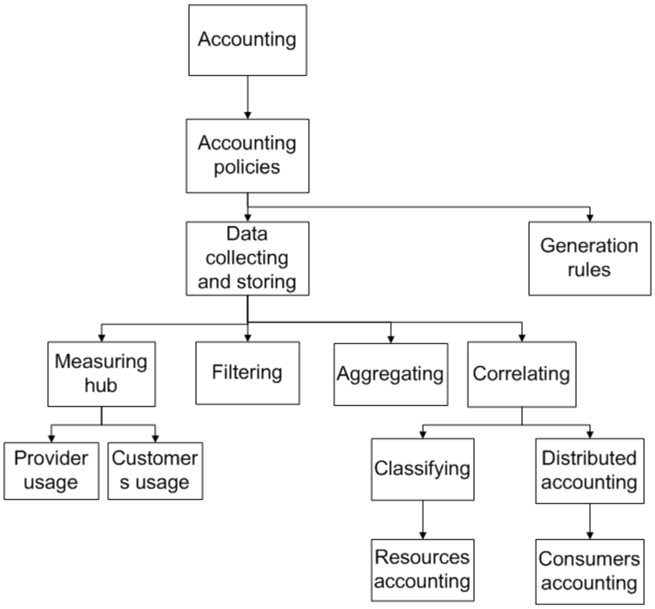


Figure 4. Accounting function overview

Accounting is the process of filtering, collecting and aggregating the information that reflects a resource usage by a certain consumer. This process will generate session records whose format will depend on the service infrastructure and the service provider [27]. The session records represent the resource usage over a session. Accounting gateways creating the session records may do so by processing interim accounting events or accounting events from several devices serving the same user [6].

This accounting function is expressed in metered resource consumption, e.g., for applications, calls, or any type of connections [22], depending on the service provided, by representing the technical specifications of the service. It includes the supervision of the data gathering from the mediation function, the collection and the storage of this data [4]. Accounting policies define how these functions behave and are specified by a set of generation rules [25].

The accounting data collection and storing is also known as archival accounting and is performed at a measuring hub. This function transports the metered data to a storage point or measuring hub [12]. The measuring hub is the point where the data from the metering readers is collected. It is also known as storage point. The measuring hub collects data from two main sources: the provider and the customer. Data from the provider is created by internal and control meters, and is used to control the provider's infrastructure. On the other hand, data from the customer represents the usage consume and is used in the whole accounting process.

Data archival may be necessary because of the memory limitations of the meter readers or because the information may be needed for long periods of time. It is also used to reconstruct missing entries, to prevent data loss and to archive the data for long periods of time. Legal or financial requirements frequently mandate archival accounting practices, and may often dictate that data to be kept confidential, regardless of whether it is to be used for billing purposes or not [6].

The concentration of the metering results in a measuring hub may be necessary to correlate information from distributed meter readers and to process the data solely in one point. The correlation are based on classifying functions that group the accounting records by resources. All the available resource accounts are stored by the correlating function, which can also group the accounting records by grouping the data from a distributed accounting organizing the data from the different consumers.

D. Roaming

Roaming is the function that allows using more than one provider while maintaining a formal, customer-vendor relationship just with one [29]. In order to offer roaming capabilities, providers need three main subsystems. The consumers' subsystem, which registers visiting consumers, the authentication subsystem, which validates the credential of the consumer, and the accounting subsystem, that has already been described [30].

In order to allow consumers to roam, providers need roaming agreements between them. They negotiate the legal aspects of authentication, authorization and billing of the visiting subscriber. There are several standards that create a work field framework for these agreements [31] [20].

Roaming can also be intra-domain or inter-domain [6]. Being intra-domain implies that there is an exchange of session records between different accounting functions but always in the same provider or administrative boundary. On the other hand, in the inter-domain roaming the session records travel from one provider to another, crossing their administrative boundaries.

E. Pricing

Pricing is the function of giving a price to a certain resource usage. It is a critical function for the full accounting process because it defines the price that a basic quantity of the service will cost. Some authors name it a rating or pricing policy [6]. This pricing policy determines the way a session record is rated. These records come from the accounting functions and are correlated to the price that is normally represented in monetary units and depends on the pricing scheme used. Fig. 5 shows an overview of the pricing function.

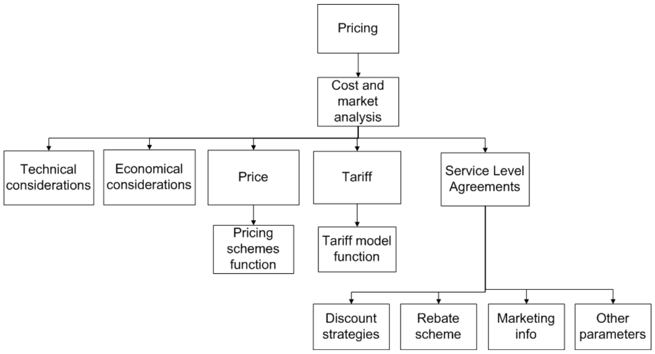


Figure 5. Pricing function overview

This process may combine technical considerations, such as resource consumption, and economical ones, such as applying tariffing theory or marketing methods [22]. The price can be calculated in many different ways (e.g. auctions, static pricing, dynamic pricing, priority pricing, cost-volume-profit analysis scheme or based on market situation analysis) [32] [33] [13]. However, it will always reflect the results of cost and market analysis. This function translates the previous economic considerations into technical quantities that can be merged with the measurable quantities of consumer resource usages.

Pricing is defined by the pricing schemes, which are a critical part of the business and are related to cost and market analysis. It is a function for calculating a price. It can be represented as a formula (pricing function) consisting of the pricing variables (consumption measure metrics of the session records) and pricing coefficients [25]. Pricing schemes can be based on many different paradigms, such as pre-paid, post-paid, time-based, volume-based, flat-rate, usage-based or location-based, among others.

Tariffs are a special case of pricing. They are normally regulated by a governmental institution and imply political economic impacts. They have been applied to the traditional telephone network, energy or gas markets. The tariffs are defined by the tariff models or functions. They determine the tariff function for a resource usage.

These functions, pricing functions or tariff functions, are applied in the charging function. They can be modified by discount strategies, rebate schemes, marketing information or any other parameter defined by the service level agreements.

F. Charging

Charging is the process of calculating the cost of a resource usage, the function that translates technical values into monetary units by applying a pricing function to the session records [22]. It correlates session records, from the accounting function, and resource usage unit price to generate charge records [27] [4].

Charging acts as an umbrella term for charging options and charging mechanisms. This separation emphasises both the technical and the economic aspects of charging [15]. Some authors refer to charging as billing. Nevertheless, as we will see later on, billing implies some different processes, such as customers' data management [22].

These charge records are formed by the technical quantities of a resource usage and their corresponding monetary units. The records can be used for multiple purposes of business intelligence: statistical analysis, data mining, auditing, revenue estimation, financial planning or structure dimensioning.

The charging policies define when and how the billing function is invoked. They define the frequency of cost allocation every time accounting data is received, at regular intervals of time (e.g. daily, each moth or each two moths) or when requested by the charging function. They also define the granularity of the billing function. Granularity is defined as how sub-divided a data field is. For example, a postal address can be recorded, with low granularity, as a single field (address) or with high granularity, as multiple fields (street address, city, postal code, country).

The charging can be distributed between multiple parties as defined in the distribution policy. This policy will split the costs between the different parties or consumers, allocating an already-known cost among several entities [6]. Each party has its own profile that could contain the client pricing function, discounts or special offers.

The consumers can also have different business relationships with the providers. This relation will define the charging mode (e.g. subscribers or pay-per-use).

G. Billing

Billing, or invoicing, is the process of transforming charge records into the final bill, or invoice, summarizing the charge records of a certain time period (usually a month) and indicating the amount of monetary units to be paid by the customer [4]. Fig. 6 shows an overview of the billing function.

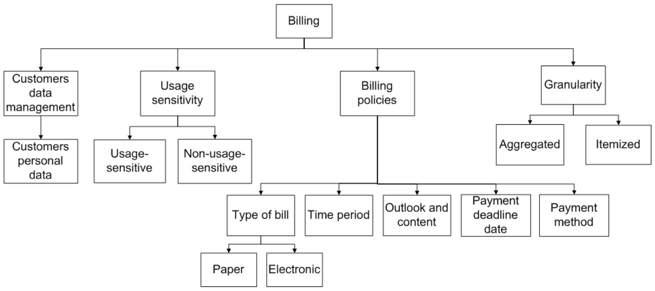


Figure 6. Billing function overview

It may include information about the customer that is gathered from the customers data management system. This system contains all the customers' personal data. The billing function has also usage-sensitivity when depends on the resource usage of the consumers. On the other hand, a process that is not affected by the resource usage is non-usage-sensitive [6].

There are also billing policies that define the type of bill (paper or electronic), the time period that the bill represents, the outlook and content, the payment deadline date and how the financial clearing is done, specifying the payment method [27].

As charging, billing can also have different granularity. An aggregated bill represents two or more charges together and an itemised bill has all the charges individualised.

H. Financial clearing

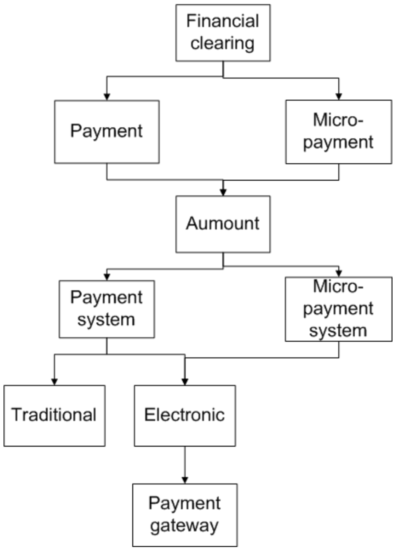


Figure 7. Financial clearing function overview

The financial clearing function includes activities from a commitment for a transaction to its settlement. In the case of resource accounting, this function implies the payment of a bill. Payment is the function of transferring the money of the client to the service provider. The amount to transfer is defined by the bill. Fig. 7 shows an overview of the financial clearing function.

The payment function will use a traditional or electronic payment system. Cash, paper checks and automatic bank clearances are in the group of the traditional payment systems. On the other hand, credit card systems are grouped in the electronic payment systems. This payment function will have a well-defined scheme specifying the way the money is exchanged between all the participants [27] through a payment gateway.

There is also a special type of payment, the micro-payments. These payments have requirements of high speed processing, delivery occurs immediately and in small sums of money. These payments use a specific micro-payment system, that are mainly electronic. A payment system also supports money transfers, which are smaller than the minimal economically feasible credit card payment [5].

IV. Conclusion

In this paper, we introduce a taxonomy of the accounting process. We have detailed all the functions involved in it, looking at all the relationships between them. We believe that the presented taxonomy contributes to the learning, training and assessing in the area of accounting because it gives an integrated vision of the process. As it defines a common vocabulary, it is also useful for the definition of the accounting requirements among different actors.

This taxonomy was defined using a developing method, as described in Section II, ensuring its quality and the maintainability of the knowledge to potential changes.

Further, we proposed an unified and controlled vocabulary that can be use in any operation related with the Internet accounting. The taxonomies have been proved to help to organize content and make connections between people and the information they need [21].

Future works include performing a proof-of-concept of the presented taxonomy by implementing it in an accounting system. Furthermore, this taxonomy defines the pillars for the developing of accounting related applications such as fraud management systems [34] or data profiling [35] among others. We also planned a validation of the proposed taxonomy, using both direct inspection and validation metrics [36] as well as updating the taxonomy itself if there are substantial changes in the area of Internet accounting.

References

[1] R. Singh Wahla, "AICPA committee on terminology," *Accounting Terminology Bulletin*, vol. 1, p. Review and Rsum, 1941.

[2] V. Agarwal, N. Karnik, and A. Kumar, "Metering and accounting for composite e-Services," in *Proc. 1st IEEE Intl Conf. on E-Commerce*, 2003, pp. 35–39.

[3] N. Blefari-Melazzi, D. D. Sorte, and G. Reali, "Accounting and pricing: a forecast of the scenario of the next generation internet," *Computer Communications*, vol. 26, no. 18, pp. 2037 – 2051, 2003.

[4] M. Koutsopoulou, A. Kaloxylos, A. Alonistioti, L. Merakos, and K. Kawamura, "Charging, accounting and billing management schemes in mobile telecommunication networks and the internet," *IEEE Communications Surveys*, vol. 6, no. 1, pp. 50–58, 2004.

[5] R. Prhonyi, "Micro payment gateways," Ph.D. dissertation, Twente University, 2005.

[6] B. Aboba, J. Arkko, and D. Harrington, "RFC2975: Introduction to Accounting Management," *RFC Editor United States*, 2000.

ISBN: 978-1-61208-101-4

[7] A. Pras, B. van Beijnum, R. Sprenkels, and R. Parhonyi, "Internet accounting," *IEEE Communications Magazine*, vol. 39, no. 5, pp. 108–113, 2001.

[8] S. Shenker, "Fundamental design issues for the future Internet," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, 1995.

[9] A. Gavras, A. Karila, S. Fdida, M. May, and M. Potts, "Future internet research and experimentation: the FIRE initiative," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 3, p. 92, 2007.

[10] P. W. Airasian, K. A. Cruikshank, R. E. Mayer, P. R. Pintrich, and J. R. M. C. Wittrock, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*, abridged edition ed., L. W. Andersonand and D. R. Krathwohl, Eds. Allyn & Bacon, 2000.

[11] C. Welty and N. Guarino, "Supporting ontological analysis of taxonomic relationships," *Data and knowledge engineering*, vol. 39, no. 1, pp. 51–74, 2001.

[12] B. Stiller, G. Fankhauser, B. Plattner, and N. Weiler, "Prestudy on Customer Care, Accounting, Charging, Billing, and Pricing," *Computer Engineering and Networks Laboratory TIK, ETH Zurich, Switzerland, Pre-study performed for the Swiss National Science Foundation within the Competence Network for Applied Research in Electronic Commerce*, 1998.

[13] M. Kouadio and U. Pooch, "A taxonomy and design considerations for Internet accounting," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 5, p. 48, 2002.

[14] M. Karsten, J. Schmitt, B. Stiller, and L. Wolf, "Charging for packet-switched network communication - motivation and overview," *Computer Communications*, vol. 23, pp. 290–302, 2000.

[15] C. Morariu, M. Waldburger, and B. Stiller, "An integrated accounting and charging architecture for mobile grids," in *Broadband Communications, Networks and Systems, 2006. BROADNETS 2006. 3rd International Conference on*, Oct. 2006, pp. 1–10.

[16] M. Koutsopoulou, A. Kaloxylos, and A. Alonistioti, "Charging, accounting and billing as a sophisticated and reconfigurable discrete service for next generation mobile networks," in *IEEE Vehicular Technologhy Conference*, vol. 4. Citeseer, 2002, pp. 2342–2345.

[17] L. Deri, "Open source VoIP traffic monitoring," *SANE 2006*, 2006.

[18] G. Detal, D. Leroy, and O. Bonaventure, "An adaptive three-party accounting protocol," in *Proceedings of the 5th international student workshop on Emerging networking experiments and technologies*. ACM, 2009, pp. 3–4.

[19] C. Mills, D. Hirsh, and G. Ruth, "RFC1272: Internet Accounting: Background," *RFC Editor United States*, 1991.

[20] I. Programme, *IPDR Service Specification. Design Guide*, http://tmforum.org/ ed., TeleManagement Forum, September 2008.

[21] M. Whittaker and K. Breininger, "Taxonomy development for knowledge management," in *World Library and Information Congress: 74th IFLA General Conference and Council*, August 2008.

[22] B. Stiller, J. Gerke, P. Reichl, and P. Flury, "Management of differentiated services usage by the cumulus pricing scheme and a generic internet charging system," in *Proceedings of the Symposium on Integrated Network Management*. Citeseer, 2001.

[23] N. Brownlee, C. Mills, and G. Ruth, "RFC2722: Traffic flow measurement: architecture," *RFC Editor United States*, 1999.

[24] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "RFC2475: An Architecture for Differentiated Service," *RFC Editor United States*, 1998.

[25] T. Zseby, S. Zander, and C. Carle, "RFC3334: Policy-Based Accounting," *Internet RFCs*, 2002.

[26] I. Programme, *IPDR Business Solution Requirements*, http://tmforum.org/ ed., TeleManagement Forum, May 2009.

[27] B. Stiller, G. Fankhauser, B. Plattner, and N. Weiler, "Charging and accounting for integrated internet services - state of the art, problems, and trends," in *Problems, and Trends; The Internet Summit (INET98)*, 1998, pp. 21–24.

[28] G. Zhang, B. Reuther, and P. Mueller, "User Oriented IP Accounting in multi-user systems," in *Integrated network management VIII: managing it all: IFIP/IEEE Eighth International Symposium on Integrated Network Management (IM 2003), March 24-28, 2003, Colorado Springs, USA*. Kluwer Academic Pub, 2003, p. 59.

[29] B. Aboba, J. Lu, J. Alsop, J. Ding, and W. Wang, "RFC2194: Review of Roaming Implementations," *RFC Editor United States*, 1997.

[30] B. Aboba and G. Zorn, "RFC2477: Criteria for Evaluating Roaming Protocols," *RFC Editor United States*, 1999.

[31] T. Hoc, "On the relaying capability of Next-Generation GSM cellular networks," *IEEE Personal Communications*, p. 41, 2001.

[32] M. Karsten, J. Schmitt, L. Wolf, and R. Steinmetz, "Cost and price calculation for internet integrated services," in *In Proceedings of Kommunikation in Verteilten Systemen (KiVS99*. Springer, 1999, pp. 46–57.

[33] X. Chang and D. Petr, "A survey of pricing for integrated service networks," *Computer communications*, vol. 24, no. 18, pp. 1808–1818, 2001.

[34] I. Ruiz-Agundez, Y. K. Penya, and P. G. Bringas, "Fraud detection for voice over ip services on next-generation networks," in *Proceedings of the 4th Workshop in Information Security Theory and Practices (WISTP 2010)*. Passau, Germany: Springer, 12-14 April 2010.

[35] S. Hung, D. Yen, and H. Wang, "Applying data mining to telecom churn management," *Expert Systems with Applications*, vol. 31, no. 3, pp. 515–524, 2006.

[36] S. Spangler and J. Kreulen, "Interactive methods for taxonomy editing and validation," in *Proceedings of the eleventh international conference on Information and knowledge management*. ACM, 2002, pp. 665–668.

Development and Performance Evaluation of PSO based Single Layer Nonlinear ANN Classifiers of Indian Online Shoppers

Ritanjali Majhi
School of Management
National Institute of Technology, Warangal, India
e-mail:ritanjalimajhi@gmail.com

Bijayalaxmi Panda
Dept. of CSE
ITER, S 'O'A University, Bhubaneswar, India
e-mail: bijaya_003@gmail.com

Babita Majhi
Dept. of IT
ITER, S 'O'A University Bhubaneswar, India
e-mail: babita.majhi@gmail.com

T. Rahul
School of Management
National Institute of Technology, Warangal, India
e-mail:rahulkitss@gmail.com

Ganapati Panda
School of Electrical Sciences
Indian Institute of Technology Bhubaneswar
e-mail : ganapati.panda@gmail.com

Abstract— **In this paper an in depth study is made on identifying primary factors on which the Indian online shoppers are influenced. Based on the dominating factors extracted from the internet shoppers' replies they are clustered. A novel and efficient single layer nonlinear ANN model with PSO based weight adaptation (SNANNP) is developed by taking the demographic input of the shopper and clustering results as the desired class. Close observation of experimental results indicate that the proposed method exhibits superior classification performance compared to that obtained by discriminant analysis and conventional single layer ANN (SNANN) model.**

Keywords-Single layer nonlinear ANN (SNANN); Consumer Classification; Online Shopping; Factor Analysis; Discriminant Analysis and Particle swarm optimization(PSO) based training

I. INTRODUCTION

In recent years, internet marketing has gained popularity. The competition among the retailers have increased due to introduction of several online services. The literature survey reveals that nearly 67% shoppers use internet and half of them do shopping online. The advantages for choosing online shopping are: reduction in the cost by employing less manpower, saving of shopping time, less fear of loss of money in online shopping, increase in the bargaining power of shoppers, increase of rivalry among the competitors and improvement in security and ease of delivery.

Gradually the interest towards online shopping is increasing during last few years. The consumers mostly prefer online services to purchase air/railway tickets, movie tickets, consumer electronic goods, audio/video files, software packages like an operating system, web browser, audio/video player etc. Usually the Indian consumers discuss with their friends and relatives before buying a product.

The neural network has been used to classify consumers in choosing hospitals and marketing implications [1]. This model is also useful for market strategy planning. In another paper [2], a new approach has been proposed for direct marketers targeting potential consumers new to their categories. It is reported that the ANN model provides improved classification accuracy. A recent study [3] deals with conjoint analysis to create health plans that optimize value for consumers.

A systematic study is made in this paper to group the consumers based upon their online shopping behaviour. The choice of people to prefer a service is diverse. A questionnaire was made on internet shopping based upon which 14 variables pertaining to the theme are chosen. Considering the correlation between the variables these are reduced into a number of factors. Using loading of the factors internet shoppers are grouped into clusters. Each cluster represents group of consumers with similar behaviour.

The consumers are then identified into number of classes depending on the demographic attributes such as age, gender, level of education, amount spent on shopping and amount spent on online shopping. The results of the cluster analysis is used as the training class of the classification model. Three types of classification methods such as discriminant analysis, single layer nonlinear artificial neural network(SNANN) and single layer ANN with PSO based training are used. The rest of the paper is organized as follows :

Section II develops a unique intelligent single layer nonlinear ANN model used for consumer classification. The basic of particle swarm optimization algorithm is outlined in Section III. Section IV deals with issues relating to data collection and data reduction using factor analysis. Cluster analysis for consumer grouping is dealt within Section V. The classification operation using discriminate analysis, SNANN and SNANNP models is carried out in Section VI. The results obtained from simulation studies are also presented and discussed in the same section. Finally Section

VII deals with the conclusion of the study.

II. SINGLE LAYER NONLINEAR ANN(SNANN) CLASSIFIER

The SNANN possesses the simplicity of single layer ANN and performance capability multilayer ANN. Hence it is chosen here to develop an efficient adaptive classifier. The enhanced performance is achieved by introducing nonlinearity through trigonometric functional expansion [6]. The block diagram of an SNANN structure is shown in Fig.1. Let the input signal vector be represented as

$$\underline{X}(k)=[x(k)\ x(k-1)....................x(k-m+1)]^T \quad (1)$$

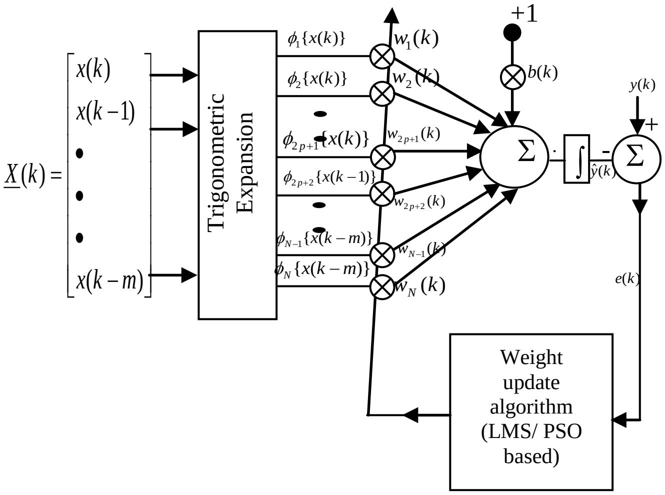


Fig. 1 Proposed block diagram of SNANN/SNANNP model with or without
PSO based training

which is the demographic input of an online shopper. Then the functional expansion (FE) block maps each element $x(k)$ into (2p +1) nonlinearly expanded independent components. For trigonometric expansion $\underline{\phi}[x(k)]$ is given by

$$\underline{\phi}[x(k)]=[x(k),\cos\{\pi x(k)\},\sin\{\pi x(k)\}.......$$
$$\cos\{p\pi x(k)\},\sin\{p\pi x(k)\}]^T\ = \quad (2)$$
$$[\phi_1\{x(k)\},\phi_2\{x(k)\},......\phi_{2p+1}\{x(k)\}]^T$$

where p is an integer. When each element of $\underline{X}(k)$ is expanded then the expanded vector is represented as

$$\underline{\phi}[\underline{X}(k)=[\phi_1\{x(k)\}.......\phi_{2p+1}\{x(k)\},\phi_{2p+2}\{x(k-1)\},.....$$
$$\phi_{2(2p+1)}\{x(k-1)\}..........\phi_N\{x(k-m)\}]^T \quad (3)$$

where $N=m(2p+1)$ and m is the number of signal samples fed into the SNANN. The output $y\hat{}(k)$ of Fig. 1 is then given by

$$\hat{y}(k)=\underline{f}\{\phi^T(\underline{X}(k))\underline{W}(k)+b(k)\} \quad (4)$$

where $b(k)$ is the bias weight and f {.} denotes tanh function. $W(k)$ represents the weight vector given by

$$\underline{W}(k)=[w_1(k),w_2(k)..........w_N(k)]^T \quad (5)$$

The weights of the SNANN model are trained using the algorithm

$$\underline{W}(k+1)=W(k)+\mu\underline{\phi}\{\underline{X}(k)\}.e(k)(1-\hat{y}^2(k)) \quad (6)$$

where the error term is given by

$$e(k)=y(k)-y\hat{}(k) \quad (7)$$

The convergence coefficient is represented by μ and its value lies between 0 and 1. Equations (2), (3), (4) and (7) represent the key equations of SNANN algorithm.

III PARTICLE SWARM OPTIMIZATION (PSO)FOR TRAINING OF CLASSIFIER WEIGHTS

The PSO is a simple but efficient population based swarm intelligence based optimization algorithm. The PSO algorithm emulates swarm behavior and each particle represents a point in the D-dimensional solution space. The swarm, a collection of particles, initially contains a population of random solutions. Each particle is given a random velocity to fly through the problem space. Each particle keeps track of its previous best position, called *pbest* and its fitness value. Each swarm remembers its best position called *gbest* . The velocity $v_i(d)$ and the position $x_i(d)$ of the d th dimension of ith particle are adapted [4]-[5] according to

$$v_i(d)=wv_i(d)+c_1*rand1_i*(p_i(d)-x_i(d))+ \\ c_2*rand2_i(d)*(p_g(d)-x_i(d)) \quad (8)$$

$$x_i(d)=x_i(d)+v_i(d) \quad (9)$$

where $p_g(d)$ and $p_i(d)$ are the d th dimensional positions corresponding to the *gbest* and *pbest* respectively, $rand1_i(d)$ and $rand2_i(d)$ represent random numbers in the range [0, 1] and are different in different dimensions, c_1 and c_2 are acceleration coefficients and $w$is the inertial weight which plays the crucial role of balancing between the global search and local search.

IV DATA COLLECTION AND FEATURE REDUCTION

(a)Data collection

The data is collected through a questionnaire from Indian online shoppers. Each answer is weighted in a 5 point-scale in which '1' indicates "strongly disagree" and '5' indicates "strongly agree".

The questionnaire comprises of 5 sections. First section contains 5 questions. In sections 2, 3 and 4, 14 variables are used. The last section deals with the questions related to the demography of the consumers. Demographic attributes include age, gender, education, average amount spent on shopping and average amount spent on online shopping. A total of 163 complete data are collected each having 14 variables excluding the demographic variables. The 14 variables used are listed in Table 1.

Table 1
List of variables affecting behavior of internet shoppers

Information Design (ID)	Internal Norm (IN)
Visual Design (VD)	Convenience(C)
Navigation Design (ND)	Online Innovativeness(OI)
Communication (Comm)	Enjoyment (E)
Self Efficacy (SE)	External Norm (EN)
Privacy (P)	Security (S)
Vender repute (VR)	Social Presence (SP)

(b)Data reduction using factor analysis

Factor Analysis [7] is a statistical tool to find interrelationships between the variables. This is a technique through which the prominent factors hidden in the data are extracted. This is essentially a data reduction method used to reduce a set of observed variables to a set of latent variables.

This analysis is an ideal method for extracting factors. There are several methods to decide how many factors has to be extracted. The most widely used method for determining the number of principal factors is based on eigen value consideration [8]. In the present study five factors influencing consumer behavior are obtained using factor analysis. The eigen values, percentage of variance and cumulative percentage of variance of five prominent variables obtained from factor analysis are listed in Table 2. The factor loadings obtained are provided in Table 3.

V. CLUSTER ANALYSIS

Clustering mainly groups unlabeled datasets. It is an unsupervised classification technique. Mainly two types of clustering techniques are used : hierarchical and partitioning/ non-hierarchical [9]. Each one has its own merits/demerits depending on the applications in which it is used. In this study hierarchical clustering is chosen because it is suitable when the number of possible clusters are not known apriori. Cosine distance is used as the similarity measure to evaluate the closeness among the data points. The dendrogram of the factor scores computed is shown in Fig.2.

The dendrogram shows that the entire data are grouped into three clusters. It is found that 41, 67 and 55 online shoppers belong to clusters 1, 2 and 3 respectively. The factor scores with major contribution to each cluster are shown in Table 4. The demographic variables of the consumers in each cluster are listed in Table 5.

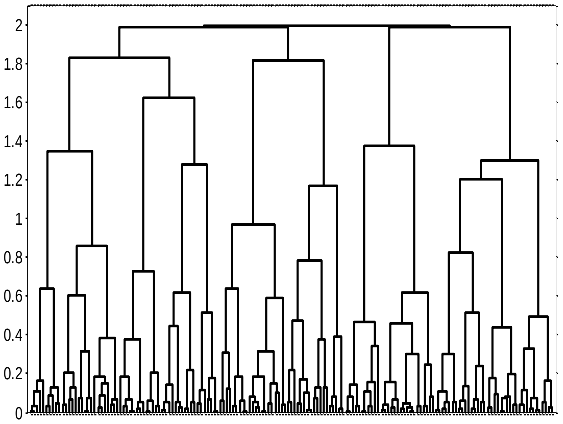


Fig. 2 Dendrogram of the factor scores using hierarchical clustering

Table 2
Eigen values, percentage of variance and cumulative percentage of variance of variables obtained from factor analysis

Eigen values	% of variance	Cumulative % of variance
4.4286	31.6330	31.6330
1.8162	12.9728	44.6057
1.6309	11.6494	56.2551
1.1081	7.9153	64.1704
1.0542	7.5303	71.7007

VI. CONSUMER CLASSIFICATION

Clustered online shoppers need to be classified. The classification is based upon their behavior towards online shopping. The demographic attributes of the consumers are used as the inputs to the classifier. The results obtained from the cluster analysis are used as the training class for the classifier. The SNANN, SNANNP as well as discriminate classifiers are employed to classify these consumers.

