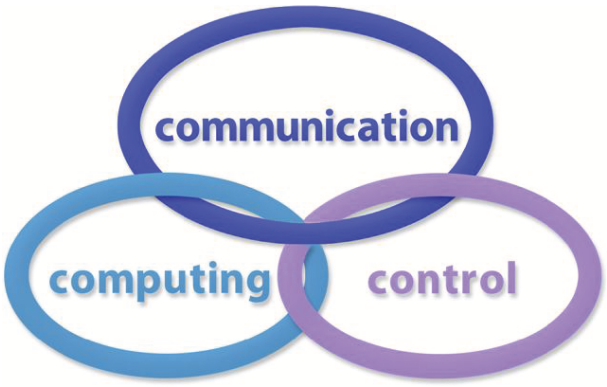
INTERNATIONAL JOURNAL

of

COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836



A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2016 Volume: 11 Issue: 2 (April)

This journal is a member of, and subscribes to the principles of, the Committee on Publication Ethics (COPE).

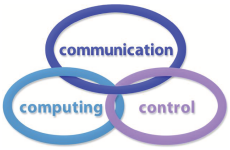
CCC Publications - Agora University

CCC Publications

http://univagora.ro/jour/index.php/ijccc/

BRIEF DESCRIPTION OF JOURNAL

Publication Name: International Journal of Computers Communications & Control.
Acronym: IJCCC; **Starting year of IJCCC:** 2006.
Abbreviated Journal Title in JCR: INT J COMPUT COMMUN.
International Standard Serial Number: ISSN 1841-9836.
Publisher: CCC Publications - Agora University of Oradea.
Publication frequency: Bimonthly: Issue 1 (February); Issue 2 (April); Issue 3 (June); Issue 4 (August); Issue 5 (October); Issue 6 (December).
Founders of IJCCC: Ioan DZITAC, Florin Gheorghe FILIP and Mişu-Jan MANOLESCU.
Logo:



Indexing/Coverage:

- Since 2006, Vol. 1 (S), IJCCC is covered by Thomson Reuters and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.
- Journal Citation Reports (JCR2014 - Science Edition): IF/3 years = 0.746, IF/5 years = 0.739.
  Subject Category:
  - Automation & Control Systems: Q4 (47 of 58);
  - Computer Science, Information Systems: Q3 (96 of 139).
- Since 2008 IJCCC is covered by Scopus (2014: SNIP = 1.029, IPP = 0.619, SJR = 0.450).
  Subject Category:
  - Computational Theory and Mathematics: Q3;
  - Computer Networks and Communications: Q2;
  - Computer Science Applications: Q2.
- Since 2007, 2(1), IJCCC is covered in EBSCO.

Focus & Scope: International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computer and control from the universities, research units and industry.

To differentiate from other similar journals, the editorial policy of IJCCC encourages the submission of original scientific papers that focus on the integration of the 3 "C" (Computing, Communication, Control).

In particular the following topics are expected to be addressed by authors:

- Integrated solutions in computer-based control and communications;
- Computational intelligence methods (with particular emphasis on fuzzy logic-based methods, ANN, evolutionary computing, collective/swarm intelligence);
- Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

Editorial Address:
Agora University/ R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piata Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032
E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Journal website: http://univagora.ro/jour/index.php/ijccc/

Gheorghe Păun
Institute of Math. of Romanian Academy
Bucharest, PO Box 1-764, Romania
gpaun@us.es

Mario de J. Pérez Jiménez
Dept. of CS and Artificial Intelligence
University of Seville, Sevilla,
Avda. Reina Mercedes s/n, 41012, Spain
marper@us.es

Dana Petcu
Computer Science Department
Western University of Timisoara
V.Parvan 4, 300223 Timisoara, Romania
petcu@info.uvt.ro

Radu Popescu-Zeletin
Fraunhofer Institute for Open
Communication Systems
Technical University Berlin, Germany
rpz@cs.tu-berlin.de

Imre J. Rudas
Óbuda University
Budapest, Hungary
rudas@bmf.hu

Yong Shi
School of Management
Chinese Academy of Sciences
Beijing 100190, China &
University of Nebraska at Omaha
Omaha, NE 68182, U.S.A.
yshi@gucas.ac.cn, yshi@unomaha.edu

Athanasios D. Styliadis
University of Kavala
Institute of Technology
65404 Kavala, Greece
styliadis@teikav.edu.gr

Gheorghe Tecuci
Learning Agents Center
George Mason University
U.S.A.
University Drive 4440, Fairfax VA
tecuci@gmu.edu

Horia-Nicolai Teodorescu
Faculty of Electronics and
Telecommunications
Technical University "Gh. Asachi" Iasi
Iasi, Bd. Carol I 11, 700506, Romania
hteodor@etc.tuiasi.ro

Dan Tufiş
Research Institute for Artificial Intelligence
of the Romanian Academy
Bucharest, "13 Septembrie" 13, 050711,
Romania
tufis@racai.ro

Lotfi A. Zadeh
Director,
Berkeley Initiative in Soft Computing (BISC)
Computer Science Division
University of California Berkeley,
Berkeley, CA 94720-1776
U.S.A.
zadeh@eecs.berkeley.edu

DATA FOR SUBSCRIBERS
Supplier: Cercetare Dezvoltare Agora Srl (Research & Development Agora Ltd.)
Fiscal code: 24747462
Headquarter: Oradea, Piata Tineretului Nr.8, Bihor, Romania, Zip code 410526
Bank: BANCA COMERCIALA FEROVIARA S.A. ORADEA
Bank address: P-ta Unirii Nr. 8, Oradea, Bihor, România
IBAN Account for EURO: RO50BFER248000014038EU01
SWIFT CODE (eq.BIC): BFER

Nomination by Elsevier for "Journal Excellence Award"- Scopus Awards 2015
https://www.elsevier.com/solutions/scopus/promo/scopus_awards_romania/award-2

"Research Excellence Award will recognize outstanding scientific journals as seen by modern bibliometric methods. To identify such journals, SNIP - Source Normalized Impact per Paper indicator is used, based on the data from Scopus database.

SNIP (Source Normalized Impact per Paper) measures a source's contextual citation impact by weighting citations based on the total number of citations in a subject field. It helps you make a direct comparison of sources in different subject fields.

SNIP takes into account characteristics of the source's subject field, which is the set of documents citing that source. SNIP especially considers: the frequency at which authors cite other papers in their reference lists; the speed at which citation impact matures; the extent to which the database used in the assessment covers the field's literature.

SNIP is the ratio of a source's average citation count per paper and the citation potential of its subject field.

The citation potential of a source's subject field is the average number of references per document citing that source. It represents the likelihood of being cited for documents in a particular field. A source in a field with a high citation potential tends to have a high impact per paper.

Citation potential is important because it accounts for the fact that typical citation counts vary widely between research disciplines. For example, they tend to be higher in life sciences than in mathematics or social sciences. If papers in one subject field contain an average of 40 cited references while those in another contain an average of 10, then the former field has a citation potential that is 4 times higher than that of the latter.

Citation potential also varies between subject fields within a discipline. For instance, basic journals tend to show higher citation potentials than applied or clinical journals, and journals covering emerging topics tend to have higher citation potentials than periodicals in well established areas."

Interview with Editor-in-Chief Ioan Dzitac

https://www.elsevier.com/solutions/scopus/promo/scopus_awards_romania/award-2/ijccc-interview

Elsevier: How do you feel about being nominated for Scopus Awards 2015?

Editor-in-Chief Ioan Dzitac: The whole team is honored by the nomination which we believe is an acknowledgement of all the hard work that stands behind the publishing of International Journal of Computers Communication & Control (IJCCC). We would like to thank our authors, reviewers and the entire team for all their dedication, originality and hard work.

Elsevier: What gap do you think your journal fills in your respective field of research?

Editor-in-Chief Ioan Dzitac: IJCCC has been focused from the very beginning on promoting research that integrates the "3Cs" - Computing, Communication and Control as to N. Wiener's theory in order to try and differentiate ourselves from the other journals indexed in the same category by Scopus.

Elsevier: If you could pick 5 articles of great importance for your field of research that have been published in your journal which would those be and why?

Editor-in-Chief Ioan Dzitac: I have decided to choose 5 of the most cited articles according to international databases such as Scopus and SCI Expanded:
(1) Spiking neural P systems with anti-spikes, by L. Pan, G. Paun, 2009.
(2) Tissue P systems with cell division, by G. Paun,M.J. Pérez-Jiménez, A. Riscos-Núnez, 2008.
(3) Computing Nash equilibria by means of evolutionary computation, by R.I. Lung, D. Dumitrescu, 2008.
(4) Lorenz system stabilization using fuzzy controllers, by R.E. Precup, M.L. Tomescu, S. Preitl, 2007.
(5) Neuro-fuzzy based approach for inverse kinematics solution of industrial robot manipulators, by S. Alavandar, M.J. Nigam, 2008.