(a)Discriminant analysis based classification

Discriminant analysis is employed to classify the objects into various groups. This is a supervised classification technique [10, 11] in which the dependent variable is the class/group and the independent variables are the features/attributes. In this, a number of discriminate

Table 3
Factor loading of different variables under different factors

Variables	Factor1	Factor2	Factor3	Factor4	Factor5
ID	0.2130	**0.4800**	0.4023	0.0602	-.0392
VD	0.1677	**0.8720**	0.0570	0.1575	-.0754
ND	0.1703	**0.7237**	0.1424	0.2232	0.0167
Comm	0.0640	0.1671	0.0258	**0.8239**	0.0458
SP	0.0789	0.1102	0.0618	**0.6818**	0.0351
SE	0.3715	0.3907	**0.4150**	-.1339	0.0158
Privacy	**0.8460**	0.1897	0.1807	0.0597	-.0419
Security	**0.7416**	0.1726	0.2853	0.0779	0.0043
VR	**0.5191**	0.2751	0.2846	0.2947	0.0616
EN	-.1596	-.1216	0.0916	0.2442	**0.6016**
IN	0.1571	0.0578	0.0655	-.1265	**0.9730**
Enjoyment	0.1318	0.1939	**0.7547**	0.2457	0.0246
Convenience	0.3304	0.0320	**0.6090**	-.1239	0.0122
Online innovativeness	0.0650	0.0423	**0.3019**	0.0332	0.0716

Table 4
Factor scores under different clusters

Factors	Cluster 1	Cluster 2	Cluster 3
Factor 1	-0.3974	**0.3371**	-0.1145
Factor 2	**0.2489**	0.1696	-0.3922
Factor 3	-0.9219	0.2284	**0.4090**
Factor 4	-0.0025	-0.6127	**0.7483**
Factor 5	-0.7892	0.1706	**0.3804**

functions are formed which is equal to the number of classes in the data. These functions represent the classification rules for each class. The actual data is applied to each function and the corresponding output is obtained. The data belongs to a group that provides maximum output. Linear Discriminant analysis is used for this purpose. 80 percent of the total data is used for training and remaining 20 percent isused for testing the performance. The classification results obtained from the tests are shown in Table 6.

(b)SNANN based classification

In this model the input signal $x(k)$ represents the five demographic data of the online shoppers, each of which is expanded to three trigonometric terms. These are then fed to the SNANN model. The three outputs obtained from the model using (4) are compared with those obtained from the cluster analysis to produce error signal

$$e(k) = y(k) - y\hat{}(k) \qquad (10)$$

The mean square error is used as the performance index given by

$$MSE(k) = \frac{\sum_{k=1}^{K} e^2(k)}{K} \qquad (11)$$

The epoch based training of weights is carried out using the LMS algorithm. The classification results obtained from test are shown in Table 7.

(c) Proposed SNANNP based classification

The training rule for updating the weights of the proposed SNANNP model is outlined in the following steps:

1.The model is provided with K input patterns each consisting of five demographic features of the consumers. Each feature is expanded to three trigonometric terms and then given to the SNANN model.

Table 5
Demographic characteristics of the consumer under each cluster

		Cluster1	Cluster2	Cluster3	Cumulative
Gender	Male	35	56	47	138
	Female	6	11	8	25
Age	<=20	2	2	2	6
	21-30	29	55	45	129
	31-40	8	9	6	23
	> 40	2	1	2	5
Education	undergraduates	2	3	3	8
	graduates	0	2	4	6
	postgraduates	36	59	46	141
	doctoral	3	2	2	7
	others	0	1	0	1
Amount spent on shopping	<=25,000	24	31	22	77
	25,001-50,000	7	12	15	34
	>50,000	10	24	18	52
Amount spent on online shopping	<=20,000	35	43	39	117
	20,001-40,000	2	11	5	18
	>40,000	4	13	11	28
					163

2. Each new term is multiplied with the corresponding weight and then summed to give an output and in this way K numbers of estimated outputs are computed.

3. Each desired output is compared with the corresponding model output and K errors are produced.

4. The mean square error (MSE) (corresponding to n$_{th}$ particle) is determined by using the relation defined in (11). This is repeated for M times, where M is the number of particles.

5. Since the objective is to minimize MSE (n), n = 1 to M the PSO based optimization method is used.

6. The velocity and position of each particle is updated using (8) and (9).

7. For each iteration the minimum MSE, MMSE is stored which indicate the learning efficiency of adaptive model.

8. When the MMSE reaches the pre-specified value the optimization process is stopped

Table 6

Classification results obtained from testing using discriminant analysis

Classified observations	Cluster 1	Cluster 2	Cluster 3
Class 1	2	3	3
Class 2	6	10	5
Class 3	0	0	3
Cumulative	8	13	11

Table 7

Classification results obtained from testing using SNANN

Classified observations	Cluster 1	Cluster 2	Cluster 3
Class 1	3	0	0
Class 2	5	13	7
Class 3	0	0	4
Cumulative	8	13	11

(b) At this stage all the particles attains almost the same positions, which represent the desired solution of the given SNANN model.

For classification purpose 80 percent of the total data is used for training and remaining 20 percent is used for testing. Various parameters chosen in the simulation study for PSO are : number of particles =10, inertia weight, $w = 0.5$, acceleration constants $c_1 = 0.5$ and $c_2 = 0$, no of generations = 1000 and no. of ensampling average used =20. The classification results obtained from test data are shown in Table 8.

VII. DISCUSSION

Five key factors are obtained from fourteen variables selected based on eigen values greater than one. The factor scores obtained are used in the cluster analysis. Clustering, operation provides 41, 67and 55 consumers in clusters 1, 2 and 3 respectively. From Table 4 it is observed that cluster 1 is heavily loaded by the factor 2. So all the consumers seeking website design belong to cluster 1. Cluster 1 represents website design. Factor1 load heavily on cluster 2.Consumers who are particular about privacy and security belong to cluster 2 and hence is named as privacy and security. Factors which load heavily on cluster 3 are factors 3, 4 and 5. In this cluster the consumers are interested in ease, enjoyment, communication and norms. So cluster 3 is named as ease, enjoyment, communication and norms. The statistics of the demography of the consumers are listed in Table 5.

The accuracy classification for three classifiers using demographic features as inputs are shown in the Tables 6, 7, and 8. The comparative results of the three classifiers are listed in Table 9.

Table 8

Classification results obtained from testing using SNANNP model

Classified observations	Cluster 1	Cluster 2	Cluster 3
Class 1	5	0	0
Class 2	3	13	3
Class 3	0	0	8
Cumulative	8	13	11

Table 9

Comparative results of efficiency obtained during testing

Model	No. of correct classifications	Classification accuracy
Discriminant analysis	15	46.87%
SNANN	20	62.50%
SNANNP	26	81.25%

The results presented in Table 9 demonstrate that the SNANNP model provides best classification performance compared to those obtained by other two methods.

VIII. CONCLUSION

In this paper, a novel method has been proposed for classification of Indian consumers based on their behavior towards online shopping. The factor analysis is carried out on the data to achieve reduced number of factors. The hierarchical based clustering is used to group the consumers. Three different classifiers : discriminant analysis, SNANN and SNANNP are used for classification. The simulation results indicate that the proposed SNANNP based model provides best classification performance compared to its statistical counterpart. The proposed research can also be extended for classification of other real life data. The classification accuracy can further be enhanced by adding physcographic and cultural inputs to the proposed classifier.

REFERENCES

[1] Wan-I Lee, Bih-Yaw Shih and Yi-Shun Chung, “The exploration of consumers‟ behavior in choosing hospital by

the application of neural network", Expert systems with applications, vol. 34, pp. 806-816, 2008.
[2] F. Kaefer, C. M. Heilman and S.D. Ramenofsky, "A neural network application to consumer classification to improve the timing of direct marketing activities", Computers and Operations Research, vol. 32, pp. 2595-2615, 2005.
[3] R. Gates, C. McDaniel and K. Braunsberger, "Modeling consumer health plan choice behavior to improve customer value and health plan market share", Journal of Business research, vol. 48, pp. 247-257, 2000.
[4] R. C. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory", in Proc. of 6th Int. symp. Micro machine Human Sci., Nagoya, Japan, 1995, pp. 39-43, 1995.
[5] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", in Proc. of IEEE Int. Conf. Neural Networks, 1995, pp. 1942-1948.
[6] J. C. Patra, R. N. Pal, B. N. Chatterjee and G. Panda, "Identification of nonlinear dynamic systems using functional link artificial neural networks", IEEE Trans. on Systems, Man and Cybernetics – Part B, vol. 29, no. 2, pp. 254-262, April 1999.
[7] S.Sharma and A. Kumar, *Cluster analysis and factor analysis*, University of South Carolina, Arizona State University.
[8] Glover, Jenkins and Doney, "Principal Component and Factor Analysis, Modeling Methods for Marine Science, pp.81-117, 2008.
[9] N. Jardine and R. Sibson, "The construction of hierarchic and non-hierarchic classifications", the Computer Journal, pp.11-177, 1968.
[10]S.Balakrishnama,A.Ganapathiraju and J. Picone,"Linear Discriminant Analysis for Signal Processing Problems", IEEE, pp. 78-81, 1999.
[11]S.Balakrishnama and A. Ganapathiraju,, Linear discriminant analysis-as brief tutorial, Department of Electrical and Computer Engineering, Mississippi State University.

Recognition of Two-handed Arabic Signs using the CyberGlove

Mohamed A. Mohandes
Electrical Engineering Department
King Fahd University of Petroleum and Minerals
Dhahran, 31261, Saudi Arabia
mohandes@kfupm.edu.sa

***Abstract*--Sign language maps letters, words, and expressions of a certain language to a set of hand gestures enabling an individual to communicate by using hands and gestures rather than by speaking. Systems capable of recognizing sign-language symbols can be used as a means of communication between hearing-impaired and vocal people. This paper represents the first attempt to recognize two-handed signs from the Unified Arabic Sign Language Dictionary using the CyberGlove and support vector machines. Principal Component Analysis is used for feature extraction. 20 samples of each of 100 two-handed signs were collected from an adult signer. 15 samples of each sign were used for training a Support Vector Machine to perform the recognition. The performance is obtained by testing the trained system on the remaining 5 samples of each sign. A recognition rate of 99.6% on the testing data was obtained. When more signs will be considered, the support vector machine algorithm must be parallelized so that signs are recognized on real time.**

Keywords—Arabic sign language; recognition; support vector machine; principle component analysis.

I. INTRODUCTION

Developing a pattern recognition system for sign language interpretation is a very difficult process. One difficulty is that use of traditional programming paradigms makes the system overwhelmingly complex and hence impractical. This dictates resorting to machine-learning methods. Another difficulty encountered is the interface issue. Ideally, the interface should deliver accurate measurements to the processing machine, have low cost, and provide input in a form that requires low pre-processing overhead. Building a system that satisfies these three requirements is very challenging. Hence, design compromises must be done to build a practical system.

Interfaces in sign language systems can be categorized as direct-device or vision-based. The direct-device approach uses measurement devices that are in direct contact with the hand such as instrumented gloves, flexion sensors, styli and position-tracking devices. On the other hand, the vision-based approach captures the movement of the singer's hand using a camera that is sometimes aided by making the signer wear a glove that has painted areas indicating the positions of the fingers or knuckles. The main advantage of vision-based systems is that the user isn't encumbered by any complex devices. Their main disadvantage, however, is that they require a large amount of computation just to extract the hands position before performing any analysis on the images. This paper deals only with the directed-devise methods.

The first widely known instrumented glove is the Digital-Data-Entry Glove [1, 2]. It was originally proposed as an alternative input device to the keyboard and worked by generating ASCII characters according to finger positions. The gloves had finger flex sensors, tactile sensors at their tips, orientation sensors and wrist-positioning sensors. The VPL-DataGlove used novel optical flex sensors that had fiber optic cables with a light at one end and a photodiode at another. A simplified version of the latter is called the Z-glove. It uses fiber optic devices to measure the angles of each of the first two knuckles of the fingers and is usually combined with a Polhemus tracking device. The Z-glove was the first commercially available instrumented glove. The Exon-Dextrous-Hand-Master was developed afterwards with 8 bits of accuracy, 20 degrees of freedom and a measurement frequency of 200 Hz [3]. The PowerGlove is a highly cost effective alternative to other instrumented gloves but less accurate [3]. It is based on VPL's glove and only measures the position in the three dimensional Cartesian space and the roll while other gloves measure the pitch and yaw as well.

Section II of this paper discusses briefly previous work related to sign language recognition. Section III introducers the proposed system, while Section IV discusses the preprocessing and feature extraction. Section V highlights the recognition of the Arabic sign language, and Section VI describes case study of the developed system. Section VII concludes the paper.

II. RELATED WORK

David L. Quam used a DataGlove Model 2 in addition to a Polhemus tracker [4]. Twenty two signs from the American Sign Language (ASL) were provided by two signers, one is the right handed male and the other is the left handed female. Most of the signs are letters or numbers in addition to two selected words. Due to the small set of signs, he used the finger flexions and hand orientation directly as features. This system was only able to identify static signs, thus had limited applications.

Fels and Hington developed a system called Glove-Talk [5]. They used a VPL-DataGlove, having 2 sensors per finger, and a Polhemus tracker. They used five neural networks to connect the gloves to a speech synthesizer. The Glove-Talk project vocabulary consists of 203 words. The Glove-Talk project gives an accuracy of 94%.

Waleed Kadous developed a system called “GRASP” for the recognition of the Australian Sign Language [6]. He used a PowerGlove for collecting data. He used the energy, time, bounding boxes and simple time division over a specified number of segments as features. 6,650 samples of signs were collected from 5 different people. With 95 different signs, the accuracy of the system was about 80%.

Waldron and Kim used a DataGlove with a Polhemus tracker to obtain the hand shape and position [7]. They developed a two-stage neural network system to recognize isolated ASL signs. The first stage recognizes the sign language phonology consisting of 36 hand shapes, 10 locations, 11 orientations, and 11 hand movements using four different neural networks. The second stage uses the recognized phonemes from the beginning, middle, and end of the sign as input to identify the actual sign. Six signers generated 14 signs. The overall performance of sign recognition was 86%.

Sagawa, Takeuchi, and Ohki developed Japanese Sign Language recognition system [8]. The sign is represented as a combination of basic components of gestures. These components are called cheremes. The system uses two CyberGloves and two trackers. A total of 14 cheremes are recognized by the system. The recognized cheremes are sent to the recognition part of sign language morpheme. The system is used to recognize 60 sings. Twenty samples were collected from each signs, 10 samples used for training and 10 for testing. The recogntion rate is 97.6%.

Kim, Jang and Bien investigated the recognition of the Korean Sing Language (KSL) [9]. Two DataGloves and two Polhemus trackers are used. KSL signs can be formed by combining a small number of basic gestures. 25 basic gestures are considered with 14 basic hand shapes. The hand shapes are recognized using a fuzzy min-max neural network. For the recognition process, the direction type is identified and then the hand shape of the motion is recognized. The recognition rate reaches 85% for this algorithm.

Jiangqin and co-investigators used a CyberGlove and a 3-D tracker to recognize signs from the Chinese sign language [10]. Each sign of the 3300 Chinese sign language is characterized by posture, orientation, position and motion trajectory. The number of postures for the right hand and the left hand are 14 and 7, respectively. They used multilayer perceptron to code the data as the input to the Hidden Markov Models. The recognition rate of the samples is over 90%.

Mohandes and Al-Buraiky used a PowerGlove to recognize single-handed Arabic signs [11]. For feature extraction, they used time division where the data is divided into segments and the average of each segment is calculated for each sensor. SVM is used for the recognition process. 36 samples of each of 120 signs were collected from a deaf signer, 18 used for training and the remaining 18 are used for testing. A recognition rate of about 70% was achieved. The CyberGlove was used in our lab for the recognition of single-handed Arabic Signs [12], while this paper uses two CyberGloves to recognize two-handed signs from the Arabic sign language using two CybeGloves and support vector machine. The SVM is optimal on the sense that it maximizes the separation margins among classes and therefore, it is expected to outperform other methods on the recognition of all sign languages.

III. THE PROPOSED SYSTEM

The proposed system consists of two CyberGloves, two hand tracking systems, and data acquisition software. The CyberGlove is a fully instrumented glove that provides 22 high-accuracy joint-angle measurements as shown in Figure 1. It uses proprietary resistive bend-sensing technology to accurately transform finger motions into real-time digital joint-angle data. Each sensor is extremely thin and flexible being virtually undetectable in a lightweight elastic glove. The CyberGlove has been used in a wide variety of real-world applications, including digital prototype evaluation, virtual reality biomechanics and animation. In addition to the CyberGloves two hand tracking devices are added to measure the location (x, y, z) and orientation (yaw, pitch, roll) of each hand with reference to a fixed point. There are three main components of a tracking device: a transmitter generating a signal, a sensor that receives the signal, and a control box for signal processing and connection to the computer.

The tracking device used in this paper is the Flock of Bird (FOB). It is used to track the position and orientation of up to thirty sensors simultaneously by a transmitter. Each sensor is capable of making from 20 to 144 measurements per second of its position and orientation when it is located within 4 feet of its transmitter. The sensor is fixed at the wrist of the CyberGlove, as shown in Figure 1.

Each CyberGlove provides 22 sensor signals and 6 signals are provided from each FOB, thus a total of 56 measurements are provided from the two gloves and the two hand tracking devices while the signer is performing the signs. These measurements are sent through the serial ports and stored in a human readable format (ASCII text).

Figure 1. The CyberGlove

IV. PREPROCESSING AND FEATURE EXTRACTION

A continuous stream of frames is generated by the gloves and the hand trackers as the hands perform the sign. Each frame bears an instantaneous measurement of each of the 56 signals for the two hands. Signs have different lengths. Even samples of the same sign performed by the same signer may have different lengths as well. The classification system requires the same number of inputs. Therefore, time division is used to make all signs have the same number of components. In this paper, the duration of the sign is divided into 10 segments. The mean and standard deviation are calculated from each signal in each of the 10 segments. Thus, each signal is represented by 20 values of the means and standard deviations of the segments. Thus, the signals from the 56 sensors will be represented by 1120 values. Principal Component Analysis (PCA) is used to reduce the dimensionality of the data. Thus PCA is used to provide features for the classification machine.

Given a d-dimensional vector representing the mean and standard deviation from each segment of the raw data, the Principal component analysis (PCA) can be used to find a subspace whose basic vectors correspond to the maximum-variance direction in the original space [13]. Let W represent the linear transformation that maps the original d-dimensional space onto a lower dimensional space F-dimensional feature subspace. The new feature vectors $y_i \in R^F$ are defined by $y_i = W^T x_i, \quad i = 1,\ldots,N$. the columns of W are the eigenvectors e_i obtained by solving the equation $\lambda_i e_i = Qe_i$, where $Q = XX^T$ is the covariance matrix, and λ_i is the eigenvalue associated with eigenvector e_i. Before obtaining the eigenvectors of Q, the vectors are normalized and the mean is subtracted from all normalized vectors. In this paper a different number of eigenvlaues have been used.

V. RECOGNITION OF ARABIC SIGNS

After feature extraction, the signs are ready for the classification. In this paper, Support Vector Machine (SVM) is used for the recognition of the signs. In this section SVM is briefly introduced. In its simplest form, SVM is based on constructing a hyperplane that separates two linearly separable classes with the maximum possible margin [14]. Assuming that there is a set of vectors *x* each having a label *y* and that the separating hyper plane is $w.x + b = 0$, where *w* is a weight vector and *b* is a constant. It can be shown that the decision function (hypothesis), which is based on the optimal hyperplane, takes the form:

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i x_i * x + b \qquad (1)$$

providing that the coefficients a_i maximize the following function:

$$L_D = \sum_{i=1}^{l} a_i - \frac{1}{2} \sum_{i,j}^{l} y_i y_j \alpha_i \alpha_j x_i * x_j \qquad (2)$$

where L_D is called the dual Lagrangian function and is obtained by deriving the dual of the optimization problem formulated to maximize the classification margin. A support vector machine for separating non-linearly separable data can be built by first using a non-linear mapping that transforms data from the input space to a higher space called the feature space, and then using a linear machine to separates them in the feature space. Mapping to the feature space can be performed by replacing the dot products with a kernel function $K(x,z) = \phi(x).\phi(z)$ (where $\phi : X \to F$ is a non-linear mapping from input space *X* to feature space *F*). A number of different kernel functions can be found in the literature. The kernel function used in this paper is the Radial Basis Kernel defined as

$$K(x, y) = e - \gamma \|x - y\|^2 \qquad (3)$$

By replacing the dot product with the kernel function the decision function (the hypothesis) becomes:

$$f(x) = \sum_{i=1}^{l} \alpha_i y_i K(x_i.x) + b \qquad (4)$$

This allows the SVM algorithm to solve real-world problems where data can be non-linearly separable. Additionally, in feature space, some slackness can be introduced in the support vector machine developed above so that some error is tolerated. This is done by adding slack variables (representing violations of the margin constraints) to the cost function. With this modification the optimization problem becomes:

Minimize $w.x + C\sum_{i=1}^{l}\zeta_{i}^{2}$ (5)

subject to $y_i\left((w_i.x_i)+b\right)\geq 1-\varsigma_i, i=1,2,..l$ (6)

where $\varsigma_i, i=1,2,..l$ are slack variables and C is a penalty error constant coefficient whose best value is determined in practice by trail and error adjustment.

Solving for support vectors (optimizing the dual function) is a quadratic convex programming problem, for which many numerical solution algorithms exist. Figure 2 shows the structure of the SVM classifier.

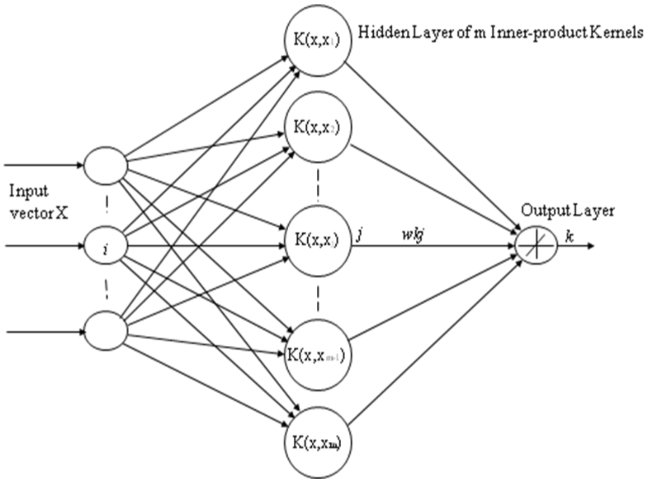


Figure 2. The Support Vector Machine Structure

The above formulations can be extended to the classification of n classes by one of the following methods :

1. Build *n* classifiers, each capable of separating patterns belonging to one class from all other patterns.
2. Build the *n*-class classifier by feeding input to each of the two-class classifiers and choosing the class corresponding to the maximum $f_k(x), k=1,2,....,n$

The multi-class problem can be solved in a direct manner as well as by generalizing the procedure used for the two-class case [14].

VI. CASE STUDY

A volunteer from the deaf community performed the signs to generate samples for the learning machine. The signer was chosen among adults to insure his fluency in sign language and the accuracy of the signs. The Signer performed twenty samples of each of 100 two-handed signs selected from the Arabic Sign Language Dictionary. The signer starts with his two hands resting on his side as shown in Figure 4. A button is pressed to signal the start of the sign. As soon as the sign is completed, the button is presses again. This process is repeated 20 times for each sig to produce a total of 2000 samples.

The 20 samples of each sign are divided into two parts: training and testing. The training data for each sign consists of 15 samples, and the remaining 5 samples are used for testing. The duration of every sign is different, even the samples of the same sign implemented by the same signer will take different time. Thus the number of data points from each sensor is different. For example, the 20 samples of the first sign have a number of data points that ranges between 15 and 22, as shown in Figure 3. The number of data points for the other signs ranges between a maximum of 40 and a minimum of 10. For the recognition machine, the number of data points on all samples of the signs should be the same. Therefore, a pre- processing step is added to unify the number of data points. The duration of each sign implementation is divided into 10 segments. If the number of data points is not a multiple of 10 then the extra points are distributed among the first segments. For example if a sign has 23 data points, then each segment will have 2 points except the first three segments will have 3 data points. The mean and standard deviation of each segment is calculated. Thus, each sensor signal is represented by 20 values which are the mean and standard deviation of the 10 segments of the signal.

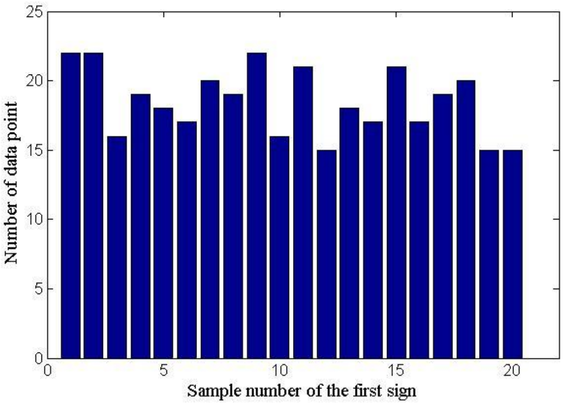


Figure 3. The number of data points of the 20 samples of the first sign

There are 56 measurements of the signs including 22 from each glove and 6 from each hand tracker. Each sensor signal is represented by 20 values which are the mean and standard deviation of each segment. Therefore, each sign is represented totally by a vector of length 1120 values. PCA is used for dimensionality reduction as explain in Section IV. Thus, the 2000 vectors representing 20 samples of each of 100 signs are normalized and the mean is subtracted. The covariance matrix is found and its eigenvectors and eigenvalues are calculated. The vectors, each of length 1120 values, which represent the samples of the signs, are transformed to a lower dimension by a linear transformation matrix formed by the eigenvectors of the covariance matrix. The number of eigenvectors chosen determines the size of the feature vector. The eigenvalues represent the variance of the data when

ISBN: 978-1-61208-101-4

projected onto the corresponding eigenvectors. Therefore, only the eigenvectors corresponding to the highest eigenvalues need to be considered. Figure 4. Shows the values of the first 100 eigenvalues, where the first eignevalue is 8,5629 and the 100th eigenvalue is 0.9133.

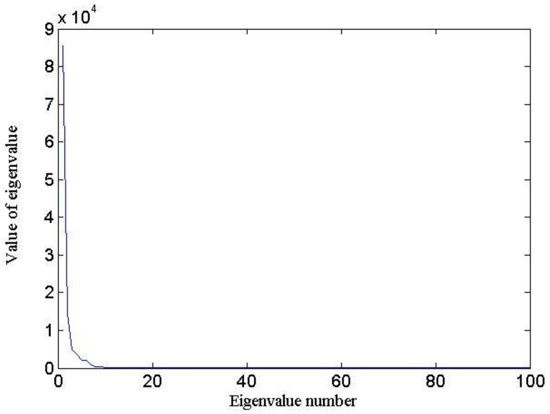


Figure 4. The eigenvalue spectrum

To determine suitable dimensionality of the feature vector for this application, an experiment is performed where several numbers of eigenvectors between 1 and 1120 are considered. The accuracy of the recognition system on the testing samples is calculated. Figure 5. shows the relation between the size of the feature vector and the accuracy of the recognition system on testing data. The figure indicates that a feature vector of size 70 would give the best performance. The figure indicates also that increasing the size of feature vector beyond 500 elements degrades the performance is it adds irrelevant information.

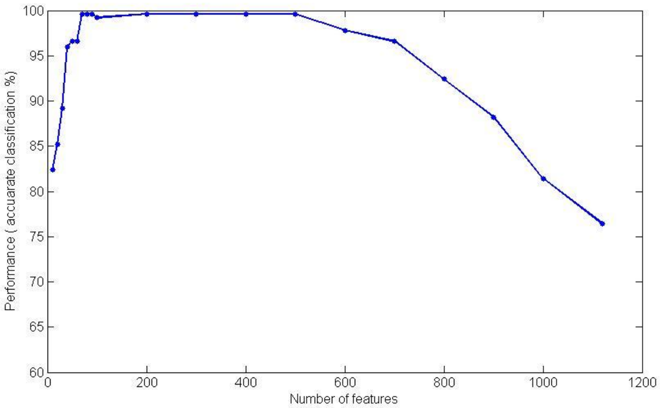


Figure 5. Performance with different number of feature vector

A support vector machine is trained using 15 samples of each of the 100 collected signs. The SVM has 70 inputs and one output. The output unit takes a value between 1 and 100. The value of the output unit indicates the sign that the input vector is assigned to. The Kernel function used in the SVM is the Radial Basis Kernel. After several experiments, it was found that a suitable values of the user defined SVM parameters for this application are the error penalty constant, C= 110, and $\gamma = 0.15$. The trained SVM is tested using the remaining 5 samples of each sign that have not been used in training. The performance of the trained SVM on the testing data lead to 99.6% correct classification where only two samples of the total 500 are misclassified as other signs. The two samples belong to one sign that is misclassified to another sign where the two signs differ only on the location of the hands, while the all the sensors have almost the same values as seen in Figure 6. The result shows the viability of SVM for the recognition of Arabic sign language.

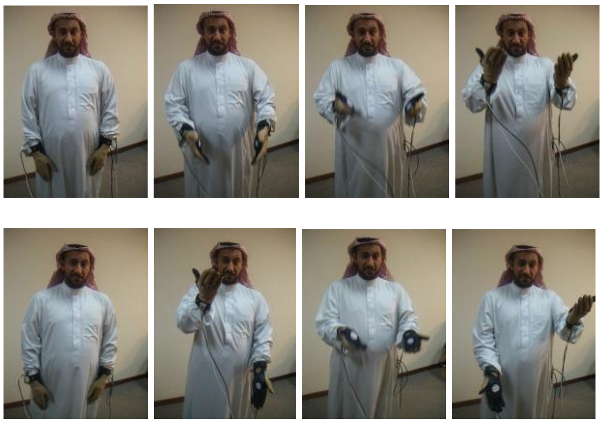


Figure 6. Frames of the two signs that has been misclassified

To further test the performance of the developed system, another signer provided about 300 samples from 15 signs. The samples were sued to test the already trained SVM. However, the recognition rate did not exceed 63%. The low recognition rate could be due to the fact that the Arabic Sign language is not fully standardized and the two signers are from two different areas of the Kingdom. However, when samples from the second signer are included in the training, the recognition rate reaches 93%. This result indicates the need to collect samples from more signers to fully test the possibility of signer independent system.

The recognition process of each sign at this stage takes about 3 seconds. However, when a large number of signs are considered, the SVM has to be parallelized so that the recognition is done in real time.

VII. CONCLUSION

This paper is a contribution to the area of Arabic Sign Language recognition, which had very limited research. Two CyberGloves and two trackers are used to collect the signs data. The gloves and trackers provide 56 signals. The durations of the signs are different; therefore, the collected data is pre-processed by dividing the duration of each sign into 10 segments and taking the mean and standard deviation of each segment. Thus each sign is represented by a vector of length 1120 components which represent the mean and the standard deviation of each segment of the sign. Principal component analysis is used for feature selection. A support vector machine is used for the recognition. 20 samples of each of 100 different signs are collected from an adult deaf signer. 15 samples are used for training and 5 for testing. The PCA is used to get effective features from these signals. A feature vector of size 70 is shown to classify the signs very well. A recognition rate of 99.6% was achieved with only 2 signs are misclassified among the 500 signs used for testing which indicates a good performance of the developed system. For future work all signs of the Arabic Sign Language Dictionary will be recognized and several signers will provide the samples so that we reach a signer independent recognition system.