Elsevier: For you, as Editor-in-chief, what is the most important development objective for 2016?

Editor-in-Chief Ioan Dzitac: Reinforcing the intellectual current we have created through publishing high quality articles that bring new ideas and multidisciplinary approaches.

Elsevier: Why did you choose to publish your journal Open Access?

Editor-in-Chief Ioan Dzitac: We have never followed financial gain as our main goal has been to publish high quality articles with a real effect on the evolution of society.

Elsevier: What do you think makes your journal stand out?

Editor-in-Chief Ioan Dzitac: First of all it stands out through the approach of the subject by integrating the 3Cs. On top of this I would add the geographical distribution of the authors that come from over 45 countries and are affiliated to more than 150 universities. The prestige of the editorial board that included researchers from 14 countries.

The editorial team is another strong point as it includes researchers affiliated to high ranked universities from top 100 QS (1. Massachusetts Institute of Technology (MIT), 17. Cornell University, 24. McGill University, 25. Tsinghua University, 26. University of California Berkeley, Berkeley (UCB), 56. Tokyo Institute of Technology, 70. Shanghai Jiao Tong University, 80. University of Sheffield, 89. Purdue University, 96. University of Alberta).

Finally I would like to point out our association with the International Conference on Computers Communications and Control.

6th INTERNATIONAL CONFERENCE on COMPUTERS, COMMUNICATIONS and CONTROL (ICCCC 2016)

Hotel President, Baile Felix, Oradea, Romania, May 10-14, 2016

Organized by Agora University of Oradea,

under the aegis of Romanian Academy: Information Science and Technology Section.

http://univagora.ro/en/icccc2016/

Scope and Topics

The International Conference on Computers Communications and Control (ICCCC) has been founded in 2006 by I. Dzitac, F.G. Filip and M.-J. Manolescu and organized every even year by Agora University of Oradea, under the aegis of the Information Science and Technology Section of Romanian Academy and IEEE - Romania Section.

The goal of this conference is to bring together international researchers, scientists in academia and industry to present and discuss in a friendly environment their latest research findings on a broad array of topics in computer networking and control.

The Program Committee is soliciting paper describing original, previously unpublished, completed research, not currently under review by another conference or journal, addressing state-of-the-art research and development in all areas related to computer networking and control.

In particular the following topics are expected to be addressed by authors:

1) Integrated solutions in computer-based control and communications;
2) Network Optimization and Security;
3) Computational intelligence methods (with particular emphasis on fuzzy logic-based methods, ANN, evolutionary computing, collective/swarm intelligence);
4) Data Mining and Intelligent Knowledge Management;
5) Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies);
6) Membrane Computing - Theory and Applications;
7) Stereovision Based Perception for Autonomous Mobile Systems and Advanced Driving Assistance.

Special Sessions

Special Session 1: Network Optimization and Security, Organizer and Chair: Yezid DONOSO (Colombia);

Special Session 2: Data Mining and Intelligent Knowledge Management, Organizers and Chairs: Gang KOU and Yi PENG (China);

Special Session 3: Computational Intelligence Methods, Organizers and Chairs: Razvan ANDONIE and Donald DAVENDRA (USA);

Special Session 4: Advanced Decision Support Systems, Organizers and Chairs: Marius CIOCA (Romania) and Felisa CORDOVA (Chile);

Special Session 5: Fuzzy Control, Modeling and Optimization, Organizer and Chair: Radu-Emil PRECUP (Romania);

Special Session 6: Membrane Computing - Theory and Aplications, Organizers and Chairs: Marian GHEORGHE (UK) and Florentin IPATE (Romania);

Special Session 7: Stereovision Based Perception for Autonomous Mobile Systems and Advanced Driving Assistance, Organizer and Chair: Sergiu NEDEVSCHI (Romania).

Keynote Speakers: Enrique HERRA VIEDMA(Spain), Zenonas TURSKIS (Lithuania), Gang KOU (China).

Conference Chairs: Ioan DZITAC, Florin Gheorghe FILIP and Misu-Jan MANOLESCU.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, Volume 11, Issue 2, April, 2016.

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):167-178, April 2016.

Improved Performance by Combining Web Pre-Fetching Using Clustering with Web Caching Based on SVM Learning Method

K.R. Baskaran, C. Kalaiarasan

Kuttuva Rajendran Baskaran*
Department of Information Technology
Kumaraguru College of Technology, Coimbatore, India
*Corresponding author: krbaski@yahoo.com

Chellan Kalaiarasan
Tamilnadu College of Engineering, Coimbatore, India
ckalai2001@yahoo.com

Abstract: Combining Web caching and Web pre-fetching results in improving the bandwidth utilization, reducing the load on the origin server and reducing the delay incurred in accessing information. Web pre-fetching is the process of fetching the Web objects from the origin server which has more likelihood of being used in future. The fetched contents are stored in the cache. Web caching is the process of storing the popular objects "closer" to the user so that they can be retrieved faster. In the literature many interesting works have been carried out separately for Web caching and Web pre-fetching. In this work, clustering technique is used for pre-fetching and SVM-LRU technique for Web caching and the performance is measured in terms of Hit Ratio (HR) and Byte Hit Ratio (BHR). With the help of real data, it is demonstrated that the above approach is superior to the method of combining clustering based pre-fetching technique with traditional LRU page replacement method for Web caching.
Keywords: Classification, support, confidence, hit ratio, byte hit ratio, web pre-fetching, web caching.

1 Introduction

In recent times, with rapid growth of WWW, there is ever increasing demand for computer networking resources. With increased number of Web based applications created by many users, the increase in bandwidth does not address the delay problems [5]. The existing prediction algorithms often predict relevant as well as irrelevant pages. In caching the pre-fetched Web pages, efficient cache replacement techniques have to be deployed to manage the cache content. The traditional cache replacement techniques used does not increase the cache hit ratio to a great extent. Machine learning techniques are deployed to improve the performance of Web proxy caching methods [2]. Compared to traditional caching approaches, intelligent Web caching methods are more efficient. Details about intelligent caching methods are found in [1]. The remaining parts of this paper are organized as follows. Section 2 gives an overview of Web caching and Web Pre-fetching techniques. Section 3 contains the proposed block diagram (with description) and the steps involved in the proposed method of combined clustering (intra clustering also considered) based pre-fetching technique with machine learning technique (SVM algorithm). It also contains the algorithm for the combined intelligent Caching (SVM) and Pre-fetching. Section 4 discusses about performance evaluation and section 5 concludes the paper and suggests possible future works.

2 Web caching and Web Pre-fetching

Enhancing the performance of Web based systems is possible by Web caching in which, the Web objects which have high probability of being accessed in the near future are kept closer to the user either in the client's machine or in the proxy server.

The factors (features) of Web objects that influence Web proxy caching considered in this work are, namely: recency (object's last reference time), frequency (number of requests made to an object), size (size of the requested Web object) and access latency of Web object.

The standard metrics used to analyze the performance of web caching methods are Hit Ratio (HR) and Byte Hit Ratio (BHR). HR is the percentage of number of requests that are served by the cache over the total number of requests.

BHR is the percentage of the number of bytes that correspond to the requests served by the cache over the total number of bytes requested. A high HR indicates the presence of the requested object in the cache most of the time and high BHR indicates reduced user-perceived latency and savings in bandwidth.

3 The proposed method of combined clustering based pre-fetching technique with machine learning technique

As shown in the Fig. 1, Web pages accessed by various users are identified from the log file. Web log file contents are preprocessed and trained using the features namely: recency, frequency, retrieval time and size of web object. Dataset is created from the proxy log file. Web navigation graph (WNG) is constructed for each user using a user's session time interval of thirty minutes. It is considered that the two subsequent requests should not have a time interval greater than thirty minutes. At the end of each time interval new navigation graph is constructed for each user based on the log files contents. WNGs show the navigations made by various users between various Web objects for inter-site clustering and between various Web pages with in a Web object for intra-site clustering. Each node in the WNG represents a Web object requested by the user and each edge represents user's transitions from one Web object to another and a weight is assigned to each edge which represents the number of transitions between those nodes. A clustering algorithm gets the contents of WNGs as inputs and two parameters namely Support and Confidence are used to keep track of frequently visited objects/pages by the user. By fixing a threshold value for these parameters, edges which have values less than this threshold can be removed [4]. Support is defined as the frequency of navigation between two nodes u1 and u2. The confidence is defined as freq(u1,u2)/pop(u1) where pop(u1) is the popularity of u1. Popularity of a node is the number of incoming edges in to that node. The WNG is partitioned in to sub graphs by removing those edges that have low Support and Confidence values. The nodes in each connected sub-graph become a cluster. When a user requests any one of the nodes in a cluster and if it is found in the long- term cache or in the short-term cache, all the remaining nodes in that cluster along with all the intra pages of the requested node can be pre-fetched in to the short-term cache if it is not in the short- term cache or in the long-term cache [4].

This pre-fetching is done during the browser idle time and this pre-fetching helps in reducing the user perceived latency time. If a Web object is accessed by the user for an access count greater than the threshold then that Web object are moved from the short term cache to long term cache. LRU method is used for removal of objects from the short term cache if sufficient space is not available for caching a new Web object.

When the objects are moved from short term cache to long term cache SVM classifier is used to classify the Web objects as class 0 or class 1 [2]. When a request is made for a Web object,

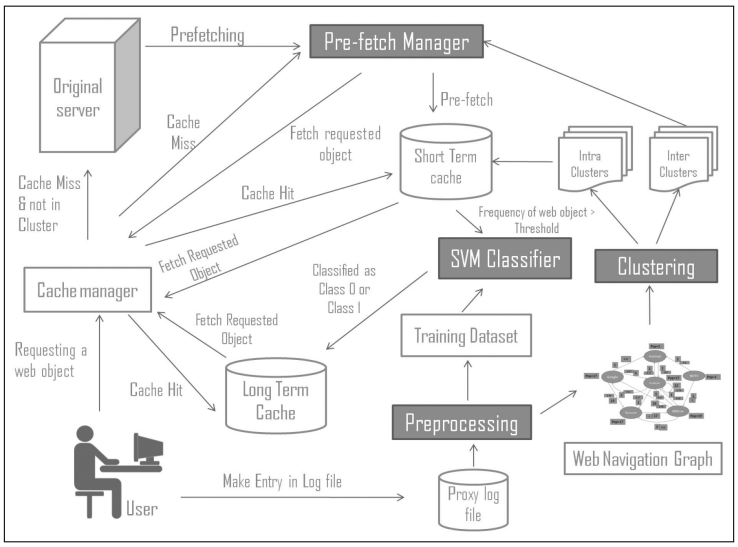


Figure 1: Block diagram

simultaneous search is made in short term cache as well as in long term cache. If that Web object is available in the short-term cache, its access count is increased by one. When the access count becomes greater than the threshold value then that Web object is given as input to SVM classifier for classification. If the classifier classifies it as class 0, it is moved to the bottom of long-term cache else if classified as class 1 then it is moved to the top of long-term cache. If sufficient space is not found in the long-term cache, then the Web object placed at the bottom of the long-term cache are removed to offer space for the new Web object. At the same time a copy of this Web object is transmitted to the requested user. If this object is found in the long-term cache, re-classification of that object is performed.

If it is re-classified as class 1, it is moved to the top of the cache (long-term cache) else it is moved to the bottom of that cache. It is then transmitted to the requested user and pre-fetching of other Web objects if any that belong to that cluster as well as all the intra pages of this Web object in to the short-term cache is initiated. If cache miss occurs then the requested object is searched in all the clusters generated by clustering algorithm for that user. If found in any one of those clusters, that Web object is fetched from the original server and it is transmitted to the user as well as placed in to the short term cache.

Other Web objects of that cluster will be pre-fetched during browser idle time and will be placed in the short-term cache. If the requested object is not found in any of the clusters of that user, then the required Web page is fetched from the original server and a copy of it is placed in the short term cache as well as transmitted to the requested user. The proposed technique of combining Web caching and pre-fetching makes it possible to increase the hit ratio, decrease the user perceived latency and lower the origin server load.

3.1 Preprocessing step and creation of training dataset

A log file generated by the proxy server consists of time stamp, machine IP size of the requested object, type of method used URL of the requested page, content type etc. While pre-processing, JavaScript files, Cascading style sheet, images that are not requested by the user are removed. A separate log file for individual users for creating inter as well as intra-site Web clustering of Web objects is generated. For creating a training dataset, information is extracted from the traces of log file. Each log file record is converted to the training pattern in the format of

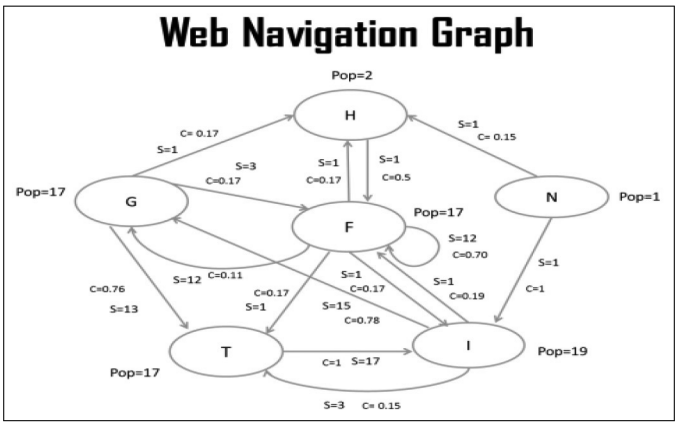


Figure 2: Web Navigation graph for User1

<a1, a2, a3, a4, b> where a1 represents the recency of the Web object, a2 represents frequency of the Web object, a3 represents the size of the Web object, a4 represents the retrieval time (access latency) of the Web object and b represents the class of the Web object. The data type of a1, a2, a3, a4 is numeric and the data type of b is nominal [2].

3.2 Web Navigation Graph (WNG)

A weighted directed Web graph G(x,y) is used to represent the requests of each user, where each node x represents a Web object and each edge y represents a user's transition from one Web object to another. The weight of each edge represents the number of transitions in the set. To make the size of the WNG manageable, the edges are removed whose connectivity between two Web objects is lower than a specified threshold. Support and confidence are the two parameters that determine the connectivity between two objects [4]. Let W:<xi,xj> be an edge from node xi to node xj. Support of G, denoted by freq(xi,xj) is defined as the frequency of navigation steps between xi to xj. Confidence of g is defined as freq(xi,xj)/pop(xi) where pop(xi) is the popularity of xi. By this definition, the support value of the edge (x3,x4) for the user X in Fig. 2 is q(x3,x4) = 1, and confidence value is freq(x3,x4)/pop(x3) = 0.5. If the support threshold chosen is very less, too many less important user's transitions for clustering may be included and if the chosen threshold value is high, many interesting transitions that occur at low levels of support may be missed.

3.3 Web clustering algorithm

The algorithm for clustering inter-site Web pages is described below [4]. A weighted directed Web graph G(x,y) that represents the access patterns of a user is used. This graph is partitioned into sub graphs by filtering edges with low support and confidence values. The nodes in each connected sub graph in the remaining navigational graph will form a cluster. The inputs to this clustering algorithm will be Web navigational graph, the number of users, support threshold and confidence threshold. Remove all the edges with support or confidence value less than the corresponding threshold values. BFS (Breadth First Search) algorithm is applied to the navigational graph. BFS takes a node in the graph (called as source) and visits each node reachable from the source by traversing the edges. It outputs a sub-graph that consists of the nodes reachable from the source. This procedure is applied for all the nodes of the graph. All the nodes in each connected sub-graph forms a cluster. The time complexity of BFS is O(|x|+ |y|) where |x|is the number of nodes and |y|is the number of edges in the graph [4].

google, facebook, google, facebook, google,
hotmail, ndtv, ibnlive, techrunch, Ibnlive,
techrunch, ibnlive, techrunch, ibnlive, google,
facebook, hotmail, facebook, ibnlive, facebook,
google, facebook, google, facebook, google,
hotmail, ndtv, ibnlive, techrunch, Ibnlive,
techrunch, ibnlive, techrunch, ibnlive, google,
facebook, hotmail, facebook, ibnlive, facebook,
google, facebook, google, facebook, google,
hotmail, ndtv ,ibnlive, techrunch, Ibnlive,
techrunch, ibnlive, techrunch, ibnlive, google,
facebook, hotmail, facebook, ibnlive, facebook,
techrunch, ibnlive, google, facebook, hotmail,
facebook, ibnlive, facebook, google, facebook,
google, facebook, google, hotmail, ndtv ,ibnlive,
techrunch, Ibnlive, techrunch, ibnlive,
techrunch, ibnlive, google, facebook, hotmail,
facebook, ibnlive, facebook

Figure 3: User access pattern for User1

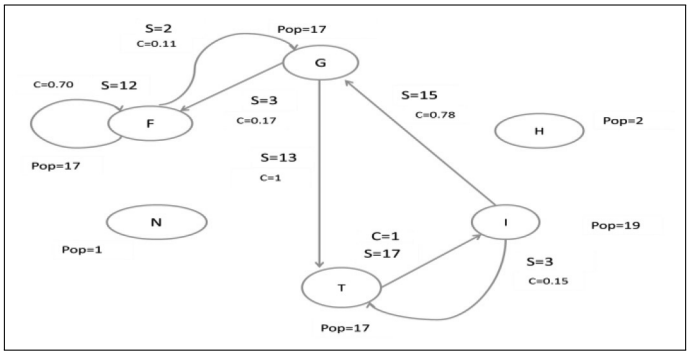


Figure 4: Applying the Support Threshold

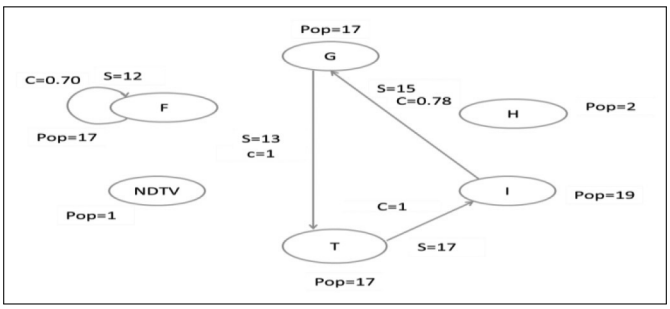


Figure 5: Applying the Confidence threshold

In the above Web Navigation Graph, G stands for Google Web object, H for Hot Mail Web object, N for NDTV Web object, F for Face Book Web object, I for IBN Web object and T for Techrench Web object.