ACKNOWLEDGEMENTS

The work in this paper is partially funded by King Fahd University of Petroleum and Minerals Grant No. IN 100045 and by Prince Salman Center for Disability Research Grant No. 0065, and by the Arab Science and Technology Foundation Grant No. ALJ- FFC07. The author thanks M. Deriche for valuable discussions.

REFERENCES

[1] J. Kramer and L. Leifer, “The Talking Glove for non-Verbal Deaf Individuals”. Technical Report CDR TR 1990 0312, Center for Design Research, Stanford University, 1990.

[2] Peter Vamplew, “The SLARTI Sign Language Recognition System” University of Tasmania.

[3] D. J. Sturman and D. Zeltzer. “A Survey of Glove-Based Input”. IEEE Computer Graphics and Applications, 14(1), pp. 30-39, January 1994.

[4] D. Qaum, ”Gesture Recognition with a DataGlove,” IEEE Proc. Aerospace and Electronics Conference, vol.2, pp. 755-760, May 1990.

[5] S. Fels and G. Hington, "Glove-Talk: A Neural Network Interface Between a Data-Glove and a Speech Synthesizer," IEEE Trans. Neural Networks, vol. 4, pp. 2-8, Jan. 1993.

[6] W. Kadous, “*GRASP:* Recognition of Australian Sign Language Using Instrumented Gloves”. Bachelor’s Thesis, The University of New South Wales, 1995.

[7] M. Waldron, and S. Kim, “Isolated ASL Sign Language Recognition for Deaf Persons”, IEEE Trans. Rehabilitation Engineering, pp. 261-270, Sept. 1995.

[8] H. Sagawa, M. Takeuchi, and M. Ohki, “Description and Recognition Methods for Sign Language Based on Gesture Components,” Proc. IUI97, pp. 97-104, Orlando, Florida, ACM, 1997.

[9] J. Kim, W. Jang, and Z. Bien, “A Dynamic Gesture Recognition System for the Korean Sign Language (KSL),” IEEE Trans. System, Man and Cybernetics, pp. 354-359, vol. 26, no. 2, April, 1996.

[10] W. Jiangqin, G. Wen, S. Yibo, L. Wei and P. Bo, “A Simple Sign Language Recognition System Based On Data Glove,” Proc. ICSP ’98, pp. 1257-1269, 1998.

[11] M. Mohandes and S Al-Buraiky, “Automation of the Arabic Sign Language using the PowerGlove”, the ICGST International Journal on Artificial Intelligence and Machine Learning (AIML), V. 7, Issue 1, pp. 41-46, 2007.

[12] I. Al-Saihati, “Feature Extraction for Real Time Recognition of the Arabic Sign language”, MS thesis, KFUPM 2006.

[13] A. M. Martinez and A. C. Kak, “PCA versus LDA”, IEEE Transaction on Pattern Analysis and machine Intelligence, Vol. 23, No. 2, pp. 228-233, 2001.

[14] V. N. Vapnik, Statistical Learning Theory, John Wiley and Sons Inc., 1998.

Streaming Legacy Desktop Software from the Cloud

Youhui Zhang, Gelin Su, Weimin Zheng
Department of Computer Science and Technology
Tsinghua University
Beijing, China
{zyh02, sgl08, zwm-dcs}@tsinghua.edu.cn

***Abstract*—Desktop Cloud can enhance business agility and reduce the total-cost-of -ownership, it introduces long network latencies and the power of local PCs cannot be utilized fully. This paper presents a light-weight mode for desktop cloud, which stores legacy desktop software in the cloud storage while streaming and running them on the user's PC locally. In details, based on the light-weight virtualization, software can be converted into portable counterparts and stored in the cloud. Moreover, a run-time system is implemented, including a user-space file-system for cloud, to stream and run the remote software on local machines. Local cache and data pre-fetch mechanisms are also adjusted to suit the file-access-pattern of software. This prototype has been implemented and tests show it is practical for much daily-used software.**

Keywords- Cloud computing; user-space file system; OS-level virtualization

I. INTRODUCTION

Existing software delivery model is usually based on a large number of distributed PCs executing operating system and desktop software independently. As mentioned by Gartner Group [1], for enterprises, deploying and managing personal operating systems and software in this mode are very expensive, which is the most important determinant of PC total cost of ownership (TCO). Personal users also face the similar problem.

Desktop Virtualization [2][3], combined with cloud computing allows users to run desktops on virtual machines (VM) hosted at the data center and access them as a service through some remote desktop protocol (RDP) [4][5], which is also called as "Desktop Cloud" [6]. Then, users can enhance business agility and reduce business risks, while lowering TCO.

But, for this solution, the client PC is used as a thin-client device, which executes the graphical interface of desktop to convey input and output between the user and the data center where software is really running. Therefore, there are two drawbacks: the user's feeling would not be good when it is employed across the Internet because of the long network latency [7]; secondly, processing power of the client PC cannot be utilized fully. To solve these problems, some Desktop clouds using Web applications are provided [8] [9]. Now, modern web applications are driving toward the power of fully functional desktop software such as email clients, productivity apps, etc. The user can access the personalized operating environment anywhere. But, the enormous legacy desktop software cannot be used in this model.

To fully utilize local PCs and legacy desktop software, a light-weight Desktop cloud solution is proposed here, which stores legacy Windows desktop software (rather than VM) in the cloud storage, and streams and runs them on the user's PC on-demand.

Because software is executed locally, the power of client PCs can be used efficiently; on the other side, as existing cloud storage services can be used as the backend without any modification, some key features of cloud computing, such as dynamic scaling of infrastructure, flexible usage based pricing, rapid service provisioning, can be still maintained.

To reach this target, three challenges should be conquered:

1) Legacy desktop software should be converted into portable software transparently.

OS-level virtualization is employed here to solve this problem. Every virtualization environment shares the same execution environment as the host machine. Therefore, such an environment can have very small resource requirements and thus its overhead is light-weight.

In our approach, existing desktop software is made portable: each software instance runs in an OS-level virtualization environment. This environment intercepts some resource-accessing APIs from the instance, and redirects them to the actual storage position(s) rather than the host. Then, in the user's view, he / she can launch software conveniently, although it does not exist on local disks.

2) Users should access the portable software in the cloud just like common desktop software; therefore a transparent and friendly delivery mechanism is needed.

A Windows user-space file system for cloud storage is designed. It acts as a proxy for file system accesses: file operation requests from portable software (e.g., CreateFile, ReadFile, WriteFile, etc.) to the Windows I/O subsystem (runs in kernel mode) will be forwarded to the corresponding user-space callback functions which visit real data in the cloud and send results back.

We implement such a file system based on Dokan [10], a development framework for Windows user-space file system (like fuse [11] for Linux), and the Amazon S3 interface.

3) Performance optimization

Some optimizations are adopted: all metadata is pre-fetched by client ends, which will be updated as necessary during the running time; local cache for frequently-used data is enabled to decrease the number of remote accesses, as well as a data pre-fetch mechanism which can adapt to different access patterns of diverse software.

We implement such a prototype for Windows OSes and extensive tests show that, these optimizations are efficient for much daily-used software.

This solution has its own limitation: portable software can only be used on compatible local OSes, while the traditional desktop clouds support any local OS only if a browser or some proper RDP client is available. However, we believe our proposal is practical because now Windows OSes still dominate the desktop PC market.

In this paper, we first present the model of portable software and the design of its runtime system. The user-space file system and optimizations are given in Section 3, as well as the access control mechanisms. The prototype is introduced in Section 4, as well as the performance tests. Finally, we present related works and the conclusion.

II. Portable Software

Usually, Windows software can be regarded as containing three parts: Part 1 includes all resources provided by the OS; Part 2 contains what are created/modified/deleted by the installation process; and Part 3 is the data created/modified/deleted during the run time. For Windows OS, the resources here mainly refer to files/folders and the related system registry keys/values.

During the runtime, the software instance accesses resources of all parts on the fly: some resources are read-only while some may be modified/added/deleted. So, no part is fixed: those modified at run time will be moved into Part 3.

To make the existing software portable, all parts should be captured and made portable except for Part 1 while Part 2 and 3 should be accessed on demand.

A. *Installation Snapshot*

The modifications made by the software's installation process must be captured to enable Part 2 portable. Some system monitoring tool, like InstallWatch [12], is used.

In this implementation, a target application is installed on one clean Windows system, while InstallWatch is running to log those files created or modified in this process, as well as registry additions and modifications. Then, all files/folders/registry-keys created or updated are collected to be stored in a dedicated position. Till now, Part 2 is obtained.

B. *Runtime System*

Detours [13], a library developed by Microsoft Research Institute, is used to intercept those Windows APIs accessing files and registry entries during the runtime. Then, all accesses to files and registry entries are intercepted and redirected to the dedicated storage position as needed. In another word, API Interception is employed to complete a lightweight virtualization environment to make all parts accessible by the software's executable file transparently.

The strategies are:

1) Any non-modification operation is executed on site;

2) Any modification is moved to Part 3 so that the local host can be kept unchanged;

3) Any query will return the combination of results from all parts. If there is any duplication, Part 3 owns the highest priority while Part 1 is the lowest.

For details, please refer to our previous work [15][16].

III. Streaming Software from the Cloud

Now the windows software can run without installation as the runtime system provides all resources transparently.

Then, the next question is how to design a delivery solution that should own the following features or functions:

Transparent to users and software; access control; high efficiency on network access; dependent on the OS to the minimum extent

We design a user-space file system for cloud storage to reach the target. Firstly, with file system interfaces, the access method is compatible with the operation style of Windows desktops. And from the viewpoint of users, the remote software looks just like stored in a local drive

Secondly, two aspects of access control are implemented. The first is based on API interception to prevent portable software from accessing some local private information. The second works the other way round, which uses the process-hierarchy information to protect files of portable software from illegal copy.

Some optimizations are also adopted: all metadata is pre-fetched by client ends, and will be updated when necessary; local cache for frequently-used data is enabled to decrease the number of remote accesses, as well as an adaptive data pre-fetch mechanism. The user-space implementation can achieve most above functions in the user level, which is helpful to port our system to other OSes.

Finally, it is necessary to note that, the backend cloud inherently owns some features like dynamic scaling, high availability and rapid service provisioning. Therefore, this paper is focused on the client-end design, which communicates with the backend via some standard protocol.

A. *The file system framework*

This framework contains four parts: the first is the software instance accessing the user-space file system. Its related file operations are sent the Windows IO subsystem, which will be intercepted by our kernel proxy driver that redirects them to the user-level interception program that registers some callback functions to process corresponding operations respectively.

For more details, please refer to our previous work [15][16].

1) File MetaData

When a user launches the file system first time, the interception program contacts the remote server for login. Then it gets all metadata of his/her customized portable software and the version number based on the user ID. The metadata contains the following information:

Full paths of all files and folders; the attribute, size, creation-time, last-access-time and last-write-time of all files.

The received metadata is saved on the client permanently. When there is any file modification in the cloud, the updated metadata will be sent back during the subsequent procedure to keep data consistency.

Therefore, any metadata access can be completed locally, which speeds up the corresponding operations (like browsing directory, etc.) remarkably.

2) File Data

When a file is opened for read, it will be redirected to the remote position to fetch the real data (the local cache and pre-fetch are both employed, which will be described later).

If there is any write, a copy-on-write method is used, which means the whole remote file will be fetched to the client at first, and then any subsequent operation can happen locally.

As the file system is being unmounted, any new and modified files/folders will be transferred to a remote position (reserved for every user) hosted in cloud. So at the next time, the user can reach his/her latest metadata and data of all files.

3) Remote access

Our file system communicates with the backend through the S3 [14] interface, which is used by Amazon's notable cloud storage service. In S3, data is organized as objects in a bucket identified by unique IDs, which can be accessed through the standard HTTP protocol. Therefore, any file of portable software is regarded as an object in S3 and is identified by its full path name. Its URL looks like http://server_address/portablesoftware /full...path/filename.

For any new or modified file of a user, its naming style is different. For example, if John creates a new file (\program files\app1\file.name), the URL looks like http://server_address/portablesoftware/john/program files/app1/file.name.

In addition, for each user, the metadata info of his/her portable software, combined with the above-mentioned lists, is stored in a special position: http://server_address/ portablesoftware/username/metadata_list_version_num.

In summary, all users share portable software stored at the common place; each has the private space for any new or modified files to avoid write conflicts, as well as all metadata. Then, any whole file can be downloaded with the HTTP GET method while any part of a file can be accessed with the same method using the Range Header Field.

B. Access control

1) Protect the local info

As mentioned in Section 2, file accesses from portable software are intercepted, therefore the user can configure a white list to restrict the allowed range to protect his/her private info.

Another method is that any process of portable software is spawned with less permission rights; for example, its Access control List (ACL) is set as the Guest privilege. So the private data of the current user can be protected.

2) Protect the portable software

Users can access software files just like they are using the local file system, so how to prevent the illegal copy is a key consideration.

An access control based on the process-hierarchy is designed to protect essential files. The root of the hierarchy is the interception program that can access all files while any process outside of the hierarchy is forbidden.

For more details, please refer to our previous work [15][16].

C. IO optimizations

The user-space file system is grounded on the backend cloud. If all reads were completed remotely, our solution would be very slow. To alleviate this, two methods are adopted.

1) Local cache

We analyzed the file-access-pattern of the running process for some frequently-used software; it is found that, most frequently-used files belong to those accessed during the startup process, which only occupied a limited ratio of the whole capacity. For example, the following frequently-used software is converted into portable versions:

Abiword[1]*, PhotoShop, Lotus Notes, VLC (a powerful media player), 7Zip, UltraEdit, ClamWin (an anti-virus program), FileZilla, Gimp (an open source picture editor), Acrobat Reader, WarZone2100 (a real-time strategy game), On-screenkeyboard.*

Tests show that, the average ratio of the amount of data accessed during the start-up process to the whole capacity for the given software is about 21%.

Based on this observation, some frequently-accessed data is cached locally and its replace strategy is also based on the usage frequency.

At the first time, the cache is empty and then the run speed is fairly slow. During the run time, the cache is fulfilled according to the usage frequencies of data. Then for the following runs, the performance is improved because reads will be partly hit in the local cache.

2) Data pre-fetch

Besides the local cache, pre-fetch is another potential method to reduce the number of data access across the Internet. And its efficiency depends on the concrete access mode: for sequential accesses, it will be highly efficient.

We study the access behavior on any single file. For a given file, two arguments are defined: a is the ratio of the number of sequential reads to the total read-number, and b is the ratio between read amounts. The greater the value of a or b is, the better the effect of pre-fetch is. A file is sequential if and only if the values of a and b are both more than a threshold. Another conclusion from the analysis is that, for given software, its file-access-pattern is fixed regardless of its storage position. Then, we only adopt pre-fetch for these sequential files. In the current implementation, the threshold is set as 66% and the pre-fetch distance is 32KB.

IV. PROTOTYPE AND TESTS

We have implemented the prototype using VC 2005.

A. Performance Tests

1) Test Methods

Two types of performance metrics are measured.

- Start-up time

[1] The Microsoft Office applications can also be made portable by us, but it cannot run on the virtual file system because of some bugs of DOKAN. As stated at http://dokan-dev.net/.

The application start-up time is the key metric of the prototypes' usability: the time it takes for applications to begin to respond to user-initiated operations is a measure of what it feels like to use the system for everyday work.

In our test, CreateProcess is invoked to launch the given software, and then another API WaitForInputIdle is used to judge whether the new process has finished its initialization and is ready to response user's input or not. As WaitForInputIdle returns, the elapsed time is logged as the start-up overhead.

- Run time

A special program is used to record the user's inputs of the keyboard and mouse and replay them after start-up. Based on this tool, we design scripts to control software to complete a series of operations, which looks like triggered by a real user. For example, for the word processing software, one document is created and compiled for several seconds and saved before termination. Moreover, between any two continuous operations, some random waiting time (less than one second) is inserted to simulate the human's behavior.

The elapsed time is logged as the run time.

2) Test Environments

The client platform is a Windows Vista PC, equipped with 2 GBytes DDR2 SDRAM and one Intel Core Duo CPU (1.86GHz). The hard disk is one 160 GBytes SATA drive.

It uses one 100M Ethernet adapter to access the Internet.

The client machine should cross the Internet for data. So where to place the server is decisive for the performance. Two cases are considered.

In Case 1, it is assumed that some edge server can be found to provide the download service, therefore the web server is located in the CERNET (Chinese Education & Research Network, is the second largest network backbone in China.) as well as the client PC. This is a common case now: Content Delivery Network has been widely used for software downloading. As disclaimed by *Akamai*, the world leading CDN provider, most visit requirements can be fulfilled by some edge server(s) just a single-hop away.

The network throughput between the client and this server is about 1.89MBps and the average response time is about 6ms, which are tested by Qcheck, a free and professional network benchmark program.

In Case 2, the server is located outside the CERNET. The throughput is 998KBps and the response time is 32ms.

Two Windows 2003 servers, equipped with one Intel Core 2 Duo E4500 CPU (2200MHz), 2 GBytes DDR2 SDRAM, and one 240GBytes SATA II disk, are used for these two cases respectively.

3) Test cases

Case 1: The original start-up time / run time (portable software is saved in one local disk).

Case 2: The start-up time / run time based on the user-space file system (portable software is saved in one local disk, which is also mirrored as a virtual drive; then the software is launched through this drive.)

Case 3: The start-up time / run time based on the user-space file system for the remote server located inside the CERNET; no cache, no pre-fetch;

Case 4: The server is located inside the CERNET; the cache hit ratio is 20%, no pre-fetch;

Case 5: The server is located inside the CERNET; the cache hit ratio is 33%, no pre-fetch;

Case 6: The server is located inside the CERNET; the cache hit ratio is 50%, no pre-fetch;

Case 7: The server is located inside the CERNET; the cache hit ratio is 66%, no pre-fetch;

Case 8: The server is located inside the CERNET; the cache hit ratio is 80%, no pre-fetch;

Case 9: The server is located inside the CERNET; no cache, the pre-fetch size is 32KB;

Case 10~Case 16: The server is located out of the CERNET and other conditions are the same as those of Case 3~Case 9.

Some software cannot be controlled by our automation method; therefore the software number in the run-time tests is less.

We present the detailed results from Figure 1 to Figure 5. To present clearly, all results have been normalized, compared with the values of Case 1.

4) Test results

For the start-up time (Figure 1 ~ 3), the user-space file system itself introduces less than 96% extra overheads (comparing Case 2 with Case 1), as the file system causes more context-switch operations.

The exception is WarZone2010 (in Figure 1), whose extra overheads are much more because its access-pattern is special: it reads one sequential file many times while each time only two bytes are fetched. Similarly, when our file system is based on the remote servers, its start-up time becomes much longer: about 7720% extra overheads in Case 3 and 26590% in Case 10 (compared with Case 1). But pre-fetch is very efficient for this game, the speed-up ratios are 11 in Case 9 and 17 in Case 16, compared with Case 3 and 10 respectively.

For the other software (in Figure 2 and 3), our system introduces about 890 % extra start-up time on average in Case 3 and 1452% in Case 10 (compared with Case 1). When the cache-hit ratio is 80%, the corresponding results are 88% and 264%. So, the network performance largely determines program behaviors; and local cache is a highly-efficient method to improve the performance, except for some tiny software because their disk-IO overhead is so small that the IO sub-system affects its whole performance little. For pre-fetch, the speed-up ratios are 1.25 in Case 9 and 1.44 in Case 16 respectively.

As mentioned in Section 3.3, the most frequently-used files only occupied a limited ratio of the whole capacity. In our tests, one local cache of 140MB can reach the hit-ratio of 80%. Then, for much frequently-used software, after several runs, their performance on our system is really acceptable.

For run-time tests (in Figure 4 and 5), the results are better: because the waiting time overlaps the background transfer operations and much data required has been fetched during start-up, about 11% extra overheads in Case 8 and 18% in Case 15 are introduced. Because our scripts only complete some simple and common operations, this result is for reference only.

V. RELATED WORK

A. Cloud Computing

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. As [16] said, there are three types of Cloud Computing, which are classified based on the level of abstraction presented to the programmer.

Amazon EC2 [17][18] is at one end. It presents a virtual computing environment, allowing customers to launch instances with a variety of operating systems, manage network's access permissions, and run image, which is compatible with legacy desktop software completely.

Google AppEngine [19] is at the other extreme. It is an application-domain specific platform, which just hosts traditional web applications. As we know, this mode is incompatible with the desktop software, which asks developers to write new applications.

Accordingly, there are mainly two types of Desktop cloud. The first is based on the thin-client computing mode, which hosts VMs running desktop systems on the data center and users access them through some RDP. IBM Smart Business Desktop Cloud [20], VMWARE's ThinApp [21] and Citrix's XenAPP belong to this catalog.

The second refers to the Web-Application based [8] [9]. It provides a desktop-like GUI on the browser, which contains many web applications.

Another important service provided by the cloud computing is cloud storage, like Amazon's S3, Microsoft's Live Skydrive [22] and so on. Cloud Storage delivers virtualized storage on demand, over a network based on a request for a given quality of service (QoS).

B. Software Streaming

Virtualization has been deployed for software streaming. A solution is Progressive Deployment System (PDS) [23], which is a virtual execution environment and infrastructure designed for deploying software on demand. Another practical solution is Microsoft's SoftGrid [24]. SoftGrid can convert applications into virtual services that are managed and hosted centrally but run on demand locally.

Our previous work [15] also provides a solution for software streaming based on lightweight virtualization and p2p transportation technologies. Compared with [15], this work is based on the cloud storage and a new user-space file system is introduced. Another previous work [16] of ours is about how to fast deploy desktop software in a VM-based cloud environment (like EC2).

VI. CONCLUSION AND FUTURE WORK

This paper presented a solution to convert the existing Windows desktop software to on-demand application stored on the cloud storage, which could be regarded as a mode of SaaS. As we know, this is the first prototype of such a solution. Two main technologies were used: the first was OS-level virtualization, which made legacy software portable; and the second was a user-space file system that provided the user a transparent interface to access them. In addition, some access control mechanisms were implemented.

Owing to the local cache and pre-fetch mechanisms, tests showed that, for much frequently-used software, their performance was acceptable with a limited local cache.

ACKNOWLEDGMENT

This research is supported by Chinese National 973 Basic Research Program under Grant 2007CB310900, Chinese National 863 High Technology Programs under Grant 2008AA01A201.

REFERENCES

[1] Federica Troni and Michael A. Silver. Use Processes and Tools to Reduce TCO for PCs, 2005- 2006 Update. Gartner Group.

[2] IBM Virtual Infrastructure Access Service Product. https://www-935.ibm.com/services/au/gts/pdf/end03005usen.pdf. 07.07.2010.

[3] Windows Server 2003 Terminal Services. http://www.microsoft.com/windowsserver2003/technologies/terminalservices/default.mspx. 07.07.2010.

[4] Tristan Richardson, Quentin Stafford-Fraser, Kenneth R. Wood, and Andy Hopper. Virtual network computing, Internet Computing, IEEE, Volume.2, No.1. January, 1998, pp.33-38.

[5] CITRIX, http://www.citrix.com/lang/English/home.asp. 07.07.2010.

[6] Kirk Beaty, Andrzej Kochut, and Hidayatullah Shaikh. Desktop to cloud transformation planning. Proceedings of 2009 IEEE International Symposium on Parallel & Distributed Processing. Rome, Italy, May 23-29, 2009, pp.1-8.

[7] Albert Lai and Jason Nieh, On the Performance of Wide-Area Thin-Client Computing. ACM Transactions on Computer Systems (TOCS), Volume 24, Issue 2. May 2006, pp. 175-209.

[8] iCloud, http://icloud.com/. 07.07.2010.

[9] Lifehacker, the Full-Screen Firefox Cloud Desktop. http://lifehacker.com/5256657/the-full +screen- firefox-cloud-desktop. 07.07.2010.

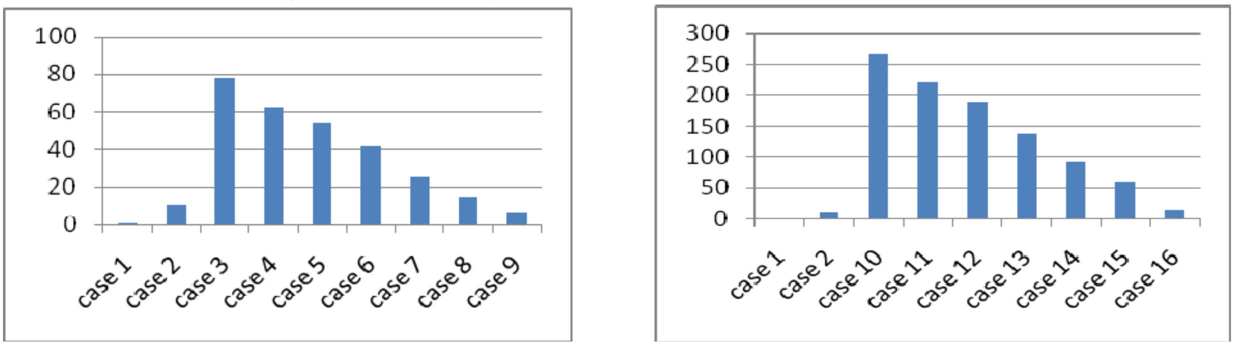


Figure 1. The start-up time of WarZone2010 (the left part is the case that the server is inside).

[10] Dokan, user mode file system for windows. http://code.google.com/p/dokan/. 07.07.2010.

[11] Filesystem in Userspace. http://fuse.sourceforge.net/. 07.07.2010.

[12] Installwatch. http://tejasconsulting.com/open-estware/feature/installwatch.html. 07.07.2010.

[13] Galen Hunt and Doug Brubacher, Detours: Binary Interception of Win32 Functions, Proceedings of the Third USENIX Windows NT Symposium, July, 1999, pp.135-144.

[14] Amazon Simple Storage Service (Amazon S3). http://aws.amazon.com/s3/. 07.07.2010.

[15] Youhui Zhang, Xiaoling Wang, and Liang Hong, Portable Desktop Applications Based on P2P Transportation and Virtualization. Proceedings of the 22nd Large Installation System Administration Conference (LISA '08) San Diego, CA. USENIX Association, November, 2008, pp. 133–144.

[16] Youhoi Zhang, Gelin Su, and Weimin Zheng: "On demand mode of legacy desktop software and its automatic deployment for Cloud-Computing Environment", Proceedings of the Sixth Workshop on Grid Technologies and Applications (WOGTA 2009), 18-19 Dec 2009, Taitung, Taiwan, pp.25-31.

[17] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, et al. Above the Clouds: A Berkeley View of Cloud Computing Export. Technical Report. 10 February 2009. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html. 07.07.2010.

[18] Amazon Elastic Compute Cloud. Developer Guide. http://docs.amazonwebservices.com/AWSEC2/latest/DeveloperGuide. 07.07.2010.

[19] Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, et al. Dynamo: Amazon's highly available key-value store. In Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles, ACM Press New York, NY, USA, 2007, pp. 205–220.

[20] Google App Engine, http://www.google.com/apps/intl/en/business/index.html. 07.07.2010.

[21] Desktop cloud computing services. http://www-935.ibm.com/services/us/index.wss/offering/eus/a1026737. 07.07.2010.

[22] VMWARE ThinApp——Agentless Application Virtualization Overview. White Paper. Available at http://www.vmware.com/files/pdf/thinapp_intro_whitepaper.pdf., 07.07.2010.

[23] Windows Live SkyDrive, http://skydrive.live.com/, 07.07.2010.

[24] Bowen Alpern, Joshua Auerbach, Vasanth Bala, Thomas Frauenhofer, Todd Mummert, and Michael Pigott. PDS: a virtual execution environment for software deployment. Proceedings of the First ACM/USENIX international conference on Virtual execution environments, March, 2005, pp. 175-185.

[25] http://www.microsoft.com/systemcenter/softgrid/default.mspx. 07.07.2010.

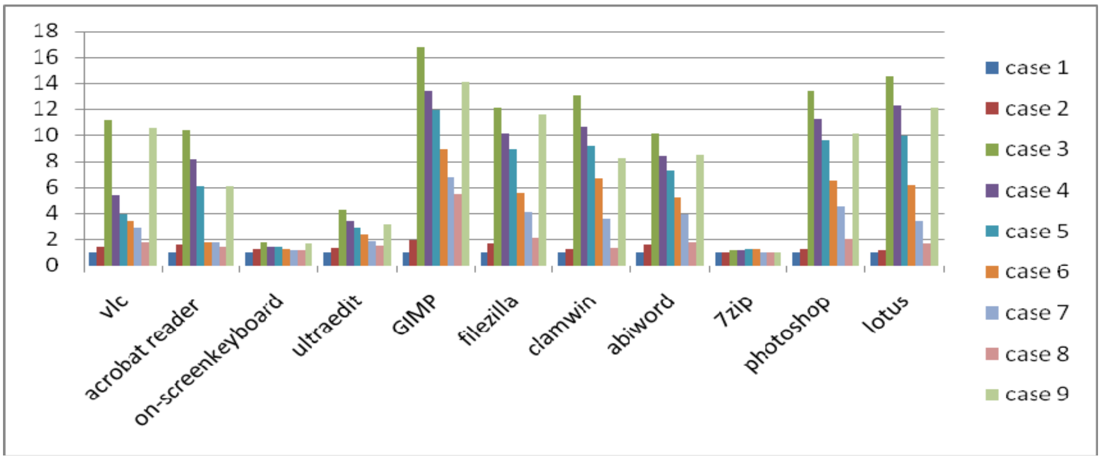


Figure 2. The start-up time (The server is inside and values have been normalized).

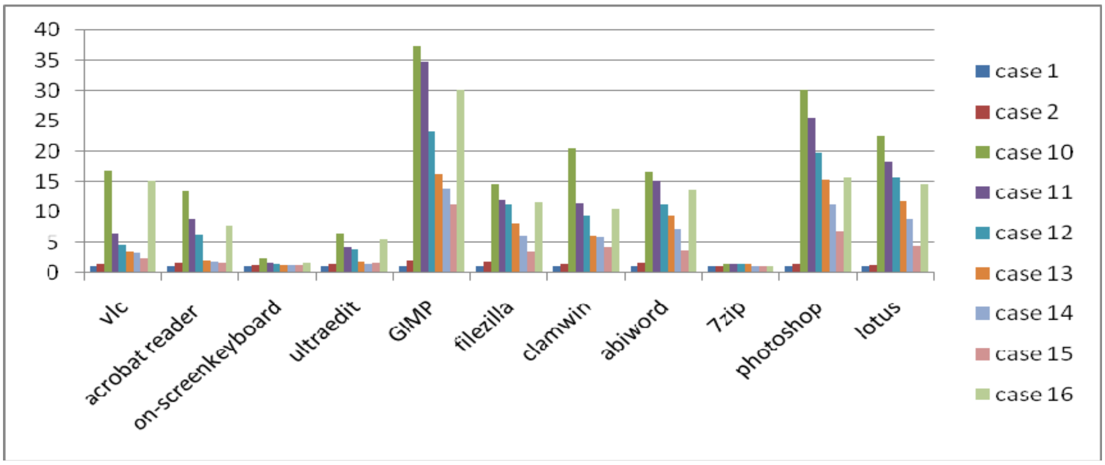


Figure 3. The start-up time (The server is outside and values have been normalized).