The access pattern for the user1 consists of 6 different Web objects. From the access pattern information, WNG is constructed.

Support and Confidence value for each edge in the WNG is calculated as defined in section 3 and the popularity for each Web object is computed. By assumption Support threshold value is taken as 2 and confidence threshold value as 0.6.

Those edges which have Support and Confidence values less than their threshold are removed. The threshold value chosen for Support and Confidence are critical in specifying the cluster size. It should be noted that the total size of all the objects in a cluster should not exceed the total cache size.

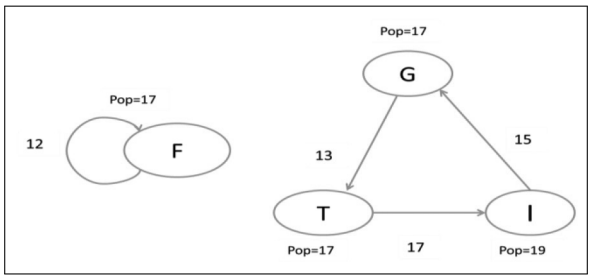


Figure 6: Applying BFS

Then BFS algorithm is applied to the above navigational graph. For each node in the graph known as source, BFS algorithm tries to visit every other node that can be reached from the source by traversing the edges.

It outputs a sub graph that consists of all the nodes reachable from the source. This procedure is iterated until BFS has traversed all the nodes of the initial graph. The nodes in every connected sub graph in the remaining graph forms a Web cluster.

3.4 Pre-fetching using clustering

The following are the steps that take place in the proposed pre-fetching method [4]:

- A user requests a web object. Using the IP address, the proxy identifies the user and maps the user to a particular user group. Given that the clusters of Web objects are known, the proxy searches inside the existing clusters of that user group to find in which cluster the requested object exists.
- All the remaining objects from the selected cluster are pre-fetched from the origin server by the proxy and they are loaded into the short-term cache during the browser idle time.
- The proxy sends to the user his/her requested object.

3.5 Algorithm for the combined intelligent Caching (SVM) and Pre-fetching

Begin
Apply-Clustering algorithm
For each web object m requested by the user
If m is in short-term cache
Begin
Cache hit occurs
Fetch the requested object m from short-term cache
Update the information of m
If the frequency of $m>$ threshold limit
Begin
While no space in the long-term cache for m
Begin
Expel f from long-term cache such that f is in bottom of the long-term cache
End
Class of m=**apply-svm(Common features)**
If class of m=1

```
  Move m to top of the long-term cache
Else
  Move m to the long-term cache
End
End
Else if m is in long-term cache
Begin
  Cache hit occurs
  Fetch the requested object m from the long-term cache
  Update the information of m
  Recalculate the class of m
  If class of m =1
    Move m to the top of the long-term cache
  Else
    Move m to the long –term cache
  End
Else
Begin
  Cache miss occurs
  Fetch m from original server
  Cluster c= getClusterForUser(m)
  If c is not NULL
  Begin
    While no space in short-term cache for web objects in c
    Begin
    Expel object using LRU from short-term cache
    End
  Load the all cluster into the short-term cache during the browser idle time
  End
End
Procedure getClusterForUser(m)
Begin
  For each cluster p for the user
    If m is in cluster p
      p=p+getTheIntrasiteclusters(p);
      return p
  If m is not in any of the cluster
    return NULL
End
End
```

Clustering Algorithm

```
Begin
For each client IP address
Begin
Construct web navigation graph G (U, V)
End
For each G (U, V)
Begin
```

Calculate *Support, Confidence* and *Popularity* of the node U
If *Confidence*< Threshold limit of Confidence
Begin
Remove *V*
End
If *Support*< threshold limit of Support
Begin
Remove *V*
End
Cluster *C* = **Apply BFS** for G (U, V)
End

Breadth first search

Begin
Input: Graph G (U, V)
Choose some starting node *u1*
Mark *u1* as visited
Initialize list *x1* with *u1*
Initialize sub-graph *T* with *u1*
While *x1* is not empty
Begin
Choose node *x2* from front of the list
Visit *x2*
End
For each unmarked neighbor *y*
Begin
Mark *y*
Add *y* to the end of the list *x1*
Add *x2->y* to subgraph *T*
End
Find all neighbors of the node *u1*
Visit each neighbor and mark the visited node
End

Intra clustering Algorithm

Procedure *getTheIntrasiteclusters(m)*
Begin
T=Total number of sessions
For each client IP address
Begin
For each session S
Begin
U=Unique set of Intra-site pages in session S
C=Total number of intra-site pages in session S
For each intra-site page *P* in *U*
Begin
Calculate *Support* and *probability(C/T)* for *P*
If *Support*< threshold limit of Support
Begin
Ignore *P*

Continue;

End

If *Probability*< threshold limit of Probability

Begin

Ignore P

Continue;

End

End

Include P into Cluster C

End End

return C

4 Performance Evaluation

4.1 Dataset

The scheme explained in this paper is tested with a dataset. The dataset is obtained from a proxy server installation ftp://ftp.ircache.net/Traces/DITL- 2007-01-09/. The filename of the dataset used for testing of the scheme is rtp.sanitized-access.20070109.gz under the website. The raw proxy server log file for the dataset contained the details of more than 3 million requests. After the log cleaning process was applied on the dataset, the dataset contained about 610,634 entries fit for analysis. The inner details about the data set are as explained below: (i) Total Number of Items: 610634 (ii) Total Size of Bytes for Dataset: ∼49 GB (iii) Total Number of Items used for *Clustering*: 427443 (iv) Total Size of Bytes Used for *Clustering*: ∼36 GB (v) Total Number of Items used for *Testing*: 183191 (vi) Total Size of Bytes Used for *Testing*: ∼13 GB.

70% of all the requests ordered by time have been used for the user's access pattern analysis, creating training dataset and testing [4].The remaining 30% of the requests were used for testing the scheme.

4.2 Experimental results

Hit Ratio and Byte Hit Ratio Analysis

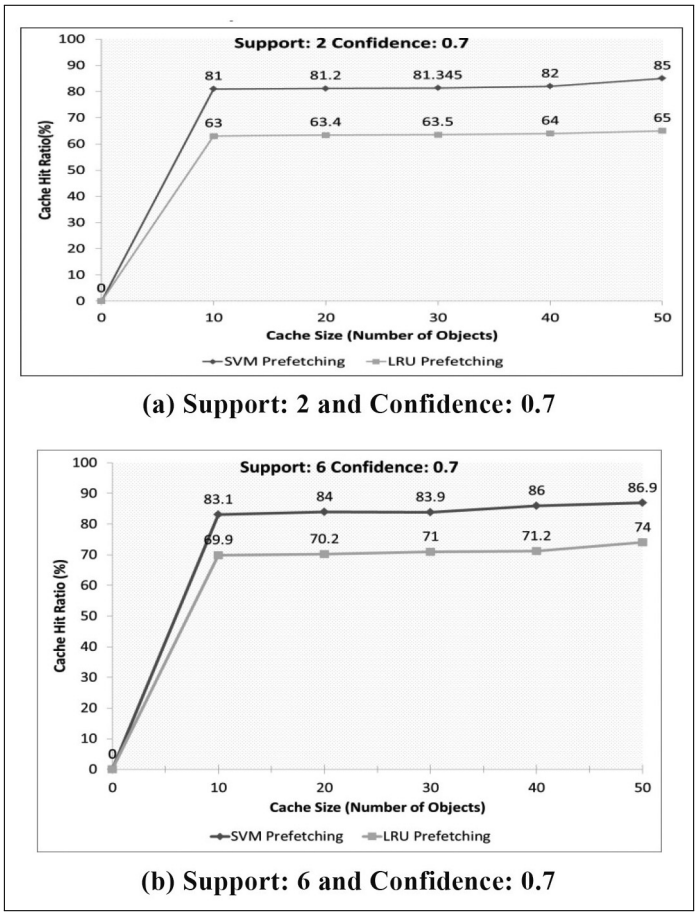


Figure 7: Analysis of HR using SVM and LRU pre-fetching on different values of Support and Confidence

HR and BHR are calculated for different values of Support and Confidence using SVM pre-fetching and LRU pre-fetching. A sample of them is shown above. In the graph, SVM pre-fetching means SVM-LRU caching with pre-fetching and LRU pre-fetching means LRU caching with pre-fetching. Results inferred from the above graphs are stated below:

1. If the Support and Confidence values are increased, it is found that there will be a marginal increase in Hit ratio for increasing cache sizes.