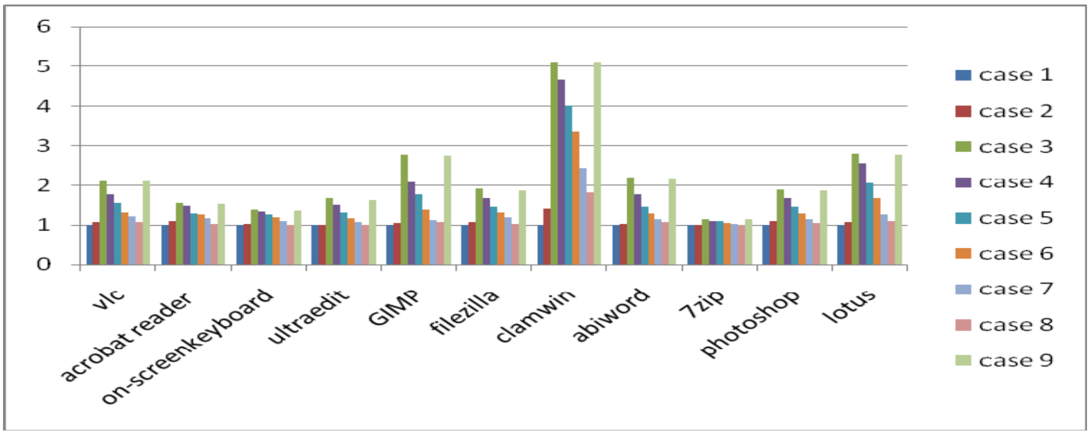


Figure 4. The run time (The server is inside and values have been normalized).

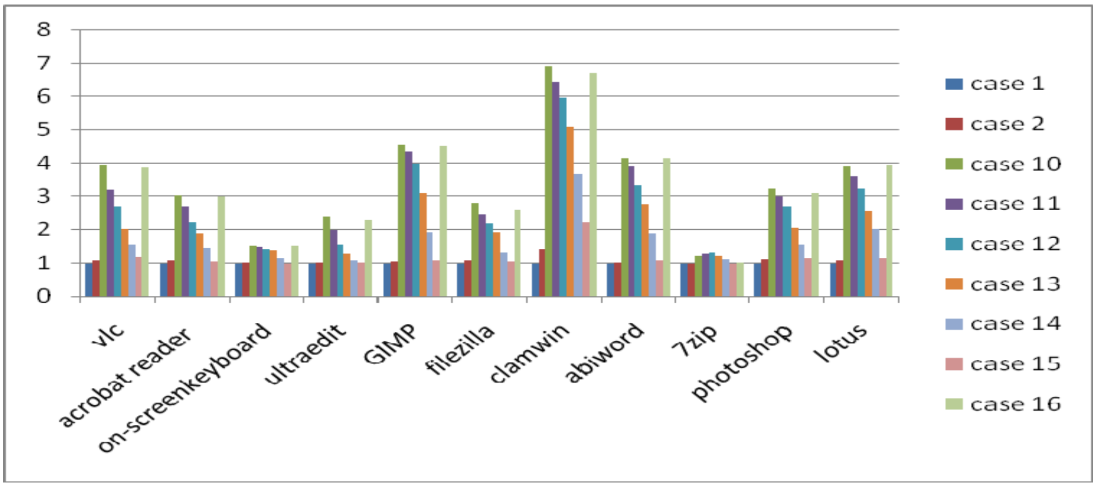


Figure 5. The run time (The server is outside and values have been normalized).

ISBN: 978-1-61208-101-4

Cloud Computing: Several Cloud-oriented Solutions

Amel Haji, Asma Ben Letaifa, Sami Tabbane
Higher School of Communication of Tunis, SUP'COM, Tunisia
7th November University at Carthage
{amel.haji | asma.benletaifa | sami.tabbane}@supcom.rnu.tn

***Abstract*— Cloud computing is known as an IT environment that includes all elements of the IT and network stack, enabling the development, delivery, and consumption of Cloud Services. In this work in progress paper, we present a brief introduction to the concept of cloud (types and services). Then we outline a state of the art of different existing solutions of cloud. A comparison tables are also proposed.**

Keywords- cloud computing; services; scalability; provisioning.

I. INTRODUCTION

The current network architectures make difficult reusability and increase costs. These architectures not take into account the changing functional requirements at the application development. Faced to costly development, redundant interconnections (point to point), a big complexity and difficulty to maintain, Coud computing and service oriented architecture are a very effective response to these issues in terms of reusability, interoperability and reduce coupling between different systems to ensure their cooperation. Cloud computing is the next generation platform that provides dynamic resources, virtualization and high availability. Cloud computing is not associated with a particular technology, protocol or provider. In practice, applications and data are no longer on the local computer but in a "cloud" composed by a number of remote and interconnected servers. Cloud computing describes a new supplement, consumption and delivery model for IT services based on Internet, and it typically involves the provision of dynamically scalable and often virtualized resources as a service.

Cloud computing offers: Ubiquitous network access, location independent resource pooling, rapid elasticity, self Service and Instant-On, elasticity and Pay-as-you [1].

This paper is divided into two sections. The first one presents types of cloud and models of services in these clouds. Then, the second section describes related works which exposes characteristics of some existing solutions of clouds and comparative tables of these solutions related to infrastructures, platforms and services.

II. CLOUD COMPUTING: TYPES & SERVICES

A. Types of clouds

Three types of cloud could be presented: Public clouds, Private clouds and hybrid clouds.
Public cloud: the services are delivered to the client via the Internet from a third party service provider
Private Cloud: these services are managed and provided within the organization. There is less restriction on network bandwidth, fewer security exposures and other legal requirements compared to the public Cloud.
Hybrid Cloud: there is a combination of services provided from public and private Clouds [2].

B. Models of Service

We find in literature everything as a Service (EaaS). EaaS is the concept of reusable component called across network. It's a subset of cloud computing. "as a Service" was been associated with others functions such as communication (CaaS) or data (DaaS). Three models of service are the most used.
Infrastructure as a Service (IaaS): This model is a modern form of utility computing and outsourcing. IaaS can manage computer resources (networking, storage, virtualized servers). This model allows consumers to deploy and manage assets or leased server instances, while the own service providers govern the underlying infrastructure.
Platform as a service (PaaS): It facilitates the development and deployment of applications without the management of the underlying infrastructure, by providing all necessary equipment to support the entire life cycle of construction and delivery of Web applications and services. This platform consists of software infrastructure, and typically includes a database, middleware and development tools. This type of service typically operates at a high level of abstraction. Users can manage and control resources that they deploy in these environments. Service providers maintain and govern application's environments, server instances, and the underlying infrastructure.
Software as a Service (SaaS): The hosted software or applications are consumed directly by users. Consumers control only the way in which they use cloud services while service providers maintain and manage software, data and the underlying infrastructure [3].

III. COMPARISION OF SEVERAL CLOUD'S SOLUTIONS

In this section, we describe some platforms of clouds and then we summarize in a comparative tables some characteristics of IaaS then PaaS [4] and SaaS as shown respectively in Table 1 and Table 2.

A. Eucalyptus

Eucalyptus for "Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems" is an open source software to implement Infrastructure as a Service in the cloud. The architecture of eucalyptus is simple, flexible and modular with a design hierarchy as shown in Figure 1.

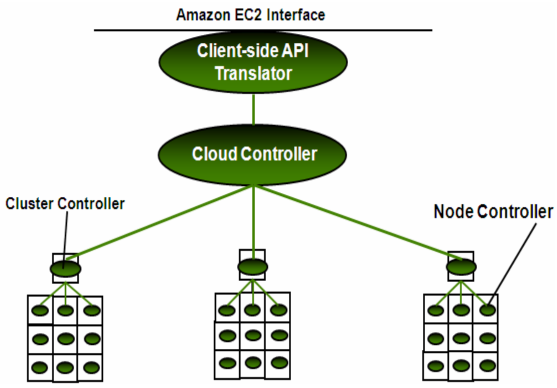


Figure 1. Eucalyptus's Architecture [5]

Three essential components of eucalyptus [6]:

Cloud controller queries information about resources from node managers, makes the scheduling decisions and executes them by using the cluster controllers.

Cluster Controller collects information about a set of virtual machines and schedules their execution on specified node controllers.

Node controller is running on each node that is designated to host virtual machine. It manages the implementation, inspection, and the termination of the VM on the host where it runs.

Users have the ability to execute and monitor virtual machines deployed throughout the physical resources in a flexible, portable, modular and easy manner. The design of eucalyptus gives users the flexibility to seamlessly move applications to Eucalyptus on-premise on the public cloud, and vice versa. Eucalyptus also makes easy the deployment on the hybrid cloud, using resources from public and private clouds for the unique advantages of each [6]. The disadvantage is the lack of an interface to manage virtual machine and an advanced monitoring.

B. OpenNebula

OpenNebula is an open source manager of virtual infrastructure [7], able to build private, public and hybrid clouds. OpenNebula offers flexible architecture, interfaces and components that could be integrated into any data center. This tool supports Xen [8] , KVM [9] and VMware [10] and access to Amazon EC2s [11].

OpenNebula was designed to be integrated into any network and storage solution. OpenNebula manages the storage, networking and virtualization technologies to enable the establishment of dynamic multi-level services (groups of interconnected virtual machines) on the distributed infrastructure, combining the resources of physical machines and cloud distance, based allocation policies

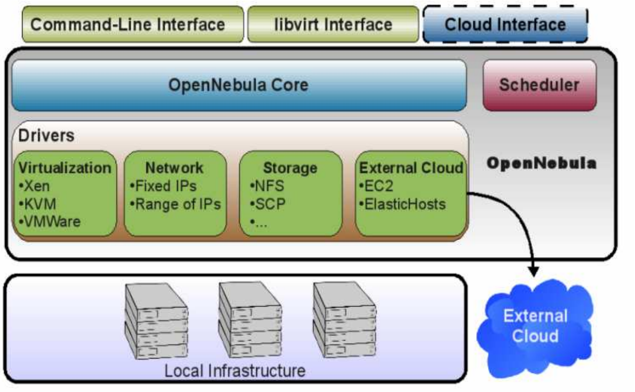


Figure 2. OpenNebula's architecture [12]

OpenNebula consists of three components [13]: *core (Virtual Infrastructure Manager*): Manages the lifecycle of the virtual machine by running the basic operations (deployment, monitoring, migration).

Capacity Manager (scheduler) module that governs the functionality provided by the core of OpenNebula: workloads balancing in virtual machines. *Virtual Access Drivers*: virtualization layer. The drawback of openNebula is the lack of GUI (Graphic User Interface)

OpenNebula provides the "load balancing" across nginx as shown in Figure 3.

Some advantages of OpenNebula [13]:

- Centralized management of the balance of the workload "load balancing", server Consolidation, resizing dynamic infrastructure, partitioning Dynamic Clustering, Support for heterogeneous workloads, and supply virtual machines on demand.

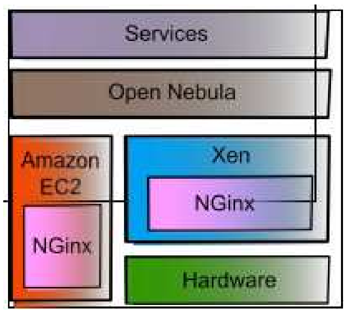


Figure 3. Load balancing in Open nebula [12]

ISBN: 978-1-61208-101-4

C. Nimbus

Nimbus is an open source toolkit that provides "infrastructure-as-a-Service». It allows a client to lease resources in distance by deploying virtual machines (VMs), for building a desired environment [14].

Nimbus Cloud Storage offers the model pay-as-you-go and scalability (scale up and scale down as needed without adding expensive infrastructure). It allows customers to reduce costs and eliminate the task of management, giving them the freedom to focus on their business.

Nimbus allows providers to build clouds: Private clouds (Workspace Service: Open source implementation EC2) and developers to experiment with clouds: research or the use / performance improvements and contributions. It requires certain dependencies are installed first. On the service node: Java (1.5 +) and bash. On the nodes of the hypervisor: Python, bash, ebtables, dhcpd, and KVM or Xen libvirt. It supports both interfaces EC2 (Elastic Computing Cloud) and WSRF (Web Service Resource Framework).

D. Abicloud

Abicloud is an open source infrastructure for building and managing public and private clouds based on heterogeneous environments. The tool offers users the ability primarily for scaling, managing, providing automatic and immediate servers and networks [15]. AbiCloud is auto scale: We can change the number of virtual servers, storage and memory. Therefore allows the platform to scale up or down as needed. The platform of AbiCloud is modular because it tries to improve the scalability of the system. The architecture is represented by the Figure 4.

abiCloud_Server: contains the business logic of the global platform of clouds and interacts with the database. *abiCloud_WS*: This virtual assembly line of the platform interacts with various virtualization technologies to manage virtual machines. *AbiCloud_VMS* (Virtual Monitor System) is the component developed to monitor the virtual infrastructure to learn about events or states. *AbiCloud Appliance Manager*: This component enables the management, distribution and scaling (scalability), allowing the import of external applications to the cloud platform.

This component is under development.

AbiCloud Storage Management: This component is currently being formulated and will be dedicated to the integration of storage platform systems. *abiCloud_client*: this web application RIA developed in Flex enables users to manage their private Cloud [15].

E. FlexiScale

The FlexiScale architecture is modular and can accommodate different implementations of its functionality. Virtual Iron is used which is built on the top of Xen Hypervisor and works as an external management layer for the virtual servers. FlexiScale is a Multi-tier architectures enabled by a high-speed internal multiple gigabit Ethernet network. It's a data center architecture which is designed to deliver a guaranteed QoS level for exported services. FlexiScale gives a pay-as-you-go virtual dedicated server. It offers Self-service provisioning of servers via API. Also, Additional servers can be launched in under a minute based on FlexiScale's operating system images or images created by user, highly automated and rapid provisioning of additional processing or storage resources [16].

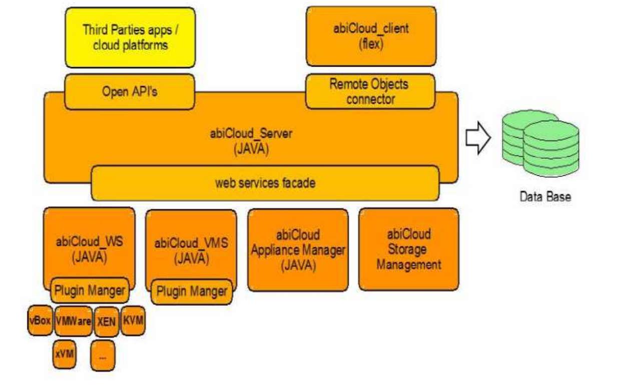


Figure 4. Abicloud's architecture [15]

F. Windows Azure

Windows Azure is Microsoft's offer on Cloud Computing. This is an application platform providing services, accommodation and administration tools. Windows Azure is an operating system for cloud services that serves the development, service hosting and service management environment for the Windows platform Azure [17]. Windows Azure provides developers ability to do computing and storage and also the management of Web applications on the Internet through on demand data center. Windows Azure is a flexible platform that supports multiple languages and could be integrate with the existing environment. In addition, Windows Azure supports popular standards and protocols, including SOAP, REST, XML and PHP. Accommodation Azure will provide a set of scalability features of operating on demand. It is thus possible to obtain and allocate additional processors if the scalability of an application requires it [4].

G. Google appEngine

AppEngine is intended solely for conventional web applications, the application is structured with a clear separation between the third load and the storage. In addition, AppEngine applications should be request-response. AppEngine provides automatic scaling and high availability. For example, AppEngine is not suited for general computing. He admits a fixed topology structure to accommodate the 3-tiers application [18].

H. Force.com

Force.com is a Multi-tenant architecture with metadata driven development model. It means that's a single instance of the hosted application which is able to serve all customers (tenants). Force.com is a SaaS service confined to API. It insures load balancing among tenants. It uses Apex language or database service and supports for .Net, C#, Apache Axis,…Concerning provisioning, with multi-tenant environment, create development, test, staging, training, and production environments are quickly, easily and cost-effectively (in a single tenant environment every new stack must be separately provisioned, managed, and scaled) [19].

IV. Conclusion and Future Works

In this paper, we introduced cloud computing overview. We presented layers of clouds, actors and its different types. Then, we summarized some current cloud solutions. Others cloud solutions like IBM solution or EmotiveCloud, Claudia, could be studied in the future in addition to IaaS frameworks. In future works, we will also try to focus on the concept of cloud provisioning and study how Service Oriented Architecture could helps us to perform cloud provisioning in scalable manner.

References

[1] DSP ipFast Forward your Development, "Introduction to cloud computing".

[2] Steve Bennett, Mans Bhuller, and Robert Covington, "Architectural Strategies for Cloud Computing", Oracle corporation, 2009.

[3] David Chou, "Understanding Cloud Computing and Cloud-Based Security", SOA Magazine, 2010.

[4] Michael Armbrust et al."Above the Clouds: A Berkeley View of Cloud Computing", Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009.

[5] Rich Wolski, "EUCALYPTUS: An Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems", Computer Science Department, University of California, Santa Barbara, 2008.

[6] Eucalyptus systems, "Eucalyptus Open-Source Cloud Computing Infrastructure - An Overview", August 2009.

[7] www.opennebula.org , April 2010.

[8] http://www.xen.org/ , September 2010.

[9] http://www.unixgarden.com/index.php/administration-systeme/virtualisationkvm-acpi-et-sysfs, September 2010.

[10] http://www.vmware.com/fr/, September 2010.

[11] http://aws.amazon.com/ec2/, September2010.

[12] Constantino Vázquez Blanco, "The OpenNebula Virtual Infrastructure Engine", Distributed Systems Architecture Research Group, Universidad Complutense de Madrid, June 27, 2009.

[13] Borja Sotomayor, Rubèn Santiago Montero1, Ignacio Martin Llorente1, and Ian Foster, « capacity leasing in cloud Systems using the Open Nebula Engine », Facultad de Informaticà, Universidad Complutense de Madrid.

[14] http://www.nimbusproject.org/docs/current/faq.html#nimbus-main-components, April 2010.

[15] http://www.abicloud.org/display/abiCloud/Home ABICLOUD TECHNICAL OVERVIEW,2009.

[16] www.flexiScale.com, September 2010.

[17] Nicolas CLERC , " Windows Azure – Présentation technologique, Expertise des ecosystems".

[18] http://code.google.com/appengine, August 2010.

[19] www.salesforce.com , June 2010.

TABLE I. COMPARAISON OF SEVERAL IAAS CLOUD'S SOLUTIONS

	Eucalyptus	OpenNebula	Nimbus	AbiCloud	FlexiScale
Service	IaaS	IaaS	IaaS	IaaS	IaaS
Cloud's type	Public / Private	Private	Public	Public/ Private	Public
Scalability	Not Scalable	Scalable	Scalable	Scalable	Scalable
Compatibility	Not support EC2	Multi-platform	Support EC2, WRSF	Support EC2	Not support EC2
VM support	VMware, Xen, KVM	Xen, VMware	Xen	virtualBox, Xen, VMware	Xen Hypervisor
Structure	Module	Module	Component	Module	Module
Provisioning Model	Immediate	Best effort+ haizea: advance reservation, immediate, best-effort + reservoir: Immediate, Best-effort	Immediate	Immediate	Self provisioning
Load balancing	Simple load balancing cloud controller	Nginx as load balancing	auto configuration of virtual clusters s		Automatic with cluster

TABLE II. COMPARAISON OF PAAS AND SAAS CLOUD'S SOLUTIONS

	Azure	Appengine	Force.com
Service	PaaS	PaaS	SaaS
Scalability	Scalable	Scalable	Scalable
VM support	Xen Hypervisor	Multitenant architecture	Multitenant Architecture
Provisioning Model	-	-	Immediate
Load balancing	Should install software	Automatic	Load balancing among tenants
Storage model	SQL Data Services Azure storage service	MegaStore/BigTable	-
Networking Model	Automatic based on programmer's declarative descriptions of app components	Fixed topology to accommodate 3-tier Web app structure	-

From Grids to Clouds: A Collective Intelligence Study for Inter-cooperated Infrastructures

Stelios Sotiriadis, Nik Bessis, Paul Sant, Carsten Maple
Department of Computer Science and Technology
University of Bedfordshire, United Kingdom
(stelios.sotiriadis, nik.bessis, paul.sant, carsten.maple)@beds.ac.uk

Abstract – **Recently, more effort has been put into developing interoperable and distributed environments that offer users exceptional opportunities for utilizing resources over the internet. By utilising grids and clouds, resource consumers and providers, they gain significant benefits by either using or purchasing the computer processing capacities and the information provided by data centres. On the other hand, the collective intelligence paradigm is characterized as group based intelligence that emerges from the collaboration of many individuals, who in turn, define a coordinated knowledge model. It is envisaged that such a knowledge model could be of significant advantage if it is incorporated within the grid and cloud community. The dynamic load and access balancing of the grid and cloud data centres and the collective intelligence provides multiple opportunities, involving resource provisioning and development of scalable and heterogeneous applications. The contribution of this paper is that by utilizing grid and cloud resources, internal information stored within a public profile of each participant, resource providers as well as consumers, can lead to an effective mobilization of improved skills of members. We aim to unify the grid and cloud functionality as consumable computational power, for a) discussing the supreme advantages of such on-line resource utilization and provisioning models and b) analyzing the impact of the collective intelligence in the future trends of the aforementioned technologies.**

Keywords – Grid computing; Cloud computing; Collective Intelligence; Mobility Agents

I. INTRODUCTION

Grid computing is defined as the combination of several distributed resources from multiple Virtual Organisations (VOs) for solving a single problem, which is usually a scientific or technical problem [13]. A VO is a group of members whose resources function as a unit. This form of distributed computing tends to be distinguished from the conventional systems, by offering a heterogeneous environment of loosely coupled connections. On the other hand, over recent years the notion of clouds has proven to be a model that has had significant commercial success [15]. The commercialized distributed resources are spread across the internet and are available for purchase from users by offering capabilities of resource management and provisioning. It may be noted that the concepts behind grid and clouds can be described from a technological perspective as a novel way for a) achieving inter-collaboration among several users or distributed resources (from the grid viewpoint) and b) a high quality and on-demand resource provisioning model involving various stakeholders (from the cloud viewpoint). In other words, clouds can utilise enterprise resources by serving multiple users across an inter-cooperated grid environment. Fundamentally, the service-oriented infrastructures of both technologies aim to provide a virtualization model of entities as services and the seamless interactions and integration of these services. In terms of virtualization we define everything that can be virtualized in any environment separated by the underlying location and spread across the internet. In general, cloud technology is derived from grid, virtualization, and utility computing [14], [18]. Utility computing is a priced service of server's capacity that is accessed over the grid [15]. In any case, all the aforementioned technologies collaborate with each other, aiming to offer a user resource over the internet.

One of the major advantages of on-demand technologies, e.g., cloud, utility and virtualized computing is that by a subscribing cost users don't have to deal with hardware and software licences, versions, incompatibilities, failures and maintenance. On the other hand, the complex requirements of users, which include subscription cost versus the usage, are drawbacks of clouds. In general, several resources including hardware and software can be delivered to users with different needs. The clients that utilize the resources may be categorized as uncomplicated consumers, business and enterprises as well as multi-tenancy users. So, the different requirements raise several complications and complexities. A simple example is that, companies are reluctant that their data are outside their firewall. From the user perspective, cloud participants want to be charged only for the amount of resources that they use. However, cloud and grid, have proven to be secure, reliable and scalable and the resources which are put to use according to actual requirements have an efficient load balancing feedback mechanism [18]. In the case of uncertain or pre-specified boundaries the complexity of users' requirements may be approached by a more generalised model [1]. It is apparent that the actual capabilities of several resources are dependent on how grids or clouds are employed. Since clouds are related to usage concepts and grid is relative with technological concepts, the collective intelligence goes one step further with on-demand resource provisioning. This may eliminate and remove the need to over-provision in order to meet the demands of millions of users by evaluating their knowledge.

The collective intelligence is a paradigm suggesting a new source of empowerment by monitoring the exchanged intelligence among cloud and grid users. This is when data, software applications or computer processing power are accessed by a cloud of online resources and can be reused to support decision making and team building [17]. The digital communication and sharing of data can be collected and be analysed by a perception model. The collective learning may offer (in the future) grids and clouds that employ a creative method by evaluating current data and offering new knowledge to a neighbourhood of users. The inter-cooperation model among unknown members may significantly improve the production of collective intelligence knowledge, and especially in a competitive and sharp environment.

In the following sections, we discuss the motivation of the study (Section 2), and the related work and definition of similar technologies (Section 3). We then continue by introducing the inter-cooperation model of grids and clouds and the collective intelligence application (Section 4, 5). We then use this to discuss a case study of clouds and grids as a means of forming the collective intelligence method (Section 6). Finally, we conclude our study with the future work section and the proposed challenges part (Section 7).

II. Motivation

Various approaches and definitions of clouds exist [14, 18], all conclude that cloud is comprised of grid, virtualization and utility computing notions [13]. Each of these technologies may be seen as a set of layers that encompass the cloud, which could exploit user behaviour and draw collective intelligence. In a broader view, a cloud can be seen as a customized grid. The members forming the cloud can access resources, solve problems similar to grids but in a more structured, scalable and personalised management manner; as well as by charging a subscription cost. A grid VO may offer the cloud a geographically distributed environment formed under a common policy management scheme either centralized, or de-centralized. Consequently, VO members may utilize resources to solve VO defined problems.

It is common that within a VO, members or resources perform interactions based on two different perspectives. Firstly, the centralized management system monitors the procedure and is responsible for the service negotiation. Secondly, the decentralized control system of autonomous acting VO members. A typical VO will have access to many facilities which are not owned and managed by the VO [13]. These facilities may be multi-participant communities, resources or VOs. Mutually, in a VO, a universally agreed model of policies has been adopted by each individual, and determines the accessibility factor of each resource.

From the viewpoint of the decentralized solution, the mobility agents' paradigm offers a novel technology of achieving communication among loosely coupled connections of multi-institutional VOs [4]. By considering the grid as a coordinated problem solving approach in dynamic environments, agents may be the means of acting dynamically and autonomously whilst performing migration to any inter-connected member [10]. Moreover, collected knowledge among cloud or grid members may be organised and shared by utilizing the intelligent agent model. The essence of the aforementioned standard may be achieved by using the framework of *Self-led Critical Friends* (SCF) which fills the gap between consumer's providers and enables inter-operation of nodes from various grid VOs [6], [7]. The ultimate goal is to extend the conventional grid's bounded VO topology to a wider dynamic community established upon SCFs knowledge of members from different grid and clouds domains also known as *Critical Friends Community* members.

Originally, the SCFs act as intermediate stations in the communication between multiple grids by providing an extended environment. SCFs may act among a cloud of users, so the collective intelligence may be extended. Essentially, an expanded intelligent agent takes information about their interest, problems and behaviour and recommends to users, and companies, solutions and information for improving services as well as proposing new forms of social applications. As typical to Web 2.0 applications an online community' interests and activities can be captured by software APIs of any device and give profit to stakeholders; even if they are providers, resellers, adopters or users [8]. The SCFs could fulfil the gap among resource consumer and providers by expanding the community boundaries.

III. Related Technologies and Definitions

The related technologies are slightly different in the use of terminology and it is hard to be distinguished as they usually offer the same type of services [14]. For that reason, we suggest that shared resources should be separated in the following ways according to their service orientation:

a) The Software as a Service (SaaS) framework provides specific cloud capabilities concerning software functionalities available through the internet. Each individual can access the applications from any device [18]. Examples include commercial SaaS, project online tools and customer relationship management tools.

b) The Hardware as a Service (HaaS) framework, also known as infrastructure as a service (IaaS) provides computational power and resources [18]. This type of cloud is similar to the virtualization environments in which applications are separated from infrastructures and computer capacity is shared over the web.

Similarly, service orientation clouds can be also separated according to their deployment orientation [18]. Firstly, there are private and public clouds whose functionality is to behave as intra and inter collaborated environments. In both scenarios, clouds are similar to private and public VOs. It is also essential that specific measures and policies are applied by a VO management control system. Secondly, there are community clouds which are typically similar to grid technology, and can aggregate public clouds or dedicated resource infrastructures.

We can hence separate clouds according to their service or deployment orientation; however, the data centres which initiate cloud capabilities may consider, from either perspective, that a collective intelligence concept can be utilized. It should be mentioned that members of grid or clouds can behave as resource owners as well as resource consumers, so they have to respect policies and credentials of VOs or VO members. From the view of a resource consumer, their requirements may be organized as follows [16]:

a) May belong to one or more VOs or clouds
b) May have several roles and actions
c) May control internal roles, actions and memberships
d) May assign priorities to members jobs
e) May list resources and internal knowledge in a metadata snapshot profile
f) May assign special credentials to specific members
g) May enable or not authentication to other members
h) May select or deselect roles and actions

On the other hand the requirements from the perspective of the resource providers are organised as follows:

a) VOs or clouds should control resource participation
b) VOs or clouds should specify their own policies and resource authentication
c) VO or clouds should have a consistent authorization process
d) VO or clouds must be able to specify requirements on any resource for specific roles.
e) Within a VO or a cloud a common policy model should be agreed

Not all of the above requirements are necessary attributes of grid or cloud members. However, it is the basis for a collective intelligence model of policies and strong communication links. Resource providers and consumers should support such infrastructures aiming to improve performance and profit.

Overall, grid and cloud environments provide a way to implement future applications about monitoring knowledge and improving software intelligence. It is evident that the analysis of requirements is a difficult aspect of such designs. Finally, an intelligent agent model may assist in the consolidation of a requirement scheme, based in a secure and elastic scalability model. In order to organize such a strategic model, it is crucial to identify the accessibility factor of each VO. In this direction, we suggest that it is necessary to store agreement protocols within each member's public profile. We define this storage as the *metadata snapshot profile*, which is available upon request from any member. The profile stores data about a member's potential. In other words, it is a unique intelligence storage place for each individual. By viewing it as such, we can construct several metadata snapshot profiles, and realize a collective intelligent model of inter-cooperating members.

IV. CO-OPERATING GRIDS AND CLOUDS

It has been proposed that "*the real and specific problem underlies the grid concept is coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations*" [10]. In other words, they suggest that resource sharing must be synchronized clearly negotiated and defined among resource owners, consumers and providers. Finally, the multi-institutional statement refers to a collaborating environment of several VOs.