2. Considering Byte Hit Ratio (BHR) it is found that on an average 68% of the total size of the information requested is found fetched from Cache (cache hit) by using SVM-LRU caching with pre-fetching and only 33% of the total size of the requested information is found fetched from the cache using LRU caching with pre-fetching and remaining information are found fetched from the origin server. Considering HR, it is 86% and 67% on average for SVM-LRU and LRU caching with pre-fetching respectively. This shows the superiority of SVM-LRU technique.

   In our earlier paper [3] where intra pages of the requested paper was not considered and LFU technique was used for removal of Web objects from the short-term cache, 64% of

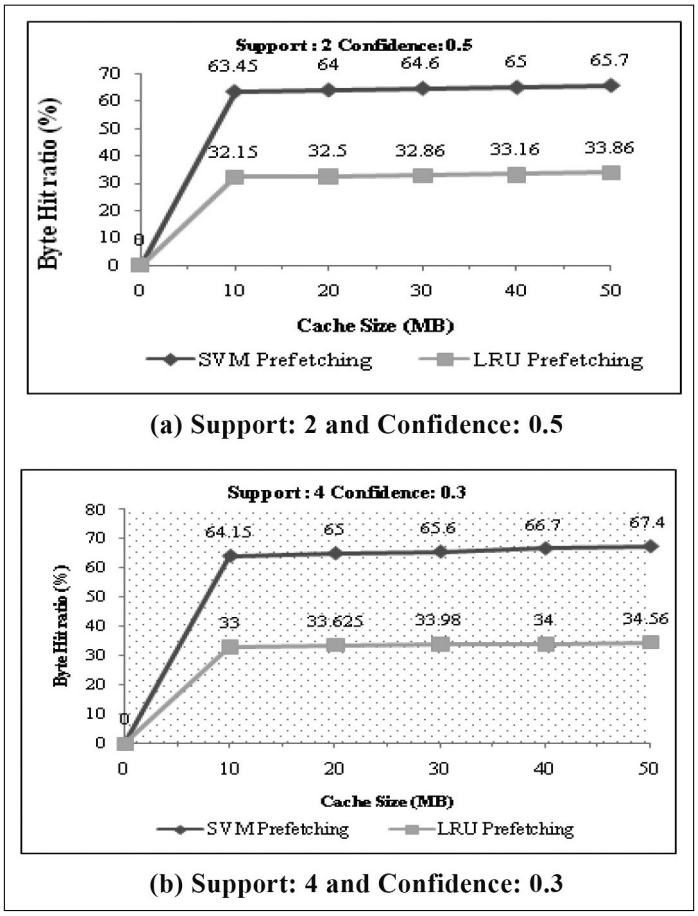


Figure 8: Analysis of BHR using SVM and LRU pre-fetching using different values of Support and Confidence

the total size of the requested information was found fetched from cache due to cache hit (BHR) and 26% of the total size of the information requested was found fetched from cache using LFU pre-fetching and remaining information are found fetched from the origin server. Considering HR, it is 84% and 47% on average for SVM-LFU and LFU caching with pre-fetching respectively.

This shows the superiority of SVM pre-fetching method. By considering the pre-fetching of intra pages of the requested object, the Byte hit ratio BHR is increased by 4% and HR by 2% compared to the Byte hit ratio BHR and HR without considering intra clustering.

3. It is demonstrated that pre-fetching will lead to decrease in network bandwidth utilization and decrease in the access latency because of more cache hits.

5 Conclusion and future work

In this work, a clustering algorithm is used to cluster the Web objects represented in the Web navigation graph. Frequently accessed Web objects are monitored by the Confidence and Support values. If the user's requested Web object is present in the short-term cache then all the other Web objects in that cluster plus all the intra pages of that object are pre-fetched and cached during the browser idle time. If a Web object and its associated pages in the short-term

cache are accessed more number of times than a fixed threshold value then they are moved to long-term cache after classifying them using SVM algorithm. If the requested Web object is found in the long-term cache then all the other Web objects in that cluster are pre-fetched during the browser idle time. If cache miss occurs in both the caches, then that Web object is fetched from the origin server and a copy of it is placed in to the short-term cache. The efficiency of SVM pre-fetching is compared with that of LRU pre-fetching using real data set and it is found that SVM pre-fetching has high HR and high BHR for various values of Support, Confidence and cache sizes. Extension of this work is possible by comparing the efficiency of SVM technique with other machine learning techniques.

Bibliography

[1] Ali W., Shamsuddin S.M., Ismail A.S. (2011), A survey of Web caching and prefetching, *International Journal of Advances in Soft Computing and Its Applications*, 3 (1): 1-27.

[2] Ali W., Shamsuddin S.M., Ismail A.S. (2012), Intelligent Web proxy caching approaches based on machine learning techniques, *Decision Support Systems*, 53(3): 565-579.

[3] Baskaran K.R., Kalaiarasan C., Sasi Nachimuthu A. (2013), Study of combined Web prefetching with Web caching based on machine learning technique, *Journal of Theoretical and Applied Information Technology*, 20th September 2013, 55(2): 280-291.

[4] Pallis G., Vakali A., Pokorny J. (2008), A clustering-based prefetching scheme on a Web cache environment, *Computers and Electrical Engineering*, 34(4): 309-323.

[5] Podlipnig S., Boszormenyi L. (2003); A survey of Web cache replacement strategies, *ACM Computer Surveys*; 35(4):374–98.

[6] Web reference: http://www.wikipedia.com/svm

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):179-193, April 2016.

Design of a Fuzzy Networked Control Systems. Priority Exchange Scheduling Algorithm

H. Benítez-Pérez, J. Ortega-Arjona, J.A. Rojas-Vargas, A. Durán-Chavesti

H. Benítez-Pérez
Universidad Nacional Autónoma de México
Apdo. Postal 20-726, Admón. 20, Del. A. Obregón, México D. F., CP. 01000.
hector.benitez@iimas.unam.mx

Jorge Ortega-Arjona
Facultad de Ciencias UNAM
Av. Universidad 3000, C. U., México D. F.
jloa@ciencias.unam.mx

Jared A. Rojas-Vargas
IIMAS UNAM
Cto. Escolar 3000, C. U., México D. F.
jared_36_23@hotmail.com

A. Durán-Chavesti*
Universidad Nacional Autónoma de México
Apdo. Postal 20-726, Admón. 20, Del. A. Obregón, México D. F., CP. 01000.
*Corresponding author: adrian.chavesti@iimas.unam.mx

Abstract: This work presents a supervisory control strategy for Networked Control Systems (NCSs). This shows the identification and control of the plant using fuzzy theory. The fuzzy model incorporates the delay dynamics within the fuzzy rules based upon a real-time hierarchical scheduling strategy. A hierarchical scheduling Priority Exchange algorithm is used based upon codesign strategy following mutual correlation among control and network algorithms in order to bounded time delays. A system of magnetic levitation is presented as a case study.
Keywords: Fuzzy control, networked control system, time delay codesign.

1 Introduction

The control design and stability analysis of network-based control systems (NCSs) have been studied in recent years [14], [8] and [24] based upon codesign strategy. The main advantages of this kind of systems are their low cost, small volume of wiring, distributed processing, simple installation, maintenance and reliability.
In a NCS, one of the key issue is the effect of network-induced delay in the system performance. The delay can be constant, time-varying, or even random, this depends on the scheduler, network type, architecture, operating systems, etc [24]. One strategy to be followed is the codesign since it takes both desired procedures to be followed. Nilsson analyzes several important facets of NCSs [15]. It introduces models for the delays in NCS, first as a fixed delay, after as an independently random, and finally like a Markov process. The author introduces optimal stochastic control theorems for NCSs based upon the independently random and Markovian delay models. In [18], introduces static and dynamic scheduling policies for transmission of sensor data in a continuous-time LTI system. They introduce the notion of the maximum allowable transfer interval (MATI), which is the longest time after a sensor should transmit a data. [18] derived bounds of the MATI such that the NCS is stable. This MATI ensures that the Lyapunov function of the system under consideration is strictly decreasing at all times. In [22] extends the

work of Walsh., he developed a theorem which ensures the decrease of a Lyapunov function for a discrete-time LTI system at each sampling instant, using two different bounds. These results are less conservative than those of Walsh, because he doesn't require the system's Lyapunov function to be strictly decreasing at all time.
Besides, following the work presented by [13] although the strategy is similar as well as the case study in here, the proposed fuzzy control follows each local time delay produced by the scheduling algorithm which is dynamic and reactive to external tasks modification (as Priority Exchange Proposes). Although the results are stable in both cases, in here the challenging strategy is to dismiss dynamic local time delays without forcing system bounds. It is important to mention that this work follows the expressions designed in [3], [4] and [3] with the characteristic of real local time delays and local gain control design following eqn. 10 and LMI procedure as presented in section 4. In [7], [17], [20] and [21] introduce a number of different linear matrix inequality (LMI) tools for analyzing and designing optimal switched NCSs. [23] takes into consideration both the network-induced delay and the time delay the plant, a controller design method is proposed by using the delay-dependent approach. An appropriate Lyapunov functional candidate is utilized to obtain a memoryless feedback controller, this is derived by solving a set of Linear Matrix Inequalities (LMIs). In [19] models the network induced delays of the NCSs as interval variables governed by a Markov chain. Using the upper and lower bounds of the delays, a discrete-time Markovian jump system with norm-bounded uncertainties is presented to model the NCSs. Based on this model, the H∞ state feedback controller can be constructed via a set of LMIs. Recently [9] introduced a new (descriptor) model transformation for delay-dependent stability for systems with time-varying delays in terms of LMIs, and she also refines recent results on delay-dependent H∞ control and extend them to the case of time-varying delays.
Alternatively [10] takes into consideration both the network-induced delay and the time delay in the plant, and thus, introduces a controller design method, using the delay-dependent approach. An appropriate Lyapunov functional candidate is used to obtain a memoryless feedback controller, derived by solving a set of Linear Matrix Inequalities (LMIs) [6]. [11] models the network induced delays of the NCSs as interval variables governed by a Markov chain. Using the upper and lower bounds of the delays, a discrete-time Markovian jump system with norm-bounded uncertainties is presented to model the NCSs. Based on this model, a H∞ state feedback controller can be constructed via a set of LMIs.
An interesting approximation has been presented by [2] where time delays incorporation has been proposed following state space representation.

2 Systems Proposal

Based on this review, this paper defines a model (Fig. 1) that integrates the time delays for a class of nonlinear system, where the actual proposal it is the enhancement of states in order to represent of control and plant states to fulfill a complete modeling of time delays according to priority exchange Dynamic Schedulling Algorithm.

It comprises two types of fuzzy rules, one that models the dynamics of the plant and another that introduces the networked-induced time delay. It involves estimating the time delay based scheduling behaviour where the fuzzy rules are such as:

$$if\ x_i(k)\ is\ \mu_{ij}\ then\ x_j^N(k+1) = A_j x(k) + B_0 u(k) \tag{1}$$

$$i{=}1{\ldots}n\ j{=}1{\ldots}r\ h{=}1{\ldots}s$$

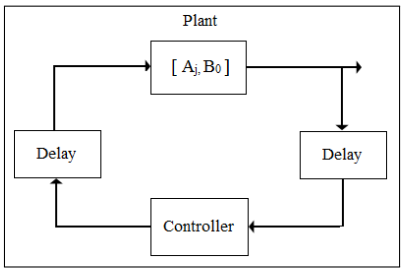


Figure 1: Fuzzy model proposed.

The overall system is:

$$\widehat{x}(k+1) = \sum_{j=1}^{r} x_j^N + \sum_{h=1}^{s} x_h^D = \sum_{j=1}^{r} R_j A_j x(k) + \sum_{h=1}^{s} S_h(B_0)u(k) \tag{2}$$

where x_i is the *ith* state of the plant, μ_{ij} is the membership function of the *ith* state and s is the total number of local time delays and the *jth* rule. $A_j \in R^{nxn}$, $B_{0,h} \in R^n$, $x \in R^n$, $u(k) \in R$, with n states, and r nominal fuzzy rules, sh is he nominal selection of current fuzzy rule. Where N and D denote nominal and delayed model respectively. The fire strength ψ_j is defined as the function multiplication between the membership functions μ_{ij}.

$$\mu_{ij} = exp\left(-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right) \tag{3}$$

$$\psi_j = \prod_{i=1}^{m} \mu_{ij} \tag{4}$$

$$R_j = \frac{\psi_j}{\sum_{k=1}^{r} \psi_k} \tag{5}$$

$$0 < R_j \leq 1, \sum_{j=1}^{r} R_j(x) = 1 \tag{6}$$

For the s fuzzy rules with delay τ_{cah}, ν_h is the gaussian membership function of the time delay with center α_h and standard deviation β_h.

$$S_h = \frac{\nu_h}{\sum_{k=1}^{s} \nu_k} \tag{7}$$

$$\nu_h = exp\left(-\frac{(\tau_{cah} - \alpha_h)^2}{\beta_h^2}\right) \tag{8}$$

The proposed decomposition in terms of feedback state space representation has been reviewed by [4], where the indexing is defined by the time delays as local and bounded situations through the network.

Firstly, as augmented states and the related bounded time delays of plant and controller, following the strategies presented in [3], [4] and [3] the control structure is modified according to a particular gain control per local time delays scenarios and different local operational points from a particular case study. In here the strategy is modified by designing local control laws as

gain rather them a dynamic state feedback control.
The results as shown in section 5 are quite promising in that respect, it is presented as such:

$$X = \begin{bmatrix} x_c \\ x_p \end{bmatrix} \tag{9}$$

$$\begin{matrix} x_c(k+1) \\ x_p(k+1) \end{matrix} = \begin{matrix} \sum_{j=1}^{N}\sum_{i=1}^{N}\left[h_j h_i\left[B_j^p\left(x_c(k-t_{cai})\right)\right] + h_j A_j^p x_p(k)\right] \\ \sum_{j=1}^{N}\sum_{i=1}^{N}\left[h_j h_i\left[F_j^c\left(c_p^i x_p(k-t_{sci})\right)\right] + h_j F_j^c\right] \end{matrix} \tag{10}$$

where the delays are independent based upon the time obtained from scheduling approximation:

$$t_{ca1} + t_{sc1} < t_{ca2} + t_{sc2} < ... < t_{cam} + t_{scm} < T \tag{11}$$

Now from the derivative of a candidate Lyapunov function is expressed as:

$$\Delta u(k) = V(k+1) - V(k) \tag{12}$$

and the related Lyapunov function is:

$$V(k) = X(k)^T P X(k) \tag{13}$$

each of the fuzzy rules is given as an expression of local delays from current condition from plant towards controller, and vice versa.

$$\begin{bmatrix} x_c \\ x_p \end{bmatrix} = \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix} \tag{14}$$

For each rule, there is a delay related to a particular condition to the plant and controller. Each of the rules maybe updated through learning procedure or LMI process. Each of the rules is unique on every specific time. In this case, these are associated to a particular relationship of last equation. In terms of the Lyapunov Candidate, this is expressed as in eqn 15 which is consistent to eqn. 8.

$$V(k+1) - V(k) = \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} - \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix} \tag{15}$$

$$V(k+1) - V(k) = \begin{bmatrix} \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(B_j^p\left(x_c(k-t_{caj})\right)\right) + h_j A_j^p x_p(k)\right) \\ \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(F_j^c\left(x_p(k-t_{scj})\right)\right)\right) \end{bmatrix}^T P$$

$$\begin{bmatrix} \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(B_j^p\left(x_c(k-t_{caj})\right)\right)+h_j A_j^p x_p(k)\right) \\ \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(F_j^c\left(c_p^i x_p(k-t_{scj})\right)\right)\right) \end{bmatrix}^T - \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_p(k) \end{bmatrix} \tag{16}$$

Therefore:

$$V(k+1)-V(k) = \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} -$$

$$\begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix} \tag{17}$$

t_{caj} and t_{scj} are the related time delays. Considering the fuzzy system representation:

$$V(k+1)-V(k) = \begin{bmatrix} \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(B_j^p\left(c_c^i x_c(k-t_{caj})\right)\right)+h_i A_i^p x_p(k)\right) \\ \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(F_j^c\left(c_p^i x_p(k-t_{scj})\right)\right)\right) \end{bmatrix}^T P \tag{18}$$

$$\begin{bmatrix} \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(B_j^p\left(c_c^i x_c(k-t_{caj})\right)\right)+h_i A_i^p x_p(k)\right) \\ \sum_{j=2}^{m}\sum_{i=2}^{m}\left(h_j h_i\left(F_j^c\left(c_p^i x_p(k-t_{scj})\right)\right)\right) \end{bmatrix} -$$

$$\begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix}^T P \begin{bmatrix} x_c(k) \\ x_c(k-t_{ca1}) \\ x_c(k-t_{ca2}) \\ \vdots \\ x_c(k-t_{cam}) \\ x_p(k) \\ x_p(k-t_{sc1}) \\ x_p(k-t_{sc2}) \\ x_p(k-t_{sc3}) \\ \vdots \\ x_p(k-t_{scm}) \end{bmatrix} \tag{19}$$

If only one of the time delays is considered:

$$0 > \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix}^T P \begin{bmatrix} x_c(k+1) \\ x_p(k+1) \end{bmatrix} - \begin{bmatrix} x_c(k) \\ x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \end{bmatrix} P \begin{bmatrix} x_c(k) \\ x_c(k-t_{caj}) \\ x_p(k) \\ x_p(k-t_{scj}) \end{bmatrix} \tag{20}$$

In here every time delay is local, independent and bounded according to dynamic scheduling algorithm which is based upon the structural codesign section.