It is very important that the resource discovery method of resources fulfils the needs of resource consumers and providers. The resource discovery method in such uncertain environments starts when a member requests information from the metadata snapshot profile of any connected grid or cloud participant. The profile contains various data but initially we are seeking for the addresses of well known and trusted nodes. In figure 1 we assume that ***n_1*** can access ***n_2*** as well as any other member of VO_1. However ***n_2*** contains a new address of the related (and inter-connected) member ***b_1***, so n_2 assigns a reference contract to the SCF contract and updates the profile of ***n_1***. At this stage ***n_1*** is capable of establishing a connection with ***b_1***. However ***n_1*** then requests information from ***b_1*** as they are both trusted members so they follow the same procedure and the new updated profile of ***n_1*** contains the addresses of all members of VO_2. The procedure continues and all members from both VOs have access to any resource available to the mutually interconnected VOs.

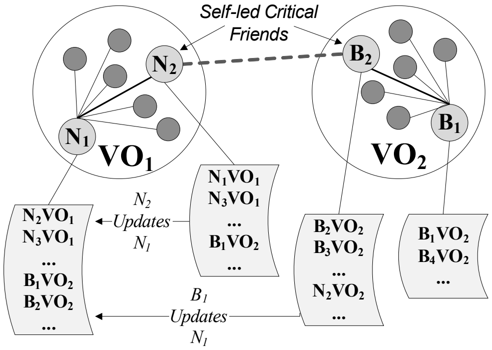


Figure 1: The SCFs communication model

In such situations mobile agent functionality may serve the aforementioned model by traversing a specific route. By moving from one location to another it visits all individuals of a VO. Each time an agent moves to a specific location it carries internal data about physical resources and internal knowledge. During their route traversal agents are capable of visiting different platforms and by collecting and updating internal data as they continue the journey. In other words, the collective intelligence carries the internal data from member to member as a list of addresses, capacities and information such as beliefs and desires. This has the potential for stakeholders, including providers, resellers, and users to utilize a users' demands and requests to identify solutions for well defined problems in a fast and reliable manner.

V. Collective Intelligence

The goal of collective intelligence is to harness the system of self-centred grids, clouds and agents to secure a sustainable relationship, so that coordinated individuals may solve problems more efficiently [5], [9]. In general, collective intelligence of unified and synchronized grid and cloud communities can offer significant advantages. A clear example in nature is the ants; an individual ant is not very powerful, but a colony of ants can achieve significant results. Collective intelligence can be found in many systems, and it is known as swarm intelligence, ant colony optimization and neural networks. So, we may describe such a method as collective intelligence, which can be seen as an infrastructure or environment in which individuals can do simple operations, however by working they are able to perform and solve complex problems. It is almost certain, that the complexity of the aforementioned environments is high as they are formed from different resource consumers and providers connected in loosely coupled groups. In our vision, intelligent agents fulfil the gaps of security and transparency in grid or cloud members' communication. So, their characteristics are that they:

a) Present a digital community with information exchange capability as well as purchasing, selling, storing, transmitting and processing means
b) Collect information about other member's potentials
c) They categorise facts, comments and opinions of members
d) Offer a secure sustainable relationship among resource providers and consumers
e) Migrate to any members' device by moving a part of their code

Inside a grid or cloud community the need for reinforcing a collective intelligence model may be prove to be significantly profitable for all the participant groups. This is the case when the individuals' contribution to the grid or cloud is equal. However, a division of labour model can be applied to these communities as each member may have a different domain of specialisation. In other words, limitations of one member may be satisfied from any other member. Since not everybody can perform all tasks, a group where different individuals contain different knowledge will collectively cover a much larger domain.

VI. Case study: An Inter-Cooperated Crowd-y Cloud

By crowd-y cloud we define a community aware cloud in which users' perceptions and desires are captured from artificially intelligent agents. The crowd contains all kinds of devices that can access the grid or cloud and can utilize resources for any purpose [21]. In our view the intelligent agent model can serve the aforementioned vision as a means of achieving decentralization of grid and cloud VOs.

The Foundation for Intelligent Physical Agents (FIPA) provides a framework for an interoperable agent solution that can be used for the development of co-operating agent systems [19]. The agent service of the above standard provides an environment for organising the procedure of an agent travelling within unknown large scale domains with dynamic behaviour. We have proposed an approach that separates each mobile grid service into static and mobile parts that can dynamically migrate across grid nodes is presented in [11]. The specific design based on the FIPA agent management specification is the following:

a) Agent Management System as the place to create the agent life-cycle
b) Directory facilitator as a yellow page directory of the well known and trusted members addresses
c) Agent Container Service which provides the run-time environment of agents
d) Message Transport System which constitutes the communication bus among the agent platforms

In a previous work [2] we have discussed the resource discovery methods of interoperable grid agents based on the above FIPA specification. We analyzed existing resource discovery methods of agents and proposed a new solution for inter-collaborated agents [1]. In this paper we suggested that resource discovery is a systematic and continually updating process occurring directly within a VO. Finally, we concluded that solution of discovery includes either internal broadcasting agents, or internal travelling agents as a decomposition model of local agents in order to achieve an efficient and effective low response time.

In the following section we describe commercial efforts of the crowd-y clouds. In this way, we aim to utilize the intelligent agents' paradigm for delivering high quality applications, which implement major parts and concepts of grid and clouds. It is essential that we use specific middleware based on the Java Agent Development Framework (JADE) and that it is installed on all cloud devices [12]. In our case study, the grid technology serves a cloud by providing the problem solving environment, and the cloud provides the personalised view of the problem description. Notably, the underlying infrastructure maintains standard policies and securities articulated from the VO. In previous works we have defined the minimum requirements that need to be addressed and supported by grid members [1], [3]. We have organised the profile information to include Policy Management Control for identifying the level of agreed protocols for communication between different parties and addresses of trusted members. Then Knowledge Base Pairing is used as the procedure for job description coupling and the Physical Resources Announcement provides the mechanism for advertising internal hardware and software capabilities. Finally, Time Constraints for storing historical data about execution and communication times from previous delegations are also utilised. It is vital to acknowledge policies among different VO parties as well as respecting internal VO rules and actions.

In our study we aim to improve the Knowledge Base Pairing for collaborating members of a crowd-y cloud. More specifically, the agents travel through the cloud and by collecting information from each member's public profile, return back to their starting point. This position may be a stakeholder, and may be either the resource provider or the resource consumer. Our Example (Figure 2) we demonstrate the cloud of members, including providers and consumers. It should be mentioned that we decompose the cloud into smaller neighbourhoods that act as VOs. In addition the SCF offers a novel resource discovery method by acting a mediator in communication between the loosely coupled members.

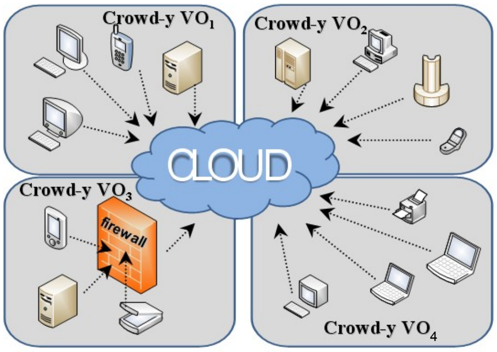


Figure 2: The crowd-y cloud of users

Each resource of the aforementioned figure contains the agent middleware for creating and destroying agents. Our example (Figure 3) consists of three members that are able to communicate with each other. A node is selected to be the host, which in our case will be able to create the agent service. The remaining nodes are capable of creating a sub-platform specification which refers to the Host member platform. The platforms consist of containers which could be scattered among many different hosts with one main container on a host running the Remote Method Invocation (RMI) service. All the agents on one platform communicate using the RMI protocol which is the intra communication mechanism internal to a platform. In other words, sub-platforms accept communication from an agent ***A***, while an internal agent waits for the connection. The agent starts from the host platform and by traversing a route to each node collects and updates the internal information and then returns back to the host.

The service can be repeated with any of the other members, however each time other parties should be alerted of the service creator address. On the other hand, inter-platform agents offer a decentralized model which creates platforms dynamically for each individual without having to know the other members platform settings. In such environments each VO member contains a different platform which is created locally and generates an agent. Agent functionality involves waiting for requests from other agent platforms in order to exchange internal knowledge. In other words, any of the agents are capable of performing communication directly so a new service can be created dynamically. Figure 3 illustrates this procedure.

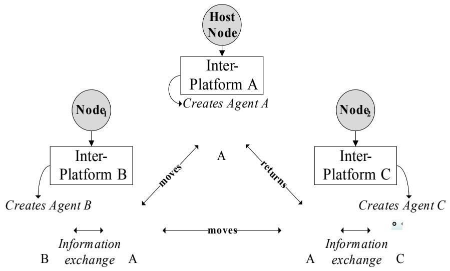


Figure 3: The inter-platform agent communication

The case discussed above illustrates the inter-platform communication model; in which agents are dynamically created by the inter-platform utility. The mobile agent migrates to a different platform and exchanges information with local agents. This solution predisposes the need for compatibility between different platforms and security issues are resolved by the agents. The economic gains of such an application to resource stakeholders are huge if we consider that these resource clouds may be transparent to the user and reliable to the provider. The virtualisation of the computational power, in conjunction with the heterogeneous nature of the grid, creates a scalable and elastic environment. Overall, the crowd-y cloud may address similar issues within any cloud system, but more importantly, we aim for a collective intelligence model of agents rather than a cloud of members.

The major benefits of a crowd-y cloud can be described from two perspectives; the resource consumer and the resource owner. We assume that each member contains a sensor API which may receive users' perceptions. These perceptions include noise levels, allergies, diseases and air pollution from the device holder environment, for example. The collected data may be transferred from member to member instantly as a form of resource information and may suggest, to users, statistics about low, medium, good and high quality for a particular locations or set of locations. By getting evidence from the environment as well as utilizing the grid functionality we may improve disaster relief, and in general generate environmental reports about specific geographical locations. Furthermore, policy makers may categorise the leasing opportunities for buying or selling properties according to noise levels. Also, governments may generate reports and warnings for health issues and welfare levels. In other words, future trends and technological possibilities are derived from the collaboration of technologies such as cloud, grid and virtualization, aiming to deliver a personalised product to users. By organising this collective intelligence we aim to propose that a larger collection of members are smarter than an elite few at solving problems, fostering innovation, coming to wise decisions, and predicting the future. The knowledge which is distributed everywhere can always be promoted, cultivated and improved. This could lead to an effective mobilization of skills. These skills are collected from an agent and stored within the metadata snapshot profile of a member.

VII. Conclusion And Future Work

In our study we have discussed a sufficient collaboration opportunity among grid and cloud computing, aiming to provide a collective intelligence model. The grid consists of VO members utilizing resources in order to solve VO defined problems but clouds are about users utilizing these resources to solve problems and proposing solutions. The future trend of these technologies is "reducing the carbon footprint" [18] so it makes them friendlier to the environment, and is also known as Green IT. We may consolidate numerous users in this direction by presenting the crowd-y cloud idea as an inter-collaboration environment with extensive capabilities. The market players should be ready for this step forward in order to improve their business by overcoming the obstacle of user requirements complexity. In this direction the agents based model may assist the resource provisioning model in a very proficient manner.

References

[1] Sotiriadis S., Bessis N., Huang Y., Sant P., and Maple C., "Defining minimum requirements of inter-collaborated nodes by measuring the heaviness of node interactions". In: *International Conference on Complex, Intelligent and Software Intensive Systems* (CISIS 2010), IEEE, Krakow, Poland, February 2010.

[2] Sotiriadis S., Bessis N., Huang Y., Sant P., and Maple C., "Towards to decentralized grid agent models for continuous resource discovery of interoperable grid Virtual Organizations", In: *The third international conference of Applications and Digital information and Web technologies* (ICADIWT), Instabul, Turkey, July 2010

[3] Sotiriadis S., Bessis N., Sant P., and Maple C., "Encoding minimum requirements of ad hoc inter-connected grid virtual organisations using a genetic algorithm infrastructure". In: *IADIS multi conference on computer science and information Systems* (MCCSIS 2010), Freiburg, Germany, July 2010.

[4] Sotiriadis S., Bessis N., Sant P., Maple C., "A mobile agent strategy for grid interoperable virtual organisations". In: *IADIS multi conference on computer science and information Systems* (MCCSIS 2010), Freiburg, Germany, July 2010.

[5] Mc Evoy, G. V., and Schulze, B., "Using clouds to address grid limitations". In: *Proceedings of the 6th international Workshop on Middleware For Grid Computing* (Leuven, Belgium, December 01 - 05, 2008). MGC '08. ACM, New York, NY, 1-6.

[6] Huang Y., Bessis N., Brocco A., Sotiriadis S., Courant M., Kuonen P., and Hisbrunner B., "Towards an integrated vision across inter-cooperative grid virtual organizations". In: *Future Generation Information Technology* (FGIT 2009), pp.120-128, Springer LNCS, Jeju island, Korea, 2009.

[7] Huang Y., Bessis N., Kuonen P., Brocco A., Courant M., and Hirsbrunner B., "Using Metadata Snapshots for Extending Ant-based Resource Discovery Functionality in Inter-cooperative grid Communities". In: *International Conference on Evolving Internet* (INTERNET 2009), IEEE, Cannes/La Bocca, France, August 2009.

[8] Hintikka, K. A.. "Web 2.0 and the collective intelligence". In: *Proceedings of the 12th international Conference on Entertainment and Media in the Ubiquitous Era,* Tampere, Finland, October 07 - 09, 2008). MindTrek '08. ACM, New York, NY, pp. 163-166

[9] Weiss, A. "The power of collective intelligence". *netWorker* 9, 3, September, 2005, pp. 16-23.

[10] Foster, I., Kesselman, C., and Tuecke, S., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", In: *International Journal of High Performance Computing Applications,* 15, 3, 2001, pp. 200-222

[11] Athanaileas, T. E., Tselikas, N. D., Tsoulos, G. V., and Kaklamani, D. I. 2007. "An agent-based framework for integrating mobility into grid services". In: *Proceedings of the 1st international Conference on Mobile Wireless Middleware, Operating Systems, and Applications* (Innsbruck, Austria, February 13 - 15, 2008). MOBILWARE, vol. 278. ICST (Institute for Computer Sciences Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, pp. 1-6.

[12] Bellifemine, F., Caire, G., Poggi, A., and Rimassa, G., "JADE: A software framework for developing multi-agent applications", In: *Lecture Notes in Computer Science, Intelligent Agents VII Agent Theories Architectures and Languages*, pp. 42-47. Springer Berlin / Heidelberg, 2001

[13] Winton, L. J. A simple virtual organisation model and practical implementation. In: *Proceedings of the 2005 Australasian Workshop on Grid Computing and E-Research - Volume 44* (Newcastle, New South Wales, Australia). R. Buyya, P. Coddington, P. Montague, R. Safavi-Naini, N. Sheppard, and A. Wendelborn, Eds. Conferences in Research and Practice in Information Technology Series, vol. 108. Australian Computer Society, Darlinghurst, Australia, pp. 57-65, 2005

[14] Pokharel, M. and Park, J. S. 2009. Cloud computing: future solution for e-governance. In: *Proceedings of the 3rd international Conference on theory and Practice of Electronic Governance* (Bogota, Colombia, November 10 - 13, 2009). ICEGOV '09, vol. 322. ACM, New York, NY, 409-410.

[15] EU DataGrid WP6, "Data Grid, EDG Users. Guide", October, 2003, Available at: http://marianne.in2p3.fr/datagrid/documentation /EDG-Users-Guide-2.0.pdf, Accessed at 10/10/2010

[16] Gregg, D. G, "Designing for collective intelligence". In: *Commun. ACM* 53, 4, April, 2010, pp.134-138.

[17] Schubertt, L., Jeffery, K., and Neidecker-Lutz, B., Expert Group Report, *The future of Cloud Computing: Opportunities for European cloud computing beyond 2010*, European commision, Belgium, 2010, Available at: http://cordis.europa.eu/fp7/ict/ssai/docs/cloud-report-final.pdf, Accessed at: 10/10/2010.

[18] The Foundation for Intelligent Physical Agents (FIPA), Available at: http://www.fipa.org, Accessed at 10/10/2010

[19] Bessis, N. (2010), 'Using Next Generation Grid Technologies for Advancing Virtual Organizations', Keynote Talk, In: *International Conference on Complex, Intelligent and Software Intensive Systems* (CISIS 2010), 15-18, February, 2010, Krakow, Poland

Accelerating Data-Intensive Applications: A Cloud Computing Approach to Parallel Image Pattern Recognition Tasks

Liangxiu Han, Tantana Saengngam, and Jano van Hemert

UK National e-Science Centre, School of informatics, University of Edinburgh, United Kingdom
liangxiu.han@ed.ac.uk; ttntans@gmail.com; j.vanhemert@ed.ac.uk

Abstract — **Performance is an open issue in data intensive applications, such as image pattern recognition tasks. To process large-scale datasets with high performance more resources and reliable infrastructures are required for spreading the data and running the applications across multiple machines in parallel. The current use of parallelism in high performance computing and with multicore hardware support is costly and time consuming. To remove the burden of building, operating and maintaining expensive physical resources and infrastructures, Cloud computing is emerging as a cost-effective solution to address the increased demand for distributed data, computing resources and services. In this paper, we explore and evaluate parallel processing performance of an image pattern recognition task in the Life Sciences based on a Cloud computing model: Infrastructure-as-a-Service. Namely, we rent computing infrastructures from cloud providers. We have developed the image pattern recognition task in both sequential and parallel ways, deployed them, and conducted our experiments on cloud infrastructure. The performance has been evaluated using speedup as a measurement. We have calculated the cost of our experiments, which demonstrates that cloud computing could be a cheaper alternative to supercomputers and clusters given this task.**

Keywords — Parallel Computing; Cloud Computing; Image Pattern Recognition; Life Sciences; Data intensive application.

I. INTRODUCTION

Advances in storage, pervasive computing, digital sensors, digital libraries and instrumentation have led to a massive growth in the volume of data collected and the number of geographically distributed data sources (e.g., in many fields, biomedical research or image-based diagnosis, data volumes are at least doubling each year). The efficient exploration on these large amounts of data is a critical task to enable scientists to gain new insights. Parallel computing is naturally as a solution to solve this kind of problems by dividing a large problem into smaller ones carrying out much small calculation concurrently. The current parallel processing systems are mainly supported by hardware with multi-core and multi-processors with multiple processing elements within a single machine and/or clusters and grids with multiple machines connected together to work on the same task simultaneously.

To deal with large datasets, more compute resources are required to access large amounts of data and perform many calculations across multiple machines concurrently. However, incorporating data and compute resources into parallel infrastructures (e.g., clusters) amenable to data exploration is not an easy task. One has to take into account scalability, reliability, fault-tolerance and cost-reduction.

To remove the burden of building, operating and maintaining expensive physical resources and infrastructures (e.g., hardware, clusters etc.), cloud computing is emerging as a cost-effective solution to address the increased demand for distributed data, computing resources and services.

Cloud computing [1][2] is a type of distributed computing paradigm augmented with a business model via a Service Level Agreement between providers and consumers. This definition has a two-fold meaning: 1) at a technical level, cloud computing offers distributed resources, infrastructures and services over the Internet to users, which are created, operated and maintained by the cloud providers. The consumers access these resources remotely without running applications on local computers. Cloud computing models are broadly classified into service and delivery models. In terms of services provided by cloud providers, there are three types of service models: Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS) and Infrastructure-as-a-Service (IaaS). Based on the way clouds are delivered, three major types of clouds include public, private and hybrid clouds; 2) at a business level, cloud providers lease these resources immediately and temporarily to cloud consumers when required. Often leases are paid by a regular credit card transaction on a consumption basis.

In this paper, we explore and evaluate the parallel processing performance of a task from the Life Science based on IaaS. Namely, we rent computing infrastructures from a cloud provider (i.e., Amazon Elastic Compute Cloud (Amazon EC2 [3]) with full control of those computing resources. We have developed the task in both sequential and parallel ways and deployed them onto the rented infrastructure. The performance has been evaluated and compared using speedup as a metric.

The rest of paper is structured as follows: Section 2 presents an image pattern recognition use case in the Life Sciences, to which we have applied parallelisation. Section 3 describes the experiments we have conducted. Section 4 concludes the work.

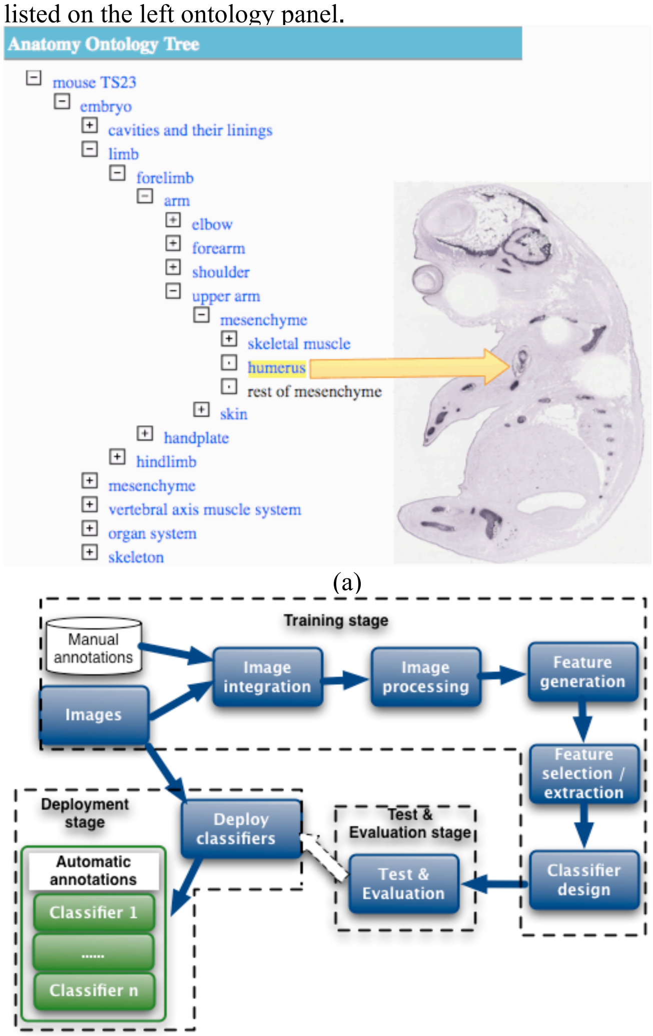
II. PARALLEL DATA INTENSIVE APPLICATIONS: AN IMAGE PATTEN RECOGNITION CASE STUDY

A. *Backgroung of the use case*

The use case is from EUREXpress [4][5], which aims to build a transcriptome-wide atlas for developing mouse embryo established by RNA in situ hybridisation. The project uses automated processes for in situ hybridisation experiments on all genes of whole-mount wild-type mouse embryos at the Stage 23. The result is many images of embryo sections that are stained to reveal where RNA is

ISBN: 978-1-61208-101-4

present, namely, where gene patterns are expressed in embryos. These images were then annotated by human curators. The annotation consists of tagging images with anatomical terms from the ontology for mouse anatomy development. If an image is tagged with an anatomical term, it means that anatomical component is present in the image and it is exhibiting gene expression in some part of the component. So far, 80% of images (4 Terabytes in total) have been manually annotated by human curators. The goal is to automatically perform annotation by tagging the remaining 20% with the correct terms of anatomical components (there are still 85,824 images to be annotated with a vocabulary of 1,500 anatomical terms) and to provide a means of tagging future data automatically. The input is a set of image files and corresponding metadata. The output will be an identification of the anatomical components that exhibit gene expression patterns in each image. This is a typical pattern recognition task. As shown in Figure1 (a), we first need to identify the features of `humerus' in the embryo image and then annotate the image using ontology terms listed on the left ontology panel.



(a)

(b)

Figure 1. An image pattern recognition task

To automatically annotate images, three stages are required: at the training stage, the classification model has to be built, based on training image datasets with annotations; at the testing stage, the performance of the classification model has to be tested and evaluated; then at the deployment stage, the model has to be deployed to perform the classification of all non-annotated images. We mainly focus on the training stage in this case. The processes in the training stage include integration of images and annotations, image processing, feature generation, feature selection and extraction, and classifier design, as shown in Figure1 (b).

The specific processes are described as follows:

- Image integration: before starting the data mining, we need to integrate data from different sources: the manual annotations have been stored in the database and the images are located in the file system. The output of this process is images with annotations.
- The size of the images is variable and there is noise in the images. We use image scaling and image filtering methods to rescale and denoise the images. The output of this process is standardised and denoised images, which can be represented as 2-dimensional arrays.
- After image pre-processing, we generate those features that represent different gene expression patterns in images. The resulting features of wavelet transforms are 2-dimensional arrays.
- Due to the large number of features, the features need to be reduced and selected for building a classifier. Either feature selection or feature extraction or both can do this. Feature selection selects a subset of the most significant features for constructing classifiers. Feature extraction performs the transformation on the original features for the dimensionality reduction to obtain a representative feature vectors for building up classifiers.
- The main task in this case is to classify images into the right gene terminologies. The classifier needs to take an image's features as an input, and outputs a 'yes' or 'no' for each of anatomical features.

B. *Parallel the image patten recognition task*

1) *Overview of parallel approach*

It is well known that the speedup of an application to solve large computational problems is mainly gained by the parallelisation at either hardware or software levels or both (e.g., signal, circuit, component and system levels) [6]. Hardware parallelism focuses on signal and circuit levels and normally is constrained by manufacturers. Software parallelism at component and system levels can be classified into two types: automatic parallelisation of applications without modifying existing sequential applications and construction of parallel programming models using various software technologies to describe parallel algorithms and then match applications with the underlying hardware platforms. Since the nature of auto-parallelisation is to

ISBN: 978-1-61208-101-4

recompile a sequential program without the need for modification, it has a limited capability of parallelisation on the sequential algorithm itself. Mostly, it is hard to directly transform a sequential algorithm into parallel ones. While parallel programming models try to address how to develop parallel applications and therefore can maximally utilise the parallelisation to obtain high performance, it does need more development effort on parallelisation of specific applications. In general, three considerations when parallelising an application include:

- How to distribute workloads or decompose an algorithm into parts as tasks?
- How to map the tasks onto various computing nodes and execute the subtasks in parallel?
- How to coordinate and communicate subtasks on those computing nodes.

There are mainly two common methods for dealing with the first two questions: data parallelism and task parallelism. *Data parallelism* represents workloads are distributed into different computing nodes and the same task can be executed on different subsets of the data simultaneously. *Task parallelism* means the tasks are independent and can be executed purely in parallel. There is another special kind of the task parallelism is called ‘pipelining’. A task is processed at different stages of a pipeline, which is especially suitable for the case when the same task is used repeatedly. The extent of parallelisation is determined by dependencies of each individual part of the algorithms and tasks.

As for the coordination and communication among tasks or processes on various nodes, it depends on different memory architectures (shared memory or distributed memory). A number of communication models have been developed [7][8]. Among them, the MPI (Message Passing Interface) has been developed for HPC parallel applications with distributed memory architectures and has become the de-facto standard. There is a set of implementations of MPI, for example, OpenMPI [9], MPICH [10], GridMPI [11] and LAM/MPI [12].

2) Parallel approach in this use case

Based on the flow chart of the use case in Figure 1b, we model this image pattern recognition task as a direct acyclic graph. Each node of the graph is a functional module, a process or a subtask and the edge connected between nodes is data flow, which shows true data dependency between them. A direct acyclic graph for the training stage is shown in Figure 2. The left-hand side shows the atomic processes of the training stage. The right-hand side shows a higher-level abstraction of the task. For instance, feature selection and extract is composed of ‘featureMean’, ‘featureVar’ and ‘featureExtract’ atomic processes.

In terms of the nature of the algorithm used in this case, parallel approaches used are mainly data parallelism and a typical task parallelism (pipelining).

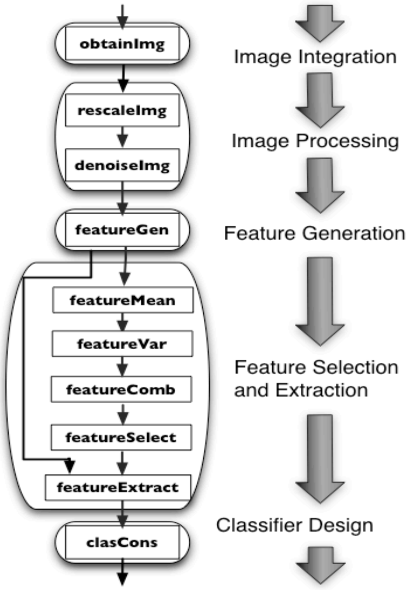


Figure 2. The processes of the task

In terms of the task graph shown in Figure 2 and the dependencies between the processes, data parallelism can be applied here. Processes such as ‘rescaleImg’ and ‘denoiseImg’ to ‘featureGen’ can have multiple instances invoked without any internal status to be maintained between these instances. After we get image samples from the process ‘obtainImg’, the samples can be partitioned into subsets and distributed to different nodes that run multiple instances of these processes. Therefore, the data can be parallelised.

Furthermore, parallelisation should consider how to decompose a process itself into parts and executed these parts in parallel. In this case, the decomposition of the algorithms mainly focuses on feature selection and extraction (i.e., Fisher's Ration algorithm [13]) and Classifier design (i.e., K-Nearest Neighbour-KNN [14]). We have developed parallel forms of these two processes, as shown in Figure 3 and Figure 4.

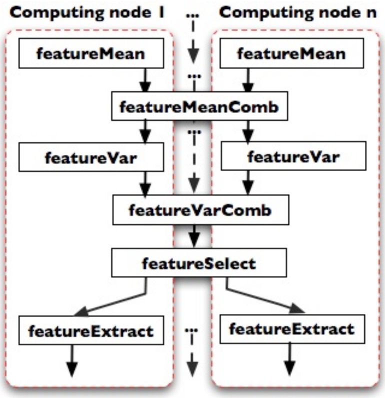


Figure 3. A parallel form of Fisher’s Ratio

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-101-4

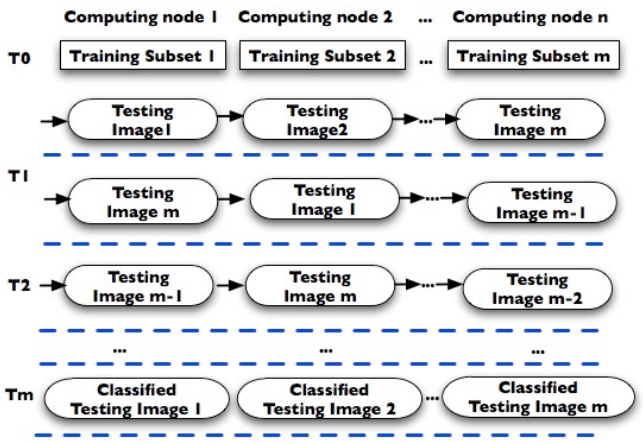


Figure 4. A parallel form of KNN

a) The rationale behind of parallel Fisher Ratio

The nature of the Fisher's Ratio algorithm for feature selection is to calculate Mean and Variance of image samples for two classes (C_1 and C_2), as shown in the following Equation 1 [13].

$$FisherRatio = \frac{(m_{1,i} - m_{2,i})^2}{(v_{1,i}^2 + v_{2,i}^2)} \quad (1)$$

where $m_{1,i}$ represents the mean of samples at the ith feature in Class C_1 and $m_{2,i}$ represents the mean of samples at the ith feature in Class C_2. $v_{1,i}$ represents the variance of samples at the ith in Class C_1. $v_{2,i}$ represents the variance of samples at the ith feature in C_2.