3 Structural Codesign

The codesign proposal follows the iteration between schedulability and stability analysis following online approximation.

In fact, according to dynamic scheduling algorithm proposal which is based upon structural codesign strategy, these time delays can be seen like a phase modification within the communication period from the involved processes. This scenario presents a complete phase modification at the entire system. The communication network plays a key role in order to define the behavior of the dynamic system in terms of time variance giving a nonlinear behavior. In order to understand such a nonlinear behavior, time delays are incorporated by the use of real-time system theory that allows time delays to be bounded even in the case of causal modifications due to external effects, based upon Priority Exchange [4].

This algorithm bounds Time delays through a real-time scheduling algorithm within communication network. According to Fig. 3, structural reconfiguration takes place as a result of Priority Exchange Scheduling algorithm and the associated user request. This reconfiguration causes a control law modification [19] which is the actual control law reconfiguration.

Scheduling approach potentially modifies frequency execution and communication of tasks in order to give certain priority to some of them during a bounded time as shown in Fig. 3. Furthermore, in this kind of strategy Tasks modifies their priority, it does not imply that neither the period nor the consumption times are modified. Therefore the tasks would have a bounded delay within the sampling time which is reflected as changing on the phase.

Potential modifications onto scheduling approach deploy change in the priorities that affects time delays and the respective control law. The delays are measured as $\Delta\ t$ and bounded into the inherent control period of time according to eqn. 11. Now by taking partial results from scheduling algorithm like t_{sj} and the related Δt, the actual time delays are used at the control law for parameters design. The involved time delays are depicted as τ_j^i and come from this scheduling design. Other delays like actuators and control delays are not used in the design of the control law, although play an important role. Therefore scheduling and control analysis merge together when time delays are complete bounded even in the case of time variance. The main restriction is in terms of predictable time delays.

The objective here is to present a reconfiguration control strategy developed from the time delay knowledge, following scheduling approximation where time delays are known and bounded according to used scheduling algorithm. The scheduling strategy proposed here pursues to tackle local faults in terms of fault tolerance. In this situation, current time delays would be inevitable. Classical Earliest Deadline First (EDF) plus Priority Exchange (PE) [4] algorithm are used here to decompose time lines and the respective time delays when present. For instance, time delays are supervised for a number of tasks as follows:

$$C1 \rightarrow CnT1 \rightarrow Tn \tag{21}$$

Priority is given as the well-known EDF algorithm, which establishes that the process with the closest deadline has the most important priority [12]. However, when an aperiodic task appears, it is necessary to deploy other algorithms to cope with concurrent conditions. To do so, the PE algorithm is used to manage spare time from the EDF algorithm. The PE algorithm [6] uses a virtual server that deploys a periodic task with the highest priority in order to provide enough computing resources for aperiodic tasks. This simple procedure gives a proximity, deterministic, and dynamic behavior within the group of included processes. In this case, time delays can be deterministic and bounded. As an example, consider a group of tasks as shown in Table 1. In this case, consumption times as well as periods are given in terms of integer units. Remember: the server task is the time given for an aperiodic task to take place on the system.

Name	**Consumption (in units)**	**Period (in units)**
Task 1	2	9
Task 2	1	9
Task 3	2	10
Server	1	6

Table 1: First example for PE algorithm.

The result of the ordering based upon PE is presented in Fig. 2.

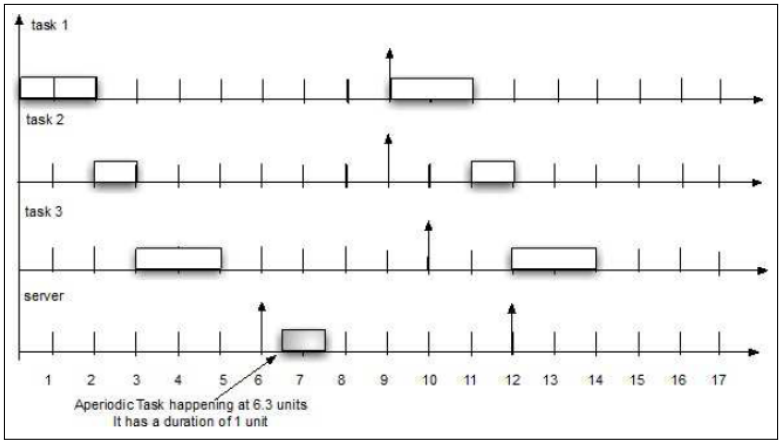


Figure 2: Related organization for PE of tasks in Table 1.

Based on this dynamic scheduling algorithm, time delays are given as current calculations in terms of task ordering. In this case, every time that the scheduling algorithm takes place, the global time delays are modified in the short and long term. For instance, consider the following example, in which four tasks are set, and two aperiodic tasks take place at different times, giving different events with different time delays.

The following task ordering is shown in Fig. 3, using the PE algorithm, where clearly time delays appear.

Now, from this, a resulting ordering of different tiny time delays is given for two scenarios, as shown in Fig. 4.

These two scenarios present two different local time delays that need to be taken into account before hand, in order to settle the related delays according to scheduling approach and control design. These time delays can be expressed in terms of local relations between both dynamical systems. These relations are the actual and possible delays, bounded as marked limits of possible and current scenarios. Then, delays may be expressed as local summations with a high degree

Name	Consumption (in units)	Period (in units)
Task 1	2	9
Task 2	1	9
Task 3	2	10
Server	1	6
Aperiodic task 1 (ap1)	0.9	It occurs at 9
Aperiodic task 2 (ap2)	1.0	It occurs at 13

Table 2: Second example of PE.

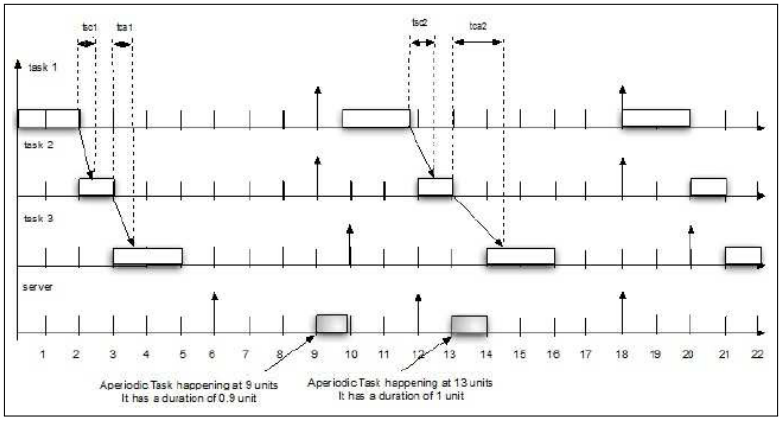


Figure 3: Related time delays are depicted according to both scenarios.

of certainty (as presented in [13]). In this last example, during the second scenario, a total delay is given as:

$$\text{Total delay} = \text{consumption\_time\_delay\_aperiodic\_task1} + \text{consumption\_time\_delay\_task1} + \text{tsc2} + \text{consumption\_time\_delay\_task2} + \text{consumption\_time\_delay\_aperiodic\_task2} + \text{consumption\_time\_delay\_task3}$$

Now, from this example, l_p is equal to 2 and l_c is equal to 3. l_p and l_c are the total number of local delays within one scenario from sensor to control and from control to actuator respectively.

In this case, local time delays as presented in the general eqn. 14 are the result of the iteration of scheduling algorithm. In the approximation presented in this paper the local delays are around four time delays as expressed as last expression called total delays.

The approach followed at the control reconfiguration does not take into account scheduler decision in a direct manner. It takes the time delays as bounded values already defined and used to design a suitable control law. Therefore, according to current state plant values, the related fuzzy rule is selected.
For a NCS, the communication network strongly affects the dynamics of the system, expressed as a time variance that exposes a nonlinear behaviour. Such nonlinearity is addressed by incorporating time delays. From real-time system theory, it is known that time delays are bounded even in the case of causal modifications due to external effects.