Therefore, the feature selection can be decomposed into smaller subtasks 'featureMean' and 'featureStd' and executed with subsets of image samples on nodes. The parallel form of the algorithm can be presented in Figure 4.

b) The Rationale behind of Parallel KNN

KNN is a classification algorithm to identify unknown samples into a class based on the nearest distance with the training samples. The common distance function is Euclidean distance. In this case, the samples are represented with features as vectors. The Euclidean distances therefore are calculated between each training sample and a testing sample, and then the nearest ones can be chosen. For all of training samples, the calculation is a typical iteration task. To exploit parallelisation in this algorithm, we use a special task parallelism called 'pipelining'. We divide an iteration of the KNN task into several pipeline stages. The sample images can be partitioned to subsets. The task of the distance calculation between subsets of training samples and the unclassified testing sample are executed at different stages respectively. Figure 4 shows the parallel form of the KNN algorithm using the concept of 'pipelining'.

III. EXPAERIMENTATION AND EVALUATION IN CLOUD COMPUTING

To evaluate the performance of the parallel image pattern recognition task described above we must conduct experiments on many physical computer nodes. However, to buy and maintain physical resources is costly and time consuming. We therefore make use of IaaS in the form of cloud computing to perform our evaluation.

A. Overview of Cloud Computing

Cloud computing is an evolution of various forms of distributed computing systems: from original distributed computing, parallel computing (cluster, to service-oriented computing (e.g., grid). Cloud computing extends these various distributed computing forms by introducing a business model, in which the nature of on-demand, self-service and pay-by-use of resources is used. Cloud computing sells itself as a cost-effective solution and reliable platform. It focuses on delivery of services guaranteed through Service Level Agreements. The services can be application software—SaaS, development environments for developing applications—PaaS, and raw infrastructures and associated middleware —IaaS.

In this study, without buying expensive clusters or supercomputers, we adopt the IaaS model that enables us to rent compute infrastructures from cloud providers. We have developed and then deployed our application onto rented compute infrastructure.

Among cloud providers (Amazon, Google, Saleforce, IBM, Microsoft, etc), we have chosen Amazon EC2, one of the most popular cloud providers, who provides Infrastructure-as-a-Service to users, with a capability to allow users to elastically expand or shrink the amount of resources used. Unlike traditional physical resource leasing, Amazon EC2 uses virtualisation techniques (e.g., Xen [15]) and releases virtual machines (instances) to users. Table 1 lists standard virtual instances from Amazon as an example (please refer to other types of virtual instances in [3]). Users can choose any instance and operating systems and pay for the time an instance is 'switched on' (Table 1 also shows the pricing for using Linux/Unix operating system on 1 June 2010).

TABLE I. AN EXAMPLE OF INSTANCE TYPES OF AMAZON EC2

Stand. Instance	**Cores**	**RAM**	**Bit**	**I/O**	**Disk**	**Cost Linux/Unix**
Small (m1.small)	1	1.7GB	32	Med.	160GB	$0.085/h
large (m1.large)	4	7.5GB	64	High	850GB	$0.34/h
Extra large (m1.xlarge)	8	15GB	64	High	1690gb	$0.68/h

B. Experimentatation and Evaluation in the Cloud

1) Performance metrics

We have used speedup as a performance indicator. Speedup is considered as a ratio between the execution time of the image pattern recognition task (Ts) on one single computing node and the execution time of the task on multiple computing nodes (Tm), represented as follows:

$$S = \frac{T_s}{T_m}$$

The execution on one single node means all of processes, data, and storage on one computing node. The execution on multiple computing nodes includes any of these situations: distributed data, distributed processes and distributed storage.

2) Experiment configuration

We have developed the image pattern recognition task in both sequential and parallel modes. We deployed them and conducted our experiments on small instances (virtual computers) of Amazon's EC2, with speedup as a performance indicator. To create an infrastructure on EC2, we have registered a user account with Amazon EC2. We launch instances by specifying the instance types and virtual images that we have created for the image pattern recognition task. In this study, we have used 12 small instances for experimentation (at this moment Amazon EC2 limits users to a maximum of 20 instances running concurrently). The specific configuration is listed in the second row in Table 1 (m1.small).

3) Evaluation result

We have measured the performance under two factors: 1) changing the size of input (i.e., number of image samples) and 2) varying number of nodes (i.e., virtual computers). We report averages over 30 independent runs for each scenario (the choice of 30 runs is based on statistics). The evaluation result is shown in Figure 5, Figure 6 and Figure 7. To validate the cost-effectiveness of Infrastructure-as-a-Service via cloud provision, we calculated the cost shown in Figure 8. Figure 5 shows the speedup. The X-axis represents input size and the Y-axis represents the speedup. The result demonstrates that the speedup increases with the increase of the number of computing nodes. The speedup increases with the increase of input size when the number of machines is 12. It fully embodies the advantage of parallel computing when processing heavy loads (i.e., with 4000 images).

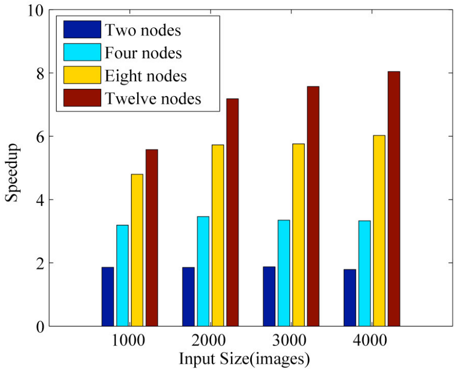


Figure 5 Average speedup for experiments with 1000, 2000, 3000 and 4000 images when using 2, 4, 8 and 12 nodes running concurrently

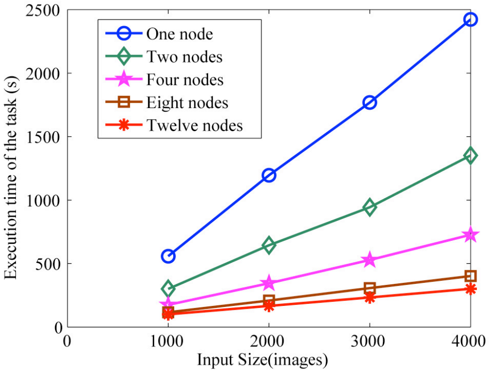


Figure 6 Average execution time of the task with increasing number of images for different number of virtual nodes running concurrently

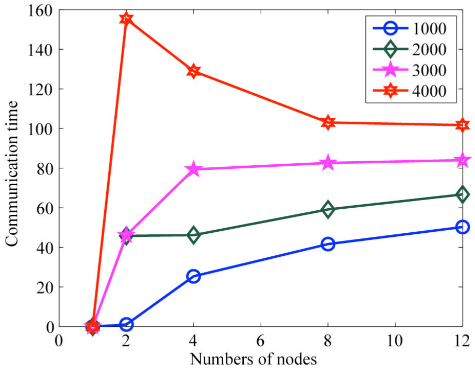


Figure 7 Average communication time vs. the numbers of nodes for increasing number of nodes with different number of images

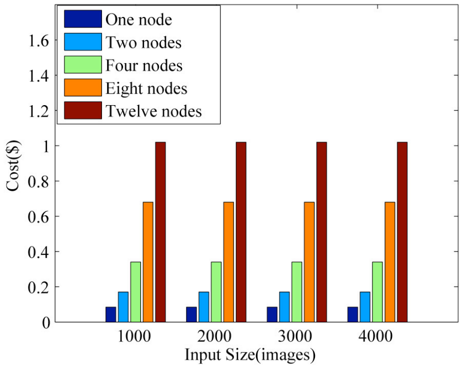


Figure 8 Average cost for performing the task using Amazon EC2 in US Dollars with 1000, 2000, 3000 and 4000 images when using 1, 2, 4, 8 and 12 virtual nodes running concurrently

Figure 6 shows the execution time under different input sizes and different numbers of computing nodes. With the increase of input size, the execution time increases; with the increase of numbers of nodes, the execution time decreases.

Figure 7 shows the relationship between communication time and numbers of nodes. With the increase of input size,

ISBN: 978-1-61208-101-4

the communication time increases; with the increase of the number of nodes, the communication time increases in the first instance and then the increase rate slows down. In Figure 7, the communication time with input size 4000 at two nodes has a spike. This is mainly caused by the latency of the Cloud during the execution after we have checked various similar experiments.

Figure 8 shows average costs for running a full task in US Dollars using Amazon's EC2. In this case, since the maximum execution time of the whole task for various experiments running on one node is taken less than one hour, the cost increases with the number of the virtual computing nodes, for example, for one node, the cost is $0.085; for 12 nodes, the cost is $1.02. From cost-effectiveness point of view, the users may consider running fewer virtual nodes (despite that the execution time of the task running on 12 nodes is less than 5 minutes, comparing with 41-minute task execution on one node).

IV. Concluding Remarks

In this paper, by a way of a case study, we explore parallel approaches for an image pattern recognition task in the Life Sciences and have developed both sequential and parallel versions for this task. We have conducted a comparison of different parallel setups that ran in a cloud infrastructure provided by Amazon's EC2. The performance of parallel processing of the task has been evaluated where the speedup increases with the increase of numbers of virtual computer nodes and we achieve a linear scale up when using maximum input size of 4000 images. The communication time increases with the increase of input size. With the increase of number of virtual computer nodes, the communication time increases at the first beginning and then the increase rate slows down.

We have calculated the average cost for running the whole task. The maximum cost is $1.02 when lunching 12 virtual nodes, which shows that the use of Cloud computing is still a cheaper solution comparing with that of buying supercomputers or clusters. Nevertheless, it is also found the cost increases with the number of computing nodes as long as the maximum execution time of the task running one node is less than one hour. In this case, there is still a possibility that the users may choose to lunch fewer virtual nodes for cost saving.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud Computing and Emerging IT Platforms: Vision, Hype and Reality for Deliverung Domputing as the 5th Utility," Future Generation Computer Systems, Vol.25, No. 6., pp.599-616, June 2009.

[2] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Grid Computing Environments Workshop, 2008.

[3] Amazon Elastic Compute Cloud, http://aws.amazon.com/ec2/, Retrieved 15, May, 2010.

[4] L. Han, J. van Hemert, R. Baldock, and M. Atkinson, "Automating Gene Expression Annotation for Mouse Embryo," In Lecture Notes in Computer Science (Advanced Data Mining and Applications, ADMA 2009), vol. LANI 5678, pp.469-478, 2009.

[5] EURExpress-II project, http://www.eurexpress.org/ee/, Retrieved 10, May, 2010.

[6] L. Silva, and R. Buyya, "High Performance Cluster Computing: Programming and Applications," ch. Parallel Programming Models and Paradigms, pp. 4–27. No. ISBN 0-13-013785-5. Prentice Hall PTR, NJ, USA, 1999

[7] P. S. Pacheco Parallel Programming with MPI. Morgan Kaufmann Publishers, Inc., 1997.

[8] PVM, 2009, http://www.csm.ornl.gov/pvm/, Retrieved 5 May, 2010.

[9] OpenMPI, 2009, http://www.open-mpi.org/, Retrieved 5 May, 2010.

[10] MPICH, http://www.mcs.anl.gov/research/projects/mpi/mpich1/, Retrieved 5 May, 2010.

[11] GridMPI, http://www.gridmpi.org/index.jsp, Retrieved 5 May, 2010.

[12] LAMMPI, http://www.lam-mpi.org/, Retrieved 5 May, 2010.

[13] R. O. Duda and P. E., Hart, "Pattern Classification and Scene Analysis," John Wiley & Sons, 1973.

[14] T. M. Cover and P. E. Hart, "Nearest Neighbor Pattern Classification," Information Theory, IEEE Transactions on, Vol. 13, No. 1. pp. 21-27,1967.

[15] P. Barham, B. Dragovic, K. Fraser, S. Hand, T.L. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield , "Xen and the art of virtualisation," In SOSP, pp. 164-177. ACM, 2003.

Using Autarky to Evaluate Quantified Boolean Formulae

Jens Rühmkorf
Simulation and Software Technology
German Aerospace Center (DLR)
Linder Höhe, D-51147 Köln, Germany
E-mail: Jens.Ruehmkorf@dlr.de

***Abstract*— In this paper, we discuss algorithmical implications for the extension of autarky from propositional logic to evaluate quantified boolean formulae (QBF). First, the Davis-Putnam procedure for the satisfiability problem (SAT) is described. Then we explain efficient known data structures for SAT and extensions to QBF which we used in our solver. Finally, we introduce the concept of autarky and describe how detecting 2-autarky structures in a given QBF formula helps pruning the search tree. To the best of our knowledge we are the first to describe such techniques for QBF.**

***Keywords*—Autarky; Davis-Putnam; SAT; QBF**

I. Introduction

In recent years, the language of quantified boolean formulae (QBF) has gained importance for practical applications. QBF allows for a concise representation of many classes of problems [1], [2]: Gopalakrishnan et al. study the problem of formally verifying shared memory multiprocessor executions against memory consistency models for the Intel Itanium by translating occurring problems to the satisfiability problem (SAT) and QBF [1]. Mneimneh et al. also consider the application area of formal hardware verification and transform the diameter problem — determining the length of a longest of all shortest paths — for a class of large digraphs to QBF, but end up converting their problem to SAT, because no existing QBF solver is able to solve their problems [2].

The paper is organized as follows: Section II introduces preliminary definitions and concepts. Section III describes the Davis-Putnam procedure for SAT along with efficient data structures. Section IV discusses efficient data structures used for extending Davis-Putnam to QBF. Based on our experimental results (due to space limitations not discussed in this paper). Sections V and VI derive enhancements to the current algorithm by extending the concept of autarky to QBF and discuss possible implementation approaches. Section VII concludes the paper by briefly evaluating the current status and provides information on future development.

II. Preliminaries

A quantified boolean formula $\Phi = q_1 z_1 \cdots q_n z_n \ \varphi$ consists of a sequence of quantified variables, the so-called *prefix*, followed by a quantifier free SAT formula φ, the *matrix* of the formula. The prefix $q_1 z_1 \cdots q_n z_n$ contains universal $\forall$ and existential $\exists$ quantifiers for propositional variables z_i occurring in φ. The *evaluation problem* for a given QBF Φ is to decide whether Φ is true or not. Example QBF formulas are $\forall y_1 \exists x_2 \ (y_1 \vee \neg x_2) \wedge (\neg y_1 \vee x_2)$ which is true and $\forall y_1 \forall y_2 \ (y_1 \vee y_2)$ which is false.

A *literal* L is a variable z or $\neg z$. For two different literals L_i, L_j with corresponding variables z_i, z_j of a QBF formula Φ we may write $L_i < L_j$ if z_i occurs to the left of z_j in the prefix of Φ. We use $\text{Lit}(Z)$ as shorthand notation for the set of literals for a given set of variables Z. Similarly, $\text{Var}(\Phi)$ and $\text{Var}(\varphi)$ are used for the variable sets occurring in a QBF formula Φ and a SAT formula φ, respectively. A *clause* is a formula $\kappa = (L_1 \vee \cdots \vee L_k)$ with literals L_i; the second clause of the first example is $(\neg y_1 \vee x_2)$ with literals $\neg y_1$ and x_2. We say a SAT formula φ is in *conjunctive normal form* (CNF), if it is expressed by a conjunction of clauses $\varphi = \kappa_1 \wedge \cdots \wedge \kappa_\ell$.

III. Davis-Putnam for SAT

Both SAT and QBF describe prototypical complete problems for the important complexity classes $\mathcal{NP}$ and $\mathcal{PSPACE}$, respectively. Syntactically restricted forms of QBF describe complete problems for $\Sigma_k^{\mathcal{P}}$ and $\Pi_k^{\mathcal{P}}$ within the polynomial time hierarchy $\mathcal{PH}$. For the remainder of this paper, we consider only quantified boolean formulae Φ whose matrix φ is in CNF. Indeed, for a given QBF formula Φ we may generate in linear time an equivalent quantified boolean formula whose matrix is in CNF [4, ch. 7].

The evaluation algorithm used within this paper is a generalization of the Davis-Putnam procedure for SAT [5] to QBF [6]. Figure 1 on the following page describes the Davis-Putnam algorithm as recursive function. The algorithm utilizes the following two fundamental observations:

Lemma 1 (monotone literal [5]) *If the literal L is* mono*tone, i.e. by definition, L occurrs only positive ($L = x$) or negative ($L = \neg x$) within the* CNF *formula α, then α is equivalent to $\alpha[L/1]$ or $\alpha[L/0]$, respectively.* □

Lemma 2 (unit clause [5]) *Let L be the literal of a unit clause of the* CNF *formula α. Then α is satisfiable if and only if $\alpha[L/1]$ is satisfiable.* □

Here $\alpha[L/\epsilon]$ denotes the formula obtained from α by replacing each occurrence of L with $\epsilon \in \{0, 1\}$. Furthermore, a clause κ is called k-clause, if κ contains only k literals $(L_1 \vee \cdots \vee L_k)$, a 1-clause (L) is called *unit clause*.

```
Function boolean Davis-Putnam(CNF formula* $\alpha$)
  Input: Pointer to CNF formula $\alpha$.
  Output: true if $\alpha$ is satisfiable and false otherwise.
  begin
      if $\alpha = 1$ then return true;
      if $\alpha = 0$ then return false;
      $L \leftarrow$ Pure-Literal($\alpha$);
      if $L \neq$ NULL then
          return Davis-Putnam($\alpha[L/1]$);
      $L \leftarrow$ Unit-Literal($\alpha$);
      if $L \neq$ NULL then
          return Davis-Putnam($\alpha[L/1]$);
      $L \leftarrow$ Choose-Literal($\alpha$);
      if Davis-Putnam($\alpha[L/1]$) then
          return true;
      return Davis-Putnam($\alpha[L/0]$);
  end
```

Figure 1: The Davis-Putnam algorithm for SAT.

Function `Pure-Literal` corresponds to lemma 1: it returns for a formula α a pointer to a monotone literal if it exists and `NULL` otherwise. Function `Unit-Literal` utilizes lemma 2 and returns a unit clause if it exists and otherwise `NULL`. The function `Choose-Literal` defines the heuristic which literal to use next for branching. Good experimental results are obtained by using the lexicographical heuristic [7]: Let $h_i(L)$ be the number of clauses of length i in which a given literal L occurs. Then calculate:

$$H_i(A) = \max\big(h_i(x), h_i(\neg x)\big) + 2\min\big(h_i(x), h_i(\neg x)\big) \quad (1)$$

Then, the variable with maximal vector $(H_1(x), \cdots, H_n(x))$ according to the lexicographical ordering is chosen.

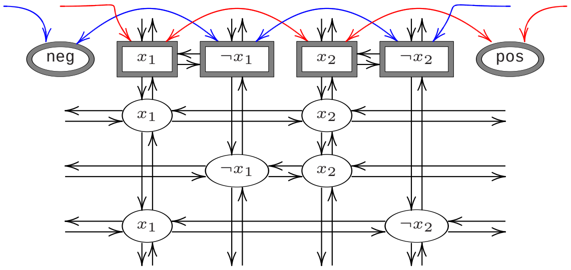


Figure 2: Data structure for the 2-CNF formula $\alpha_1 = (x_1 \vee x_2) \wedge (\neg x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)$.

As can be seen in figure 1, Davis-Putnam uses a depth first strategy where backtracking occurs when a leaf labelled with 0 is reached in its execution tree. The algorithm executes a method call `Assign(`α, L`)` or `Assign(`α, $\neg L$`)` for every occurrence of $\alpha[L/1]$ or $\alpha[L/0]$, respectively. Upon leaving the recursion on level $\alpha[L/1]$ the implicitly called method `Unassign(`α, L`)` modifies the current formula to α by making use of a recursion stack.

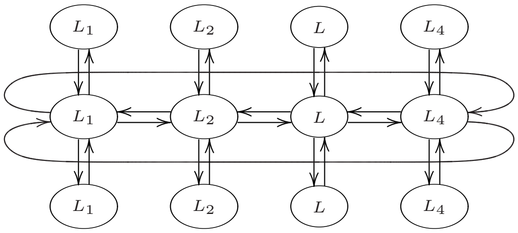


Figure 3: Data structure of clause $\kappa_L = (L_1 \vee L_2 \vee L \vee L_4)$.

The formulation of Davis-Putnam is surprisingly simple. Of key importance is the realization of data structures that efficiently support necessary operations. With the sparse data structure by Böhm and Speckenmeyer [7] used for our solver the operations `Assign()` and `Unassign()` need time $\mathcal{O}\big(|\alpha| - |\alpha[L/1]|\big)$, the test for unit clauses needs time $\mathcal{O}(1)$.

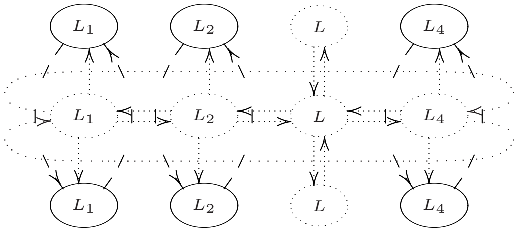


Figure 4: Remove clause κ_L in time $\mathcal{O}(|\kappa_L|)$.

Figure 2 shows the used data structure for a 2-CNF formula. There, literals and clauses are connected in the following way: Every occurrence of a literal in a clause corresponds to a literal object within the data structure, in figures 2 to 5 depicted by $\neg x$ or L.

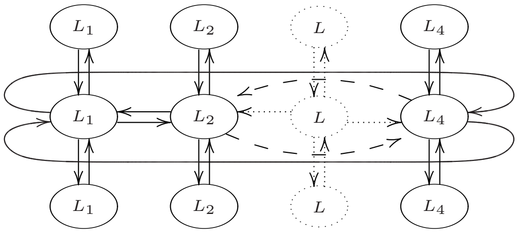


Figure 5: Shorten clause κ_L in time $\mathcal{O}(1)$.

All literals of a clause are connected through a doubly-linked circular clause list as shown in figure 3. All literals of the same type are connected through doubly-linked circular lists, so called *literal occurrence lists* depicted by the columns in figure 2. The head of such a literal occurrence list is displayed by $\neg x$. These list heads are themselves divided in two doubly-linked, circular lists. The list pos connects the list heads of all positive literals, whereas the list neg connects the list heads of all negative literals. These two lists represent the yet unassigned literals in the given formula.

When applying changes within the data structure, bookmarking links are set to be able to easily revert changes made when traversing the execution tree. Figure 4 on the preceding page shows the changes to the datastructure when removing a clause (e.g. when performing $\alpha[L/1]$) and similarly figure 5 shows the changes performed when shortening a clause (e.g. when performing $\alpha[L/0]$).

Operation	Runtime
`Unassign(`α,L`)`, `Assign(`α, L`)`	$\mathcal{O}(\lvert\alpha\rvert - \lvert\alpha[L/1]\rvert)$
`Unit-Literal(`α`)`	$\mathcal{O}(1)$
Remove clause κ from α	$\mathcal{O}(\lvert\kappa\rvert)$
Remove L from clause κ	$\mathcal{O}(1)$
Find clause κ with $L \in \kappa$	$\mathcal{O}(1)$

Figure 6: Runtime of operations for CNF data structure.

Figure 6 lists important operations of the data structure with their corresponding runtime behaviour.

IV. Evaluating QBF

This section describes the changes necessary to extend Davis-Putnam to QBF. The following lemma proves to be an easy but fundamental tool for this:

Proposition 3 (substitution lemma [4]) *Let Π be a prefix and let Φ_1 as well as Φ_2 be two quantified boolean formulae. Then follows from the equivalence of Φ_1 and Φ_2, written $\Phi_1 \approx \Phi_2$, that the quantified formula $\Pi\ \Phi_1$ is equivalent to $\Pi\ \Phi_2$, in other words: $\Pi\ \Phi_1 \approx \Pi\ \Phi_2$.* □

According to proposition 3, we may transform the matrix of a QBF formula, just like we would for a CNF formula. The proof is an easy induction on the length of the prefix Π. Furthermore, we consider the following two lemmas which may be easily derived from their CNF counterparts.

Lemma 4 (monotone quantified literal [6]) *Let Φ be a quantified boolean formula and let L be a monotone literal of Φ, i.e. a literal whose complement $\neg L$ does not occur in the matrix of Φ. Then the following holds:*

1) *In case L is $\exists$-quantified, then Φ is true if and only if $\Phi[L/1]$ is true.*
2) *In case L is $\forall$-quantified, then Φ is true if and only if $\Phi[L/0]$ is true.* □

Lemma 4 means that in case of an $\exists$-quantified monotone literal L we may remove all clauses containing L, whereas in case of an $\forall$-quantified such literal we may shorten all clauses that contain L by removing L.

We call a clause of a quantified boolean formula *unit existential clause* if it contains exactly one $\exists$-quantified literal for a variable y and all $\forall$-quantified literals for variables x occur to the right of y within the prefix of the formula. The QBF datastructure saves all literals of a clause in the order their corresponding variables occur within the prefix.

Lemma 5 (existential unit clause) *Let L be the $\exists$-quantified literal of an unit existential clause κ of a quantified boolean formula Φ, i.e., $L < L_i$ for all other ($\forall$-quantified) literals L_i of the clause κ. Then Φ is true if and only if $\Phi[L/1]$ is true.* □

```
Function boolean DP-QBF(QBF formula* $\Phi$)
  Input: Pointer to QBF formula $\Phi$.
  Output: true if $\Phi$ evaluates to true and false otherwise.
  begin
      if $\Phi$ = true or $\Phi$ = false then return $\Phi$;
      $L \leftarrow$ Pure-Literal($\Phi$);
      if $L \neq$ NULL then
          switch Quantifier($L$) do
              case $\exists$ : return DP-QBF($\Phi[L/1]$);
              case $\forall$ : return DP-QBF($\Phi[L/0]$);
      $L \leftarrow$ Unit-Literal($\Phi$);
      if $L \neq$ NULL then
          return DP-QBF($\Phi[L/1]$);
      $L \leftarrow$ Choose-Literal($\Phi$);
      switch Quantifier($L$) do
          case $\exists$ :
              if DP-QBF($\Phi[L/1]$) or DP-QBF($\Phi[L/0]$)
              then
                  return true;
              else
                  return false;
          case $\forall$ :
              if DP-QBF($\Phi[L/1]$) and DP-QBF($\Phi[L/0]$)
              then
                  return true;
              else
                  return false;
  end
```

Figure 7: Skeleton of Davis-Putnam algorithm for QBF.

The Davis-Putnam extension to QBF may be formulated as described in figure 7. There the functions `Pure-Literal()` and `Unit-Literal()` correspond to lemmata 4 and 5, respectively. We examine the heuristic which delivers the literal to set next. Different to SAT the choice for QBF is restricted to the leftmost group of variables within the prefix that have the same quantifier. That means for a quantified boolean formula $\Phi = \forall Y_1 \exists X_2 \forall Y_3 \cdots \exists X_k\, \varphi$ with $\forall$-quantified variable sets Y_i and $\exists$-quantified sets X_j first all literals belonging to variables from Y_1 are considered, then all literals belonging to variables from X_2, and so forth.

The choice of a literal out of $\text{Lit}(Y_i)$ respectively $\text{Lit}(X_j)$ is then determined by the function `Choose-Literal()`. For the lexicographic heuristic for SAT described by equation (1), the literal is chosen which occurs most often in the shortest clauses of a given formula. Translated to QBF, a literal with such properties out of the leftmost prefix group is chosen. For QBF, the length of a clause is measured by counting the number of $\exists$-quantified literals whithin a clause, irrespective of its corresponding position within the clause (see [8] for a similar approach). For example: a clause $(x_1 \vee y_2 \vee y_3)$ is treated by this modified heuristic just like the clause $(y_4 \vee x_5 \vee y_6 \vee y_7)$, while the unmodified heuristic from equation (1) would rank the literals of the first clause better than literals from the second clause.

V. Utilizing Autarky for QBF

A function $\mathfrak{I} : \{x_0, x_1, x_2, \ldots\} \to \{0, 1\}$ for variables x_i is called a *truth assignment*. If $\mathfrak{I}$ is a *partial assignment* that operates on a subset of the variables $\text{Var}(\varphi)$ of a SAT formula φ, then $\mathfrak{I}(\varphi)$ denotes the formula obtained by assigning truth values to this subset's variables accordingly.

Definition 6 (autark assignment [9]) *A truth assignment $\mathfrak{I}$ of some variables $\{x_{i_1}, \ldots, x_{i_k}\}$ of a* SAT *formula φ is called* autark*, if the following holds: every clause of φ that contains a variable x_{i_j} is already satisfied by $\mathfrak{I}$.*

If such an $\mathfrak{I}$ "touches" a clause of φ, this clause is already satisfied by $\mathfrak{I}$: every clause of $\mathfrak{I}(\varphi)$ occurs in φ. Autarky has the nice property that we may remove all clauses with variables x_{i_j} from φ without changing the satisfiability of φ. The following easy remark gives further insight:

Remark 7 (Satisfiability of autark assignments) If $\mathfrak{I}$ is an autark assignment of variables $V_{\text{aut}} = \{x_{i_1}, \ldots, x_{i_k}\}$ for a SAT formula φ, then φ is satisfiable if and only if an assignment $\mathfrak{I}'$ exists that satisfies φ and the restriction of $\mathfrak{I}'$ to V_{aut} is identical to $\mathfrak{I}$, i.e., $\mathfrak{I}'|_{V_{\text{aut}}} = \mathfrak{I}$.

Proof. Let $\mathfrak{I}$ be an autark assignemnt for φ, with variable set V_{aut} and let $\mathfrak{H}$ be a thruth assignment that satisfies φ. We may alter $\mathfrak{I}$ in accordance to $\mathfrak{H}$ by defining $\mathfrak{I}'(x) = \mathfrak{I}(x)$ if x belongs to V_{aut} and $\mathfrak{I}'(x) = \mathfrak{H}(x)$ otherwise. Then $\mathfrak{I}'$ satisfies φ. □

For the easy case of 1-autarky with $|V_{\text{aut}}| = 1$ remark 7 corresponds to lemma 1 on page 1, the rule monotone literal. Our experiments showed that this rule lead to good results for quantified formulas, i.e. to considerably less branching nodes within our execution tree.

Therefore we examine how to extend the concept of autarky to quantified boolean formulae. We consider the case of 2-autarky for a QBF formula Φ. Without loss of generality Φ may contain no monotone literals. That means we examine all variable subsets from $\text{Var}(\Phi)$ with size 2, respecting their order in the prefix. The idea is to compute these subsets in advance and later utilize them for search tree pruning. For this, the following cases need to be considered:

Case 1: 2-$\exists\exists$-autarky $\{x_{i_1}, x_{i_2}\}$. If an autark truth assignment exists for two $\exists$-quantified variables $x_{i_1} < x_{i_2}$, we have an already known case: all clauses, that contain either x_{i_1} oder x_{i_2} may be removed. This is also true if x_{i_1} and x_{i_2} do not belong to the leftmost prefix group.

Case 2: 2-$\forall\forall$-autarky $\{y_{i_1}, y_{i_2}\}$. For this case we closer examine the structure of all clauses that contain y_{i_1} or y_{i_2}. There are eight possibilites for membership of y_{i_1} or y_{i_2} within a clause. Figure 8(a) shows the four possible ways that a clause contains either y_{i_1} or y_{i_2}, positive or negative. Figure 8(b) shows the four possibilities that y_{i_1} as well as y_{i_2} are contained in a clause. There $\bullet$ describes a positive literal, $\bar{\bullet}$ describes a negative literal, and $\circ$ shows that the variable in question is not contained in the clause.

$$\begin{array}{cc}
\bullet & \circ \\
\bar{\bullet} & \circ \\
\circ & \bullet \\
\circ & \bar{\bullet}
\end{array}
\qquad\qquad
\begin{array}{cc}
\bullet & \bullet \\
\bullet & \bar{\bullet} \\
\bar{\bullet} & \bullet \\
\bar{\bullet} & \bar{\bullet}
\end{array}$$

(a) 1-structure. (b) 2-structure

Figure 8: Clause structure for 2-autarky.

We have $2^8 = 256$ possible structural occurrences of two given distinct variables in the clauses of a CNF formula. Of these occurrences only those need to be considered where both variables occur at least once in a clause; so seven cases may be rejected. By a combinatorical argument with some case distinctions we may identify 90 cases of 2-autarkies and therefore 159 cases of not-2-autarkies — this includes symmetries and renamings of the kind $z \leftarrow \neg z$.

$$\begin{array}{cc|cc|cc|cc|cc}
 & & & & \Downarrow & & \Downarrow & & \Downarrow & \\
\bullet & \bullet & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\
\bullet & \bar{\bullet} & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
\bar{\bullet} & \bullet & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}$$

(a) Three necessary branchings.

$$\begin{array}{cc|cc|cc|cc|cc}
 & & & & \Downarrow & & \Downarrow & & & \\
\bullet & \circ & 1 & & 1 & & 0 & & 0 & \\
\bullet & \bar{\bullet} & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
\bar{\bullet} & \bullet & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}$$

(b) Two necessary branchings.

Figure 9: Branchings for 2-$\forall\forall$-autarky.

We discuss essential ideas with the help of some examples. Figure 9(a) shows a 2-autarky with three possible types of clause-structures $(\ldots y_{i_1} \vee y_{i_2} \ldots), (\ldots y_{i_1} \vee \neg y_{i_2} \ldots)$ as well as $(\ldots \neg y_{i_1} \vee y_{i_2} \ldots)$. First we consider the case that both variables are part of the leftmost prefix group. Then branching according to the first column (i.e. $\mathfrak{I}(y_{i_1}) = 1$ und $\mathfrak{I}(y_{i_2}) = 1$) does not make sense due to the 2-$\forall\forall$-structure.

Copyright (c) IARIA, 2010 ISBN: 978-1-61208-101-4

Then, in the worst case, each of the other branchings needs to be considered.

How does the prefix order influence the algorithm? If y_{i_1} belongs to the leftmost prefix group we may branch immediately. Because we have 2-autarky, the assignment of y_{i_1} leads to a monotone literal y_{i_2} in the reduced formula (which may be pruned, column 2). If this branch does not lead to an abort, columns 3 and 4 need to be considered. Here we must wait with the assignment of y_{i_2} until either allowed by the prefix ordering or a special rule (monotone quantified literal, unit existential clause) applies.

For the case that y_{i_1} does not belong to the leftmost prefix group we may bookmark the 2-autarky for later consideration.

$$\begin{array}{cc|cc|cc|cc|cc}
 & & \multicolumn{2}{c}{\Downarrow} & & & & & \multicolumn{2}{c}{\Downarrow} \\
\bullet & \circ & 1 & & 1 & & 0 & & 0 & \\
\bullet & \bar{\bullet} & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
\bar{\bullet} & \bullet & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0
\end{array}$$

Figure 10: Two branchings for 2-∀∃-autarky.

For the case of figure 9(b) on the preceding page we have a different situation. Here also three different clause types need to be considered, but with only two meaningful branchings. Just like for figure 9(a) we do not need to branch as described in column 1. This also holds for column 4 due to the structure of column 3. Furthermore the same remarks as above apply with regard to the moment we are allowed to branch.

Case 3: 2-∀∃-autarky $\{y_{i_1}, x_{i_2}\}$, with y_{i_1} ∀-quantified and x_{i_2} ∃-quantified, and $y_{i_1} < x_{i_2}$. We look at the preceeding example: In case $\Im(y_{i_1}) = 1$, then x_{i_2} becomes monotone, and only branching for column 1 is necessary. If $\Im(y_{i_1}) = 0$, also because of monotony only column 4 needs to be considered.

Case 4: 2-∃∀-autarky $\{x_{i_1}, y_{i_2}\}$, with x_{i_1} ∃-quantified, y_{i_2} ∀-quantified, and $x_{i_1} < y_{i_2}$. For a structure analogous to figure 9(b) on the previous page we only need to branch for columns 2 and 3 (similar to the 2-∀∀-autarky).

$$\begin{array}{cc|cc|cc|cc|cc}
 & & & & & & \multicolumn{4}{c}{\overbrace{\hphantom{0\quad 0\quad 0\quad 0\quad 0}}} \\
\bar{\bullet} & \circ & 1 & & 1 & & 0 & & 0 & \\
\circ & \bullet & & 1 & & 0 & & 1 & & 0 \\
\bullet & \bullet & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
\bullet & \bar{\bullet} & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1
\end{array}$$

Figure 11: Two branchings for not-2-autarky.

All four cases have in common, that one reduction occurs because of the rule monotone literal. This may ease the later implementation. Also, not-autarky allows for simplifications as well, as shown in column 3 and 4 of figure 11.

VI. Changes to QBF Data Structures

From existing experiments with the rule monotone literal for SAT formulas it is known [7], that this rule does not lead to considerable improvements and is usually left out. For SAT solvers which use the described data structure the heuristic (and its computation cost) has considerable implications on the overall practical runtime of the solver. For QBF formulas the heuristic has less choice due to the prefix, on the other hand a wrong choice has stronger implications. Here we consider how to practically implement the proposed considerations by integrating them into our data structure.

For a quantified k-CNF clause α with n variables and m clauses the data structure requires $\mathcal{O}(k \cdot 2n + k \cdot m)$ space so far: each literal L has a field of length k that counts occurrences of L in clauses of length $1, \ldots, k$, where $k \cdot m$ is the size of the matrix. Therefore, the runtime of the lexicographic heuristic is $\mathcal{O}(k \cdot 2n)$. For QBF formulas the heuristic requires $\mathcal{O}(k \cdot 2 \cdot |Z_1|)$ time, where Z_1 denotes the leftmost prefix group.

In order to consider structural information for a given variable pair (z_1, z_2) for the heuristic `Choose-Literal()`, a field s of length 8 is used for each relevant combination, which counts the occurrence of the pairs (z_1, z_2) in the clauses of the formula. Only combinations of pairs are relevant that occur at least once together in a clause. For example: if x_1 and y_2 occur only as $(\ldots \vee x_1 \vee \ldots)$ or $(\ldots \vee \neg x_1 \vee \ldots)$ and $(\ldots \vee y_2 \vee \ldots)$ or $(\ldots \vee \neg y_2 \vee \ldots)$, the removal of a clause containing x_1 or $\neg x_1$ may not lead to y_2 becoming monotone. Therefore, structural information of at most $\mathcal{O}(k^2 \cdot m)$ many of $\mathcal{O}(n^2)$ possible pairs needs to be kept: for each of the m clauses only up to $\frac{k \cdot (k-1)}{2}$ new, so far not considered pairs may be introduced.

The fact that the set of $\mathcal{O}(k^2 \cdot m)$ many field addresses is static for a given input formula may be used for accelerated access: A supporting data structure must provide for fast access for any given variable pair. A possible solution for this is to use corresponding *hash functions*.

When using hashing, a set of keys $S \subseteq \mathcal{U}$ with universe $\mathcal{U} = \{1, \ldots, N\}$ and $|S| \ll |\mathcal{U}|$ is mapped to numbers $0, \ldots, t-1$ with $t \geq s := |S|$. Here, a hash function $h : \mathcal{U} \to \{0, \ldots, t-1\}$ is used. In case $h_{|S}$ is one-to-one, we call it a *perfect* hash function, which is per definitionem collision free. We cite from [10] the result: for every $t \geq 3 \cdot s$ there exists a perfekt hash function, which can deterministically be computed in time $\mathcal{O}(s \cdot N)$ and probabilistically in time $\mathcal{O}(s)$ and whose execution requires time $\mathcal{O}(1)$. One such hash function is described by a program with $\mathcal{O}(s \cdot \log N)$ bits. For a short overview on the subject we refer to [11].

Aside from hash functions we may also consider a hybrid data structure like the *trie* or *prefix trie* for our purposes. So [12] describes a variant of such a trie data structure suitable for storing a set $S \subseteq \mathcal{U}$, where $s := |S|$ and $N := |\mathcal{U}|$ are as above, with $\mathcal{O}(s)$ memory slots with $\mathcal{O}(\log s)$ bits each and worst-case access of $\mathcal{O}\big(\log_s(N)\big)$. For our case ($r$ in $r \cdot n = m$ denotes the ratio of clauses to variables) $N = n^2$

is polynomial in $s = k^2 \cdot m = k^2 \cdot r \cdot n$, which leads to the access time of $\mathcal{O}\big(\log_s(N)\big) = \mathcal{O}\big(\frac{\log N}{\log s}\big) = \mathcal{O}(1)$. The layout of the data structure requires some effort, therefore for first experiments a method is proposed that uses probabilistic methods to calculate an adequate hash function for the given instance.

The information on the structure of a given variable pair (z_1, z_2) may now be managed as follows: First for all relevant variable pairs corresponding memory is allocated (access time $\mathcal{O}(1)$) and the corresponding counters are initialized with 0; then, for each of the m clauses and therein for each of the $\mathcal{O}(k^2)$ pairs the corresponding 8 structure counters are updated. To later allow a clause $(L_1, \ldots, L_{\ell-1}, L_\ell)$ of length ℓ to be shortened by setting a literal to `false`, $\ell - 1$ counters must each be decreased by 1. In case a clause of length ℓ is removed, $\frac{\ell \cdot (\ell-1)}{2}$ many counters need to be decreased by 1. Accordingly, these operations have to be reverted in reverse order when returning from a lower recursion level. The relevant structural classes necessary to identify a 2-autarky as such may be deposited in tabular form, where the table can be generated during compile time.

Altogether, the space requirements for the data structure is increased from $\mathcal{O}(k \cdot 2n + k \cdot m)$ to $\mathcal{O}(k \cdot 2n + k^2 \cdot m)$. The time requirement to shorten a clause is still $\mathcal{O}(k)$, whereas the removal of a clause leads to the changed runtime requirement of $\mathcal{O}(k^2)$.

VII. Conclusion and future work

The strength of the described SAT data structure may also be observed for its extension to QBF: because of a small memory footprint along with its operation, formulas of considerable size fit into the CPU data cache, with small runtime requirements for elemenary operations. Up to date, recent QBF solvers in contrast to recent SAT solvers can only cope with comparatively small randomized instances of quantified boolean formulae [13], which shows the benefits of a compact data structure.

Therefore a paramatrized analysis of 2-autarky based SAT reductions seems promising to identify measures that significantly purge the QBF search tree. Also subject of future examinations is the analysis how to efficiently integrate and parametrize these SAT reductions with other implemented reductions (like trivial truth or trivial falsity), while still keeping the memory footprint of the corresponding QBF data structure small.

To the best of our knowledge, no research has been undertaken yet to utilize the detection of 2-autarky structures for pruning the search tree of existing QBF solvers.

VIII. Acknowledgments

The author would like to thank Ewald Speckenmeyer, Stefan Porschen, and Bert Randerath for many fruitful discussions and the reviewers who helped improve the original manuscript.

References

[1] G. Gopalakrishnan, Y. Yang, and H. Sivaraj, “QB or Not QB: An Efficient Execution Verification Tool for Memory Orderings,” in *Proceedings of the 16th International Conference on Computer Aided Verification (CAV 2004)*, 2004, pp. 401–413.

[2] M. Mneimneh and K. Sakallah, “Computing Vertex Eccentricity in Exponentially Large Graphs: QBF Formulation and Solution,” in *Proceedings of the 6th International Conference on Theory and Applications of Satisfiability Testing (SAT 2003)*, 2003, pp. 411–425.

[3] J. Rühmkorf, “Entwicklung eines leistungsfähigen Lösers für Quantifizierte Boolesche Formeln,” Master’s thesis, University of Cologne, 2005.

[4] H. Kleine Büning and T. Lettman, *Propositional Logic: Deduction and Algorithms*. Cambridge University Press, 1999.

[5] M. Davis and H. Putnam, “A Computing Procedure for Quantification Theory,” *Journal of the ACM*, vol. 7, no. 3, pp. 201–215, Mar. 1960.

[6] M. Cadoli, A. Giovanardi, and M. Schaerf, “An Algorithm to Evaluate Quantified Boolean Formulae,” in *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI 1998)*, Madison, WI, 26.–30. Jul. 1998, pp. 262–267.

[7] M. Böhm and E. Speckenmeyer, “A Fast Parallel SAT-Solver – Efficient Workload Balancing,” *Annals of Mathematics and Artificial Intelligence*, vol. 17, pp. 381–400, 1996.

[8] R. Feldmann, B. Monien, and S. Schamberger, “A Distributed Algorithm to Evaluate Quantified Boolean Formulae,” in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI 2000)*. Austin, TX: American Association of Artificial Intelligence, 30. Jul. – 3. Aug. 2000, pp. 285–290.

[9] B. Monien and E. Speckenmeyer, “Solving Satisfiability in less than 2^n Steps,” *Discrete Applied Mathematics*, vol. 10, no. 3, pp. 287–295, Mar. 1985.

[10] M. L. Fredman, J. Komlós, and E. Szemerédi, “Storing a Sparse Table with O(1) Worst Case Access Time,” *Journal of the ACM*, vol. 31, no. 3, pp. 538–544, Jul. 1984.

[11] K. Mehlhorn and A. K. Tsakalidis, “Data Structures,” in *Handbook of Theoretical Computer Science, Volume A: Algorithms and Complexity*, J. van Leeuwen, ed. Amsterdam: Elsevier, 1990, pp. 301–342.

[12] R. E. Tarjan and A. C.-C. Yao, “Storing a Sparse Table,” *Communications of the ACM*, vol. 22, no. 11, pp. 606–611, Nov. 1979.

[13] “The Third Competitive Evaluation of QBF Solvers,” 2008, http://www.qbflib.org/, last accessed 1. Jul. 2010.

Asymptotic Bounds on Minimum Number of Disks Required to Hide a Disk

Nataša Jovanović*, Jan Korst†, Zharko Aleksovski† and Radivoje Jovanović‡
*Department of Mathematics and Computer Science
Eindhoven University of Technology, Eindhoven, The Netherlands
E-mail: n.jovanovic@tue.nl
†Philips Research Europe, Eindhoven, The Netherlands
E-mail: jan.korst@philips.com
‡Gimnazija u Lebanu, Lebane, Serbia

***Abstract*—We consider the problem of blocking all rays emanating from a closed unit disk with a minimum number of closed unit disks in the two-dimensional space, where the minimum distance from a disk to any other disk is given. We study the asymptotic behavior of the minimum number of disks as the minimum mutual distance approaches infinity. Using a regular ordering of disks on concentric circular rings we derive an upper bound and prove that the minimum number of disks required for blocking is quadratic in the minimum distance between the disks.**

***Keywords*-asymptotic bounds; blocking set; hiding disk.**

I. Introduction

Let U be a closed unit disk, i.e., a disk with radius 1, in the two-dimensional plane and let $\mathcal{R}$ denote the set of all rays that emanate from U. A ray $r \in \mathcal{R}$ is said to be blocked by a disk δ if r and δ have a non-empty intersection. A set $\mathcal{D}$ of closed unit disks, with $U \notin \mathcal{D}$, is called a *blocking set* if every ray $r \in \mathcal{R}$ is blocked by a disk in $\mathcal{D}$. In addition, a blocking set $\mathcal{D}$ is called *d-apart* if the distance between each pair of disks in $\mathcal{D} \cup \{U\}$ is at least d, where distances are measured from center to center.

Minimum Cardinality Blocking Set Problem. *Given d, what is the minimum cardinality N_d of a d-apart blocking set?*

More specifically, we are interested in the asymptotic behavior of N_d, as d tends to infinity. For reasons of convenience, we focus on the following problem, which is equivalent to the minimum cardinality blocking set problem.

Maximum Distance Blocking Set Problem. *Given N unit disks, what is the maximum distance d for which the disks may form a d-apart blocking set?*

Motivation. The problems considered in this paper are related to occlusion problems in table-top interaction devices, where multiple sensors, for example, light sensors or cameras, scan the two-dimensional plane just above the table's surface for objects like game pieces or fingers. A circular object emitting light in that plane cannot be "seen" by the sensors, in other words, it is no longer visible if all rays emanating from it are blocked by other circular objects, for example. The results presented in this paper give a valuable insight on the number of objects required for one such occlusion problem to occur. In other words, we explored one of the "limitations" of the described technology for object detection and by presenting the results, we showed that the occlusion problems can be easily avoided in practice, using a small number of objects, for instance, or designing an application in such a way, that it does not allow objects to be relatively close one to another.

Our Contributions. In this paper we show that both upper and lower bounds on the minimum number N_d of disks are quadratic in d, i.e., we prove that $N_d = \Theta(d^2)$. In more detail, we first show that $N \geq 6$ disks can be positioned such that they form a 2-apart blocking set. The disks of that blocking set are placed on a circle concentric to U with neighboring disks being mutually tangent. We present a simple algorithm of pushing the disks towards the center of U such that the blocking of rays is preserved. The algorithm provides a regular ordering of disks on concentric circular rings such that the disks form a d-apart blocking set, where $d > 2$. This is used to show that

$$\frac{\pi^2}{16} \leq \lim_{d\to\infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{2},$$

where the lower bound is derived as an immediate consequence of the existing lower bound in [7].

Related Work. Jovanović, Korst and Janssen [6] consider a variant of the above blocking set problem, where they consider blocking all lines intersecting a given unit disk, instead of blocking all rays emanating from a given unit disk. The authors presented upper and lower bounds for small values of the minimum mutual distance d between the disks, namely, for $2 \leq d \leq 4$. Jovanović et al. [7] show that the minimum number of unit disks needed to block all rays emanating from a single point is quadratic in d. In addition, we refer to Fulek, Holmsen and Pach [3], who focus on hitting a maximum number of disks with one ray from an arbitrary point, while we aim at blocking all rays emanating from a given disk with a minimum number of disks. The problem of our interest is also related to the

ISBN: 978-1-61208-101-4

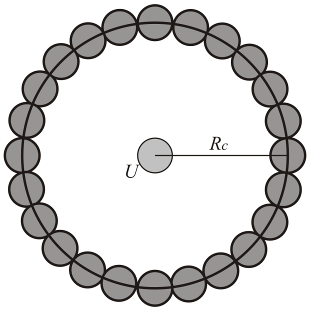


Figure 1. 24 disks positioned on the circle of radius R_c concentric to U.

work of Dumitrescu and Jiang [1] and Mitchell [2], where the authors consider an illumination problem for maximal disk packings by proving the existence of points that are not visible from outside a disk packing. We are not aware of other work that is closely related, although there are many more remotely related visibility problems; see e.g. Chapter 28 on visibility by O'Rourke in [8] or the work presented in [9], [10], [11]. For further details on object detection on related table-top devices we refer to [4], [5].

Overview. The rest of the paper is organized as follows. In Section II we present a construction of a 2-apart blocking set and a method that transforms the constructed blocking set into a d-apart blocking set. In Section III we introduce an ordering of disks on circular rings with which we maximize the distance d between the disks of the blocking set, and we present a simple algorithm that for a given number of disks determines the described ordering. Section IV gives upper and lower bounds on the minimum number of disks required to hide a disk. We conclude the paper with the discussion in Section V.

II. Blocking rays

In this section we propose an ordering of disks that enables blocking all rays from $\mathcal{R}$ for a given number N of disks. We assume for convenience that $N = 6n$. The N disks are placed on a circle c concentric to the given disk U, such that the centers of the disks are on the circle c and there is no gap between neighboring disks; see Figure 1.

More precisely, two neighboring disks positioned on c are mutually tangent. The radius R_c of circle c is easily derived from $R_c = 1/\sin\frac{\pi}{6n}$. Given the mutual tangency of each pair of neighboring disks, one can easily see that any ray $r \in \mathcal{R}$ is blocked by at least one and at most two disks of the given set of $6n$ disks. Hence, these disks form a blocking set. The distance between the neighboring disks on c is 2, while the distance between U and a disk from the blocking set is at least 2 for any $n \geq 1$. Therefore, the constructed blocking set is 2-apart. Let this blocking set be denoted by $\mathcal{D}_2$.

For the maximum distance blocking set problem, we are interested in the maximum distance d for which the $6n$ disks form a d-apart blocking set for $\mathcal{R}$. As such, the problem appears to be hard: constructing a d-apart blocking set for an arbitrary d is certainly challenging, because it requires proving that a set of N disks is a blocking set. Therefore, we focus on *transforming* the constructed 2-apart blocking set into a d-apart blocking set.

In order to transform $\mathcal{D}_2$ into a d-apart blocking set, with $d > 2$, the disks of $\mathcal{D}_2$ should be separated from each other, while the blocking of all rays should be preserved. Let us next describe one step of the proposed transformation.

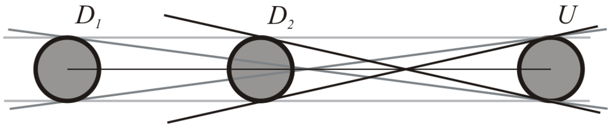


Figure 2. Each ray blocked by D_1 is also blocked by D_2.

Let D_1 and D_2 be two unit disks such that their centers and the center of the given disk U are collinear and D_2 is between U and D_1; see Figure 2. Let $\mathcal{R}_1$ and $\mathcal{R}_2$ denote the sets of rays blocked by disks D_1 and D_2, respectively. Since each ray r that is blocked by D_1 is also blocked by D_2, as shown in [6], we can conclude that $\mathcal{R}_1 \subset \mathcal{R}_2$.

Hence, *the rays blocked by a given disk D are still blocked by D after the disk is moved towards the center of* U, i.e., along the line segment that connects the two disks' centers. Consequently, a transformation of the blocking set $\mathcal{D}_2$ where some disks of $\mathcal{D}_2$ are shifted from their original position on circle c towards the center of U represents a transformation into a d-apart blocking set, where d is the minimum of all pair-wise distances between the disks; see Figure 3. The problem of interest to us now is to determine the maximum d for which we can transform $\mathcal{D}_2$ into a d-apart blocking set.

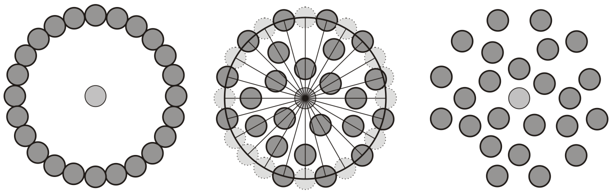


Figure 3. Transformation of $\mathcal{D}_2$ into a d-apart blocking set.

III. Ordering disks on circular rings

In Section II we proved that we can construct blocking sets by pushing the disks of $\mathcal{D}_2$ into the interior of the circle c, given that the disks are moved in the direction of the center of c. In this section we propose a regular ordering of disks forming a blocking set that can be obtained as follows.

Let $\mathcal{D}_2$ be the 2-apart blocking set constructed as in Section II, consisting of $6n$ disks. In the interior of the circle c we can define a number of circles called *rings* and

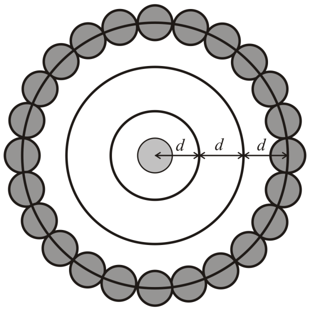


Figure 4. The definition of three circular rings with radii d, $2d$ and $3d$.

denoted as $c_1, c_2, \ldots, c_k$, where the radius of the ring c_1 is d, the radius of c_2 is $2d$, etc. The last ring c_k with the radius kd is assumed to be the given circle, which has radius $R_c = 1/\sin\frac{\pi}{6n}$; see Figure 4. In the process of shifting the disks of $\mathcal{D}_2$ towards the center, we place the center of each of them exactly on one of the rings.

The line segment that connects the center of a disk in $\mathcal{D}_2$ and the center of U is called a *thread*. Thus, the disks of $\mathcal{D}_2$ define $6n$ threads. Since we chose to place the disks on the rings and the disks can be moved only along their threads, each disk can be placed in one of the k intersection points of its thread and the k rings. Note that the d-apart rings ensure that the distance between any two disks positioned on *different rings* is at least d. However, choosing an arbitrary ring for each disk may result in two disks of the *same ring* being less than distance d apart; see Figure 5.

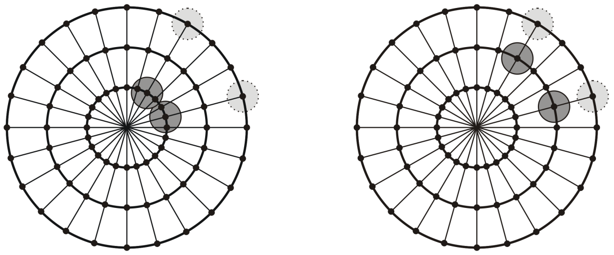


Figure 5. Shifting two disks onto inner rings: left, the disks are not d-apart, and right, the disks are d-apart.

The number k of rings determines the distance d for given n. Given that the radius of the largest ring is $R_c = 1/\sin\frac{\pi}{6n}$ and as we mentioned above $R_c = kd$, we have that

$$d = \frac{1}{k \sin\frac{\pi}{6n}}. \tag{1}$$

Hence, in order to maximize the distance d, we need to minimize the number k of rings needed, for $6n$ disks to form a d-apart blocking set.

For a ring of given radius, it is easy to determine the maximum number of disks that can be positioned equally spaced, such that the distance between two neighboring disks on this ring is at least d. For example, at most 6 disks can be placed on the first ring, at most 12 disks on the second ring, at most 18 disks on the third ring, etc. In this way, we can easily derive a lower bound on the minimum number k of rings needed, for a given n. However, the minimum number of rings that suffices for disks to form a d-apart blocking set is often larger than this lower bound. This is because of the restriction of fixed positions for placing the disks, which does not always allow placing the maximum number of disks on the rings. In the construction we propose, we place less than maximum disks on some of the rings or even keep some of the rings empty.

In more detail, we choose to place $6n_j$ disks on the j-th ring, where

$$n_j = 2^{\lfloor \log_2 j \rfloor}, \tag{2}$$

such that the disks form a regular polygon. Note that $6n_j$ is *equal* to the maximum number of disks that can be placed, only for the rings $j = 2^l$, for some $l \geq 0$, however, it is *less* than maximum for all other rings; see the comparison given in Table I. For symmetry reasons, we focus on one of the six sections of $\mathcal{D}_2$ with n disks. We show that any set of n disks can be split into k subsets, where the j-th subset contains either $2^{\lfloor \log_2 j \rfloor}$ or 0 disks. The j-th subset is then placed on the j-th ring such that the distance between each two disks is at least d. More precisely, we show that the given number n can be represented as

Ring j	1	2	3	4	5	6	7	8
Max disks	6	12	18	24	30	36	42	48
$6n_j$	6	12	12	24	24	24	24	48

Table I
THE MAXIMUM NUMBER OF DISKS AND THE CHOSEN NUMBER OF DISKS FOR RINGS 1 TO 8.

$$n = \bar{n}_1 + \bar{n}_2 + \cdots + \bar{n}_k, \tag{3}$$

where $\bar{n}_j \in \{0, n_j\}$, or simplified, any natural number n can be represented as

$$n = b_0 + \underbrace{2+2}_{max\ 2} + \underbrace{4+4+4+4}_{max\ 4} + \cdots + \underbrace{2^t + 2^t + \cdots + 2^t}_{max\ 2^t}, \tag{4}$$

for some $t \geq 0$ and $b_0 \in \{0, 1\}$. Note that the total number of addends in (3) is k, i.e., each addend corresponds to a ring, more precisely, to the number of disks placed on each of the six sections of the ring. This results in including the zero-addends in counting, since they indicate the presence of empty rings. More precisely, we include the zero-addends in counting when we have less than the maximum number of equal addends, for all addends except for the largest ones. For example, $n = 15$ can be represented as $15 = 1 + 2 + 0 + 4 + 4 + 4$ and the number of rings needed is $k = 6$, with the third ring being empty.

Formally, we prove the existence of a representation of n in form (4), using the following lemma.

Lemma 1: For any positive integer n a sequence $A_n = (a_0, a_1, \ldots, a_t)$ exists such that

$$n = \sum_{i=0}^{t} a_i \cdot 2^i \qquad (5)$$

where $0 \leq a_i \leq 2^i$ and $a_t > 0$.

Proof: The proof of the lemma follows from the binary scale representation of n. ■

For a given n, there are generally multiple sequences A_n. From Equation (1), to construct a d-apart blocking set, where distance d is as large as possible, we need to minimize the number k of rings. The number of rings we define is equal to the number of addends in (4). Hence, the number k of rings is given by

$$k = (1 + 2 + 4 + 8 + \cdots + 2^{t-1}) + a_t = 2^t - 1 + a_t \quad (6)$$

where a_t is the number of addends of size 2^t in (4). Hence, our interest is in the sequences A_n^* for which $2^t + a_t$ is minimal.

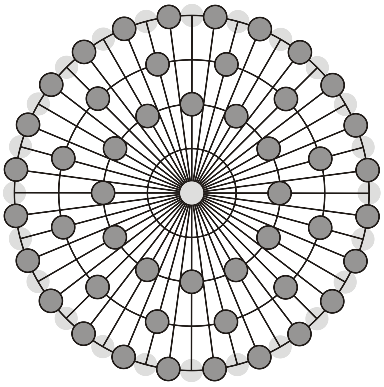


Figure 6. An example of a d-apart blocking set for $n = 8$, where $d \approx 4$.

A. Disk ordering algorithm

In the previous section we showed how to determine the number of rings and the number of disks on each of them, using Lemma 1 and choosing the sequence A_n^* for which the number of rings is minimal. In this section, we present an algorithm that given the sequence A_n^*, for each disk of $\mathcal{D}_2$ determines the ring on which it should be placed, which results in the disks forming a d-apart blocking set; see Figure 6.

We restrict ourselves to finding the solutions for all n that are divisible by their largest addend 2^t in the representation (4). Note that $2^t | n$ implies that $n_j | n$, for all j.

Let us define a table T with k rows and n columns, such that each thread corresponds to one column of T and each ring corresponds to one row of T, with the outermost ring corresponding to the top row. Each cell of the table T then represents a position on which the corresponding disk can be placed, i.e., it is the intersection of its thread and a ring. When one disk is moved to a certain position, the value in the corresponding cell of T is set to 1 or "full", while the other cells of the same column have values 0 or "empty"; see Figure 7. The defined table represents one of the six identical sections of the blocking set, thus, we consider the table as if its columns are cyclic (its first and its last column are connected).

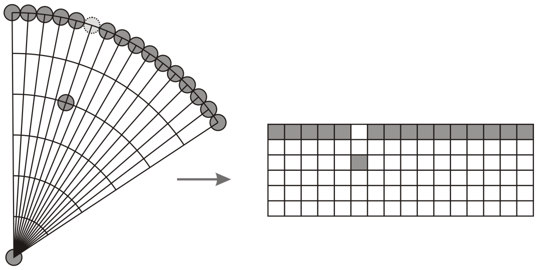


Figure 7. A set of 16 disks with 6 rings and the corresponding 6 x 16 table.

An ordering of full cells in a table T is called *valid* if and only if the following conditions hold:

- There is exactly one full cell in each column;
- The j-th row is either empty or it contains exactly n_j full cells;
- The number of empty cells between any two successive full cells of the j-th row is exactly $\frac{n}{n_j} - 1$.

Lemma 2: A valid table T exists for any positive integer n represented by (4) for which $2^t | n$.

Proof: The proof of the lemma is given by a method for constructing a valid table, which follows from the equation $2^m = 2^{m-1} + 2^{m-1}$. In more detail, a complete row of full cells can be split into $n/2^t$ rows, where each row contains 2^t full cells, as illustrated in Figure 8. Each of the resulting rows can again be split into two rows, by pushing every second full cell to a new row. After a finite number of ”splitting” steps, each row corresponds to a non-zero addend in representation (4). The rows can be swapped then if necessary, such that each row r that is directly above a row r' contains at least the same number of full cells as r'. The process is completed by inserting empty rows where needed. ■

Note that the proof of Lemma 2 represents a *disk ordering algorithm* that for each of the n disks determines the ring on which it should be placed, such that the disks form a d-apart blocking set.

IV. Upper and lower bounds

In Sections II and III, we showed that we can construct a d-apart blocking set for each n that is divisible by its largest addend in representation (4). In this section, we present upper and lower bounds on the cardinality N_d of such a

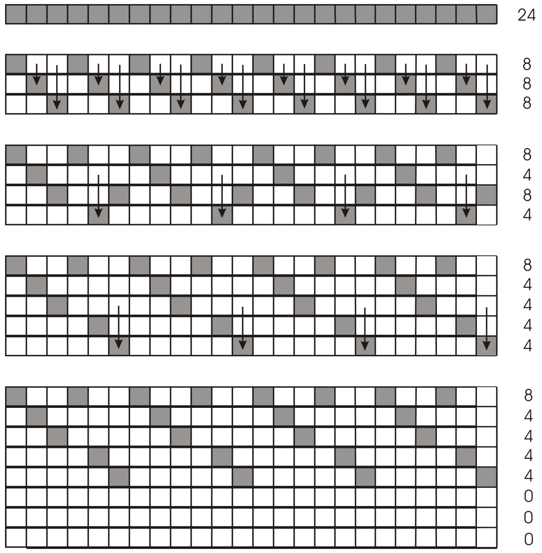


Figure 8. Constructing a valid table for $n = 24$.

blocking set, as a function of the minimum distance d. We start by deriving an upper bound.

One can easily show that the ordering of disks presented in Section III-A implies that the minimum of all pairwise distances between the disks is d. The relation between the distance d, the given number n and the corresponding number k of rings is given by

$$d = \frac{1}{k \sin \frac{\pi}{6n}} \tag{7}$$

From the choice of sequence A_n^* in Lemma 1, for which $a_t + 2^t$ is minimal, we have that

$$\sum_{j=0}^{t-1} 2^{2j} + (a_t - 1) \cdot 2^t \leq n \tag{8}$$

where a_t is the number of largest addends 2^t in representation (4). From (8) and

$$\sum_{j=0}^{t-1} 2^{2j} = \frac{1}{3}(4^t - 1) \tag{9}$$

it follows that

$$4^t + 3(a_t - 1)2^t \leq 3n + 1 \tag{10}$$

With further transformations of inequality (10) we have

$$\begin{aligned} ((2^t)^2 + 2(a_t - 1)2^t) + (a_t - 1)2^t &\leq 3n + 1 \\ \Leftrightarrow k^2 + (a_t - 1)(2^t - a_t + 1) &\leq 3n + 1 \quad (11) \end{aligned}$$

Since $1 \leq a_t \leq 2^t$, we have that

$$(a_t - 1)(2^t - a_t + 1) \geq 0 \tag{12}$$

Finally, from (11) and (12), we bound the number k of rings by a function in n as follows.

$$k \leq \sqrt{3n + 1} \tag{13}$$

We transform (7) into

$$\frac{1}{kd} \leq \sin \frac{\pi}{6n} \tag{14}$$

and multiply (13) by $\sqrt{n}$

$$k\sqrt{n} \leq \sqrt{3n^2 + n} \tag{15}$$

Multiplication of (14) and (15) and expressing the limit for $d \to \infty$, results in

$$\lim_{d \to \infty} \frac{n}{d^2} \leq \frac{\pi^2}{12} \tag{16}$$

and since $N = 6n$, we derived an upper bound on N_d, i.e.,

$$\lim_{d \to \infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{2} \tag{17}$$

In [7], the authors proved that the lower bound on the minimum number of disks which form a d-apart blocking set for the set of all rays emanating from a single point is $\frac{\pi^2}{16} d^2$, as d tends to infinity. To block the rays emanating from a given unit disk we need at least as many as to block the rays emanating from its center. Hence, the lower bound on the minimum number N_d of disks is given by

$$\lim_{d \to \infty} \frac{N_d}{d^2} \geq \frac{\pi^2}{16}. \tag{18}$$

Combining the results of (17) and (18), we proved the following theorem.

Theorem 1: For the minimum cardinality N_d of a d-apart blocking set to block all rays emanating from a unit disk we have

$$\frac{\pi^2}{16} \leq \lim_{d \to \infty} \frac{N_d}{d^2} \leq \frac{\pi^2}{2}.$$

V. Conclusion

We expect that both bounds, especially the upper bound, can be further improved. The following discussion provides some directions for potential improvements.

Constructing a d-apart blocking set from $\mathcal{D}_2$ through a sequence of transformation steps where a number of disks is pushed towards the center results in the rather large constant $\pi^2/2$. The disks pushed inside circle c block much larger sets of rays than the sets of rays they block from their original positions on c. Consequently, the sets of rays blocked by two disks on different rings may not be disjoint. This implies that constructing blocking sets for which the overlap of sets of blocked rays is minimized may potentially provide a better upper bound. In addition, the number of disks on one ring is less than the maximum possible number for the majority of rings. Placing the maximum number of disks on each of the rings may further improve the upper bound.

The combination of the last two conjectures may be used to define an optimization problem, similar to the problem of opening a combination lock with k rings, i.e., to find the rotation angle for each of the k rings that are d-apart and contain the maximum number of d-apart disks, such that the disks form a blocking set and the total overlap of blocked rays is minimized. We expect that the solution of this problem provides a better upper bound. The main challenge here is still the problem of proving that a set of disks, positioned following some constraints, is a blocking set for the set $\mathcal{R}$ of all rays.

REFERENCES

[1] A. Dumitrescu, and M. Jiang. *The forest hiding problem.* Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms, 2010, Austin, Texas, USA.

[2] J. Mitchell. *Dark points among disks.* Open Problems from the 2007 Fall Workshop in Computational Geometry, Hawthorne, New York, USA.

[3] R. Fulek, A.F. Holmsen, and J. Pach. *Intersecting convex sets by rays.* Proceedings of the 24th Annual ACM Symposium on Computational Geometry, 2008, College Park, MD, USA, 385–391.

[4] G. Hollemans, T. Bergman, V. Buil, K. van Gelder, M. Groten, J. Hoonhout, T. Lashina, E. van Loenen, and S. van de Wijdeven. *Entertaible: Multi-user multi-object concurrent input.* Adjunct Proceedings: Demonstrations, Proceedings 19th Annual ACM Symposium on User Interface Software and Technology, 2006, Montreux, Switzerland, 55–56.

[5] N. Jovanović, J. Korst, and V. Pronk. *Object Detection in Flatland.* Proceedings of the 3rd International Conference on Advanced Engineering Computing and Applications in Sciences, 2009, Sliema, Malta.

[6] N. Jovanović, J. Korst, and A.J.E.M. Janssen. *Minimum blocking sets of circles for a set of lines in the plane.* Proceedings of the 20th Canadian Conference on Computational Geometry, 2008, Montréal, Canada, 91–94.

[7] N. Jovanović, J. Korst, R. Clout, V. Pronk, and L. Tolhuizen. *Candle in the Woods: Asymptotic Bounds on Minimum Blocking Sets.* Proceedings of the 25th ACM Symposium on Computational Geometry, 2009, Aarhus, Denmark, 148–152.

[8] J. O'Rourke. Visibility, chapter 28 in: J.E. Goodman & J. O'Rourke, *Handbook of Discrete and Computational Geometry*, 2nd Edition, 2004, Chapman & Hall/CRC, 643–664.

[9] M. van Kreveld. *Bold Graph Drawing.* Proceedings of the 21st Canadian Conference on Computational Geometry, 2009, Vancouver, Canada, 119–122.

[10] H. Martini, and V. Soltan. *Combinatorial problems on the illumination of convex bodies.* Aequationes Mathematicae 57, 1999, 121–152.

[11] L. Szabo, and Z. Ujvary-Menyhart. *Clouds of planar convex bodies.* Aequationes Mathematicae 63, 2002, 292–302.

A Novel Multiple Valued Logic OHRNS Adder Circuit for Modulo $(r^n - 1)$

Reza Farshidi
Islamic Azad University, Dezful Branch
Dezful, Iran
farshidi@iaud.ac.ir

Ahmad Habibi Zadnavin
Islamic Azad University, Tabriz Branch
Tabriz, Iran
manhabibi@yahoo.com

Ehsan Gholami
Islamic Azad University,Shoushtar
Shoushtar, Iran
e.gholami@iau-shoushtar.ac.ir

***Abstract*— Residue number system is a carry free and non-weighted number system. This system is appropriate for applications that require fast arithmetic computation. Residue Number System is defined by a moduli set. Selecting the moduli set is an important issue in this number system. Each number in this system is represented by its remainders in moduli set, so it introduces smaller numbers than conventional systems, which results in fast calculation and low power consumption. Multi Valued Logic increases the dynamic range by using same positions rather than binary logic. One Hot Residue Number System is a method, which reduces the delay of arithmetic computations such as addition and multiplication to just one transistor delay. In this paper, a new adder circuit is introduced for modulo (r^n-1) by combining One Hot Residue Number system and Multi Valued Logic, which has significant improvement in terms of number of applied transistors and power consumption in comparison to the ordinary One Hot Residue Number System with Multi Valued Logic.**

Keywords-Residue Number System;One-hot; Multiple Valued Logic; Low Power Circuits; VLSI.

I. INTRODUCTION

Some applications such as digital signal processing require fast computation with low power consumption. Residue Number System (RNS) satisfies these requirements by presenting a weighted number into smaller numbers with carry-free property, which results in parallel arithmetic operation [1][3]. One of the most efficient methods to achieve parallelism in arithmetic computation in VLSI digital systems is applying RNS. VLSI digital systems can be designed with smaller chip area, lower power consumption and more speed using RNS rather than conventional numeric systems.

Each RNS is based on moduli set, which consists of a set of relatively prime integers.

One of the most applied modulo in moduli sets is $(a^b - 1)$. Some well-known moduli sets which use this modulo are $\{r^a - 1, r^b, r^c + 1\}$, $\{2^{2n-1}, 2^{2n+1} - 1, 2^{2n+1} + 1\}$ $\{2^n - 1, 2^n, 2^{n+1} - 1\}$, $\{2^n - 1, 2^n, 2^n + 1\}$, etc.

In this paper, an adder circuit for modulo $(r^n - 1)$ is proposed, which has high speed and low power consumption. One Hot RNS (OHRNS) is a method that has lowest delay for addition and multiplication operations in RNS moduli. Since applying OHRNS is normally associated with using of Multi Valued Logic (MVL) and large number of transistors, so the VLSI circuit design based on OHRNS requires high power consumption and large area. The designed circuit in this paper decreases the number of transistors, which decreases power consumption, area and delay-power product (DP-product) significantly in OHRNS for $(r^n - 1)$ modulo adder.

The rest of this paper is organized as follow: Section II describes the necessary background. The proposed circuit is discussed in Section III. Sections IV and V contain the performance evaluation and conclusion respectively.

II. BACKGROUND

A. *Residue Number System(RNS)*

Residue Number System is specified by moduli set like $(m_1, m_2, ..., m_n)$ in which all the moduli are positive integers. If all the modulus be relatively pair wise prime the system will have the largest possible dynamic range which equals [α , α +M) in which α is an integer and M is:

$$M = \prod_{i=1}^{n} m_i \tag{1}$$

An integer X is represented in the Residue Number System by an n-tuple $(x_1, x_2, ..., x_n)$ where x_i a non negative integer is satisfied by:

$$x_i = X \bmod m_i \tag{2}$$

In RNS, arithmetic computations such as Addition, subtraction and multiplication for two given integer numbers X and Y, which are represented by $(x_1, x_2, ..., x_n)$ and $(y_1, y_2, ..., y_n)$ in moduli set $\{m_1, m_2, ..., m_n\}$, are as follow:

Assume $Z = X \circ Y$ and $Z = (z_1, z_2, ..., z_n)$ where $\circ$ operand is one of the noticed operations. For $1 \le i \le n$:

$$z_i = (x_i \circ y_i) \bmod m_i \tag{3}$$

z_i s can be calculated in parallel without any dependency between them, which leads to high speed computations in RNS. To convert a residue number $(x_1, x_2, ..., x_n)$ into its binary representation *X*, the Chinese Remainder Theorem is widely used. In CRT, the binary number *X* is computed by:

$$X = \left| \sum_{i=1}^{n} N_i \left| N_i^{-1} \right|_{m_i} x_i \right|_M \tag{4}$$

where $N_i = \frac{M}{m_i}$ and $\left| N_i^{-1} \right|_{m_i}$ is the multiplicative inverse of $\left| N_i \right|_{m_i}$ [1].

$$\begin{array}{cccc} a_{n-1} & a_{n-2} & .... & a_0 \\ \downarrow & \downarrow & .... & \downarrow \\ r^{n-1} & r^{n-2} & .... & r^0 \end{array}$$

Figure 1. Digits of a number in MVL

B. Multi Valued Logic

Another important feature of RNS is an alternative named multiple valued logic (MVL) that despite of binary system which is limited to just two possible logic states, the number of discrete signal values or logic states extends beyond two. Using MVL can result in more effective usage of silicon resource and circuit interconnections [2]. Therefore, by defining r levels for MVL each position has one of the $(0,1,....,r-1)$ levels. The value of each position in r-level MVL is r^i, where *i* indicates each position. For a given number $A(a_{n-1}......a_0)$ the values of its digits are as shown in Fig. 1.

Since by using high radix in MVL much more information can be stored in each location in compare with binary logic, so speed of arithmetic computations is increased in this logic that has advantages of reduction of chip area, interconnections and increasing of chip performance.

C. One Hot RNS(OHRNS)

Power consumption is one of the most important factors, which is considered in designing the VLSI circuit. Many techniques have been proposed to decrease the power consumption. In many cases decreasing the power consumption results in reduced circuit performance, Therefore designers have focused on circuit design considering a more important factor is called Delay-Power product (DP product).

In one-hot, remainders of each modulo m_i $(0,1,...,m_i-1)$ are represented by a separated line as shown in Fig. 2, where in each moment just one line that is equal to x_i, a remainder of m_i modulo, is active and others are inactive. By changing the input value, the amount of two lines changes at maximum level. Therefore the wasting of power is at minimum level.

OHRNS makes this ability to do arithmetic computations such as addition, subtraction and multiplication rapidly and based on barrel shifters. OHRNS structure can be represented by a state machine.

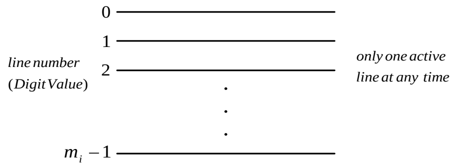


Figure 2. One- Hot representation for remainder of a modulo

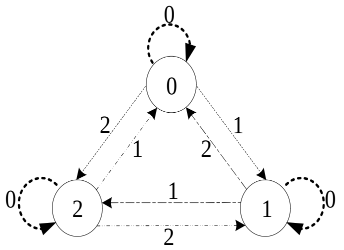


Figure 3. Representation of OHRNS as a state machine

For example for addition in OHRNS there are two operands, which one of them can be considered as the current state of machine and second operand is a shifter that shifts the first operand to the correct output as the final result. As an example, OHRNS state machine for addition in modulo 3 is illustrated in Fig. 3.

Structure of a computational OHRNS for each operation in modulo m_i has been shown in Fig. 4. It has two series of inputs, which are actually two required operands for addition operation. One of them is applied as a shifter and another as a data that must be shifted. In this architecture, transistors play the role of a shifter. More details about this structure have been introduced in [9]. Since modulo (r^n-1) is widely used in moduli sets in RNS, an adder for this modulo based on OHRNS has been designed.

According to above expressions and considering the details of this structure in [9], for adding two numbers in modulo (r^n-1), the required transistors are equal to (r^n-1)^2. However, the delay of this circuit is equal to just one transistor delay. In next section, a method is proposed, which decreases the number of applied transistors considerably. Furthermore, our proposed circuit handles the problem of previous work about OHRNS in which the circuit may encounter the fault and incorrect output due to the using of same unit for producing the addition result and carry-out digit.

III. MVL OHRNS Adder for Modulo (r^n-1)

In this paper, an OHRNS–based adder circuit is proposed for modulo (r^n-1), which has significant improvement in terms of hardware cost and power consumption. Furthermore, its DP is considerably lower in comparison to the previous circuits. Assume that *A* and *B* are two numbers in modulo (r^n-1). Their values are totally in defined below span:

$$\begin{aligned} A &= (a_{n-1}.....a_2a_1a_0), \quad 0 \le A < r^n-1 \\ B &= (b_{n-1}.....b_2b_1b_0), \ \ 0 \le B < r^n-1 \end{aligned} \qquad (5)$$

where a_i and b_i have the values of:

$$0 \le a_i < r$$

$$0 \le b_i < r$$

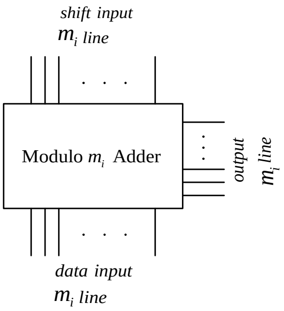


Figure 4. Block diagram of an OHRNS adder

In proposed modulo (r^n-1) adder, number *A* which has *n* digits is represented as two parts. The first part represents first *n/2* digits, which is considered as low significant part and another as high significant part such that:

$$A=(a_{n-1}a_{n-2}.....a_2a_1a_0)_r \rightarrow A=(A_1A_0)_r$$

where:

$$A_0=\begin{cases}(a_{n/2}a_{(n/2)-1}.........a_1a_0)_r & if\ \ n\ is\ even\\ (a_{\lfloor n/2\rfloor}a_{\lfloor (n/2)\rfloor-1}.........a_1a_0)_r & if\ \ n\ is\ odd\end{cases}$$
$$A_1=\begin{cases}(a_na_{n-1}.....a_{n/2}) & if\ \ n\ is\ even\\ (a_na_{n-1}a.....a_{\lfloor n/2\rfloor+1}) & if\ \ n\ is\ odd\end{cases} \qquad (6)$$

In these equations both A_1 and A_0 are numbers in radixes R_0 and R_1, which are equal to:

$$\begin{matrix} R_0,R_1=r^{n/2} & if\ \ n\ is\ even\\ R_0=r^{\lfloor n/2\rfloor},R_1=r^{\lfloor n/2\rfloor-1} & if\ \ n\ \ is\ odd\end{matrix} \qquad (7)$$

In the same way for number *B* we have:

$$B=(b_{n-1}b_{n-2}....b_2b_1b_0)_r \rightarrow B=(B_1B_0)_r$$

B_1 and B_0 are defined as well as A_1 and A_0. In new definition *A* and *B* have two low and high significant parts. Therefore, to add numbers in modulo (r^n-1) the adding method in conventional systems is used in which for adding two n-digits numbers all digits with same position are added together from least significant position to most significant position and the carry is added with the digits in next position. But the carry digit, which is obtained from adding most significant digits, must be added to the resulted number. So, to add *A* and *B* numbers the operation is performed as shown in fig. 5, where *C* is the result of adding *A* and *B* in modulo (r^n-1)**.** The proposed circuit for (r^n-1) modulo adder is implemented by an OHRNS structure as shown in Fig. 6.

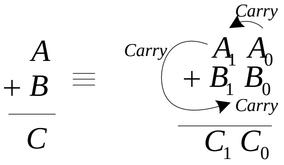
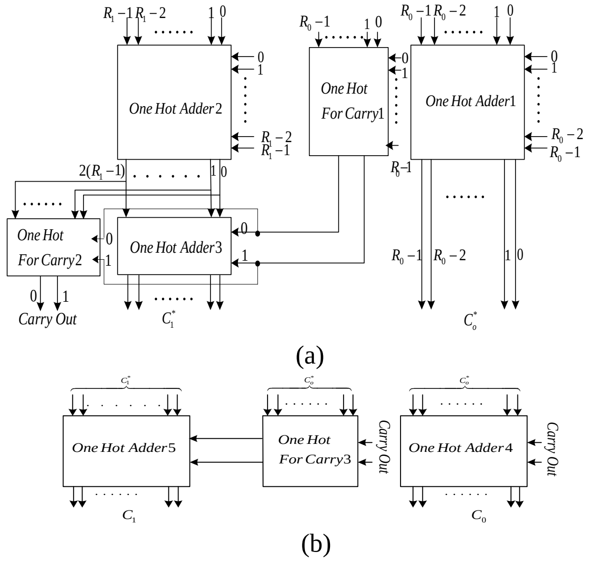


Figure 5. Adding two numbers in modulo (r^n-1)

According to Fig. 6, addition operation once requires carry propagation from low significant part to high significant part and in next step from high significant part to low significant part. The carry in each step is obtained by one Hot for carry unit, which has a structure same to the One Hot adder unit. In spite of previous papers about OHRNS that each One Hot transistor is connected to two outputs, one for carry and another for the result of addition, we do not apply such method because coupling all output results together makes this problem that when value of an output is equal to one, the value of carry digit will be one even if its previous value is zero and it means an incorrect result. Therefore, to avoid such problem a separated One Hot unit is used to produce carry digits.

In this implementation for adding each part of two numbers as well as Fig. 5 a One Hot adder and a One Hot for carry units are used. Since the carry digit is equal to one or zero so, two lines of transistors are used to add it to the next part as shown in fig. 6.a. But as it can be observed in fig. 6.b for obtaining C_1 no One-Hot for carry unit is used because in modulo (r^n-1) the carry digit is just added once to the resulted number. So, it is ignored in the last part of the circuit.



(a)

(b)

Figure 6. Block diagram of OHRNS adder in modulo (r^n-1)

TABLE I. PERFORMANCE EVALUATION

	Hardware		**Delay**
Conventional OHRNS adder for modulo $(r^n - 1)$	$(r^n - 1)^2$ **Transistors**		1 Transistor delay
Proposed OHRNS adder for this modulo	***Even n***	***Odd n***	4 Transistors delay
	$\left((r^{n/2})^2 \times 3\right) + \left((r^{n/2} \times 2) \times 5\right)$	$\left((r^{(n+1)/2})^2 \times 2\right) + (r^{(n-1)/2})^2 +$ $\left(6 \times (r^{(n-1)/2})\right) + \left(4 \times (r^{(n+1)/2})\right)$	

For example assume r and n are 3 and 4 respectively. For modulo (r^n-1) the value is:

$$(r^n - 1) = (3^4 - 1) = 80$$

To represent each number such as A in this modulo, four digits are required and it can be written as:

$$A = (a_3 a_2 a_1 a_0)_m \quad = (A_1, A_0)_r$$

$$A \in \{0,1,2,...,79\} \ and \ a_i \in \{0,1,2\} \ , \ i = 0,1,2,3$$

$$(A_1)_{R_1} = (a_3 a_2)_{R_1} \ , \ R_1 = 3^2 \ \ and \ (A_2)_{R_0} = (a_1 a_0)_{R_0} \quad , \ R_0 = 3^2$$

To make a comparison between conventional OHRNS and proposed method, the number of applied transistors is calculated for r=3 and different values for n. The results have been illustrated in Fig. 7 and Fig. 8. According to the results a significant reduction in applied transistors can be observed by using this method for modulo (r^n-1). The delay of this circuit is just equal to the delay of four transistors.

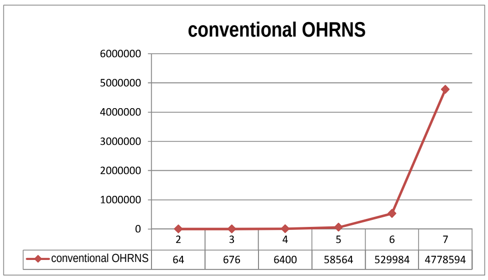


Figure 7.The number of applied transistors in ordinary OHRNS

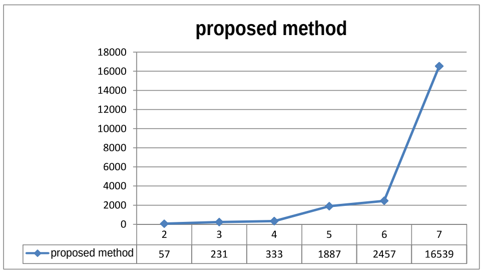


Figure 8.The number of applied transistors using proposed method

V. PERFORMANCE EVALUATION

In this section, the proposed OHRNS adder is compared with conventional OHRNS adder in terms of propagation delay and number of used transistors. In ordinary OHRNS for adding two numbers in a given modulo m, the required transistors are equal to m^2. In proposed method the numbers are divided into two parts that each one is calculated separately by using the One Hot adder unit. In previous work this unit was responsible for producing carry digits too. It should be taken into account that coupling these outputs together may result in incorrect output. To handle the problem another unit is called One Hot for carry has been added to the circuit. As it can be observed in Fig. 6 One hot adder1 and One-hot for carry1 have $R_0 \times R_0$ transistors for each unit and One-hot adder2 has $R_1 \times R_1$ transistors. One-hot adder4 and One-hot for carry3 have 2× R_0 transistors and the remained three units have 2× R_1 transistors. According to the values of R_0, R_1 and r obtained from (7), all required transistors have been calculated for odd and even n. The results of comparison have been shown in Table I. According to them a significant reduction in number of applied transistors is considered; however, the total delay of circuit is increased to four transistors delay, which is negligible versus the huge amount of reduced transistors. By using conventional OHRNS method for modulo (r^n-1), (r^n-1)^2 transistors are required. But as it can be observed in Table I, the proposed OHRNS adder decreases the number of applied transistors considerably. This reduction in number of applied transistors has a remarkable effect on decreasing the DP product.

VI. CONCLUSION

RNS is used widely for high speed arithmetic circuits according to its carry free property. OHRNS is a method which reduces the delay of addition and multiplication operation circuits to the delay of just one transistor; however, it has large power consumption according to the huge number of applied transistors.

In this paper, a novel method for One-Hot adder circuit has been proposed for modulo (r^n-1) which has significant improvements in terms of number of applied transistors and power consumption which, decreases delay-power product factor consequently.

REFERENCES

[1] H. Garner, "The Residue Number System," IEEE Transactions Electronic Computer, Vol. 8, pp.140-147, 1959.

[2] B. Parhami, "RNS Representation with Redundant Residues," *Proc. of the 35th Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, pp. 1651-1655, Nov. 2001.

[3] Y. Wang, X. song, M. Aboulhamid, and H. shen, "Adder Based Residue to Binary Number Converters for $(2^n-1, 2^n, 2^n+1)$," *IEEE Transactions Signal Processing*, Vol. 50, No. 7, pp. 1772-1779, 2002.

[4] M. Abdallah and A. Skavantzos, "On Multi Moduli Residue Number Systems With Moduli of Forms (r^a, r^b-1, r^c+1)," *IEEE Transactions Circuits System I: Regular Paper*, vol. 52, no. 7, July 2005, pp. 1253-1266.

[5] A. Hariri, K. Navi, and R. Rastegar, "A Simplified modulo (2^n-1) Squaring Scheme for Residue Number System," *IEEE International Conference on Computer as a tool*, Nov. 2005, pp. 615-618.

[6] M. Hosseinzadeh, K. Navi, and S. Timarchi, "Design Circuit Residue Number System in Current mode," *14th Iranian Conference of Electrical Engineering*, May 2006, pp. 16-18.

[7] W.A. Chren, Jr., "One-Hot Residue Coding for Low Delay-Power Product CMOS Design," *IEEE Transactions On Circuits And Systems II: Analog And Digital Signal Processing*, Vol. 45, No. 3, Mar. 1998, pp. 1-12.

[8] A. Hariri, K. Navi and R. Rastegar, "A New High Dynamic Range Moduli Set with Efficient Reverse Converter," Elsevier Journal of Computers and Mathematics with Applications, vol. 55, no. 4, pp. 660–668, 2008.

[9] M. Hosseinzadeh, S. J. Jassbi, and K. Navi "A Novel Multiple Valued Logic OHRNS Modular r^n Adder Circuit," International Journal of Electronics, Circuits and Systems, 2007, pp. 245-249.

[10] M. Hosseinzadeh and K. Navi, "A New Moduli Set for Residue Number System in Ternary Valued Logic," Journal of Applied Sciences, 2007, pp. 3729-3735.

[11] A. S. Molahosseini, K. Navi, O. Hashemipour, and A. Jalali, "An Efficient Architecture for Designing Reverse Cconverters Based on a General Three moduli Set," Elsevier Journal of Systems Architecture, In Press, 2008, pp. 929-934