4 Case of Study

The case of study consists of a simulation from magnetic levitation system whose sensors and actuators are operated by a "host", the signals from the sensors are sent by the host through a ETHERNET 10/100 network and received by a "server" where the control input is calculated and sent over ETHERNET network to the host. Fig. 5 shows current system configuration as in real state.

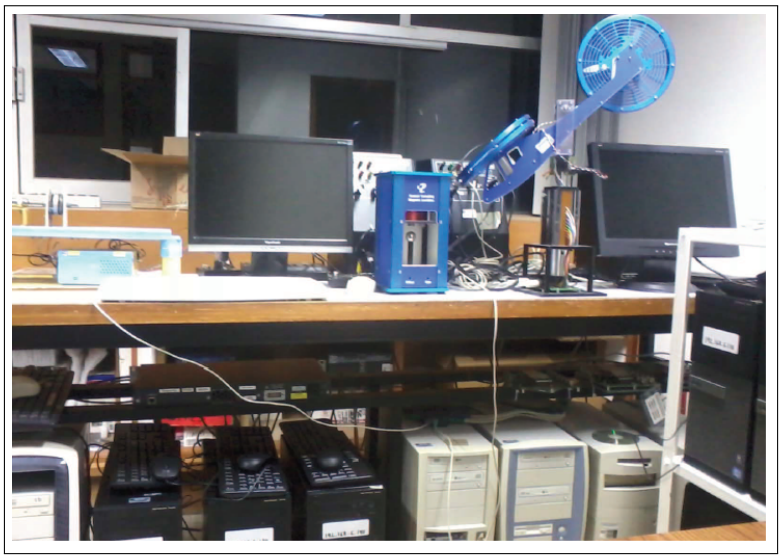


Figure 4: Current configuration of magnetic levitation system.

The system consists of a coil inside a cabin, the coil levitates a steel ball that rests on a black post. The elevation of the ball is measured from the post using a light sensor inside the post. The issue of the experiment is to design a controller that does levitate the steel ball following a desired trajectory.

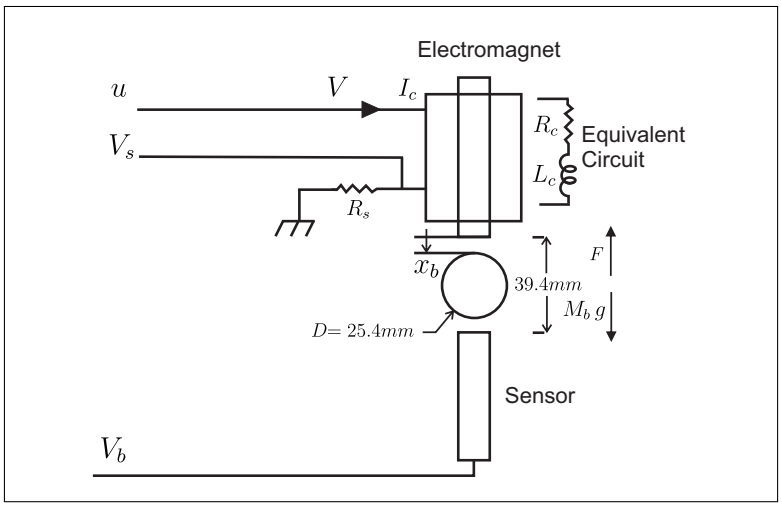


Figure 5: Maglev system.

The nonlinear equations for the Magnetic Levitation System are:

$$\dot{x_1} = x_2$$
$$\dot{x_2} = \frac{-K_m x_3^2}{2M_b(x_1)^2} + g$$
$$\dot{x_3} = \frac{1}{L_c}(-Rx_3 + u)$$

were $R = R_c + R_s$ and $u = V_c$ input voltage and
R_c electromagnet resistance
R_s resistor in series wiht the coil
K_m constant of electromagnet force
M_b mass of the ball
g gravitacional constant
L_c coil inductance

The values of the parameters are provided in [16].

The method generated three rules for the nominal fuzzy control and the range of delay was divided in six parts then the delayed fuzzy control has six fuzzy rules. For the fuzzy model three feedback vector F_j were designed to ensures the stability of the overall system.

Following eqn. 10 and resolving eqn. 20 through LMI it is possible to verify the stability in an asymptotic procedure.

5 Results

Once the fuzzy control laws are designed according to equations (17)-(20) where the objective is to find a common positive definite matrix P satisfying the linear matrix inequality. Two tests are performed to prove the effectiveness of the method proposed. In all tests the reference trajectory signal applied is a sine signal to be followed by the steel ball.

Three fuzzy rules are defined to approximate the magnetic levitation system by means of three linear models, as follows:

Rule 1:
IF $x_1(t)$ is about $0.006\,m$,
THEN $x(k+1) = A_1x(k) + B_1u(k)$

Rule 2:
IF $x_1(t)$ is about $0.009\,m$,
THEN $x(k+1) = A_2x(k) + B_2u(k)$

Rule 3:
IF $x_1(t)$ is about $0.013\,m$,
THEN $x(k+1) = A_3x(k) + B_3u(k)$
where x_1 is the ball position in meters and

$$A_1 = \begin{bmatrix} 1.0016 & 0.0010 & 0 \\ 3.2718 & 1.0016 & -0.055 \\ 0 & 0 & 0.9737 \end{bmatrix}$$
$$A_2 = \begin{bmatrix} 1.0011 & 0.0010 & 0 \\ 2.1808 & 1.0011 & -0.0175 \\ 0 & 0 & 0.9737 \end{bmatrix}$$
$$A_3 = \begin{bmatrix} 1.0012 & 0.0010 & 0 \\ 2.3774 & 1.0012 & -0.0212 \\ 0 & 0 & 0.9737 \end{bmatrix}$$
$$B_1 = B_2 = B_3 = \begin{bmatrix} 0 \\ 0 \\ 0.0024 \end{bmatrix}$$

The control gains obtained by means LMI Matlab's toolbox are:

$$\begin{aligned} F_1 &= \begin{bmatrix} -51650 & -1102 & 379 \end{bmatrix} \\ F_2 &= \begin{bmatrix} -48530 & -1058 & 341 \end{bmatrix} \\ F_3 &= \begin{bmatrix} 22546 & -479 & 128 \end{bmatrix} \end{aligned}$$

These control gain values guarantee the stability of the system during the presence of local time delays according to table 3. In this case local time delays are responsive in terms on an periodic external task, that is presented every determined seconds.

With the next positive definite matrix P:

$$P = \begin{bmatrix} 0.1980 & 0.0042 & -0.0007 \\ 0.0042 & 0.0001 & -0.0000 \\ -0.0007 & -0.0000 & 0.0000 \end{bmatrix} \tag{22}$$

In order to prove the effectiveness of the metod proposed, two experiments were performed, in the first scenario the plant tracks a reference signal (sine wave) and the transmission task were the following (Table 3)

Name	**Consumption (in milliseconds)**	**Period (in milliseconds)**
Task 1	2	10
Task 2	1	12
Task 3	2	14
Aperiodic Task	1	90

Table 3: PE

The activation task was performed using Stateflow as shown in Fig. 6 where according to Table 3 the *task* 1 is the controller transmission task and has the priority one, the *task* 2 is the sensor transmission task an has the priority two and the *task* 3 *and* 4 are the transmissions task

and the sporadic transmission task from others nodes.

The system response obtained in this first experiment is shown in Fig. 7 (without time delays).

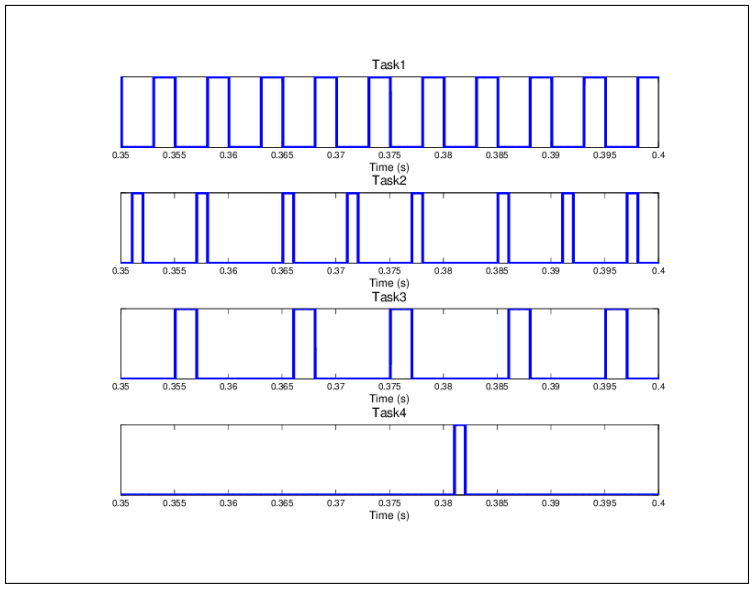


Figure 6: Activation tasks following table 3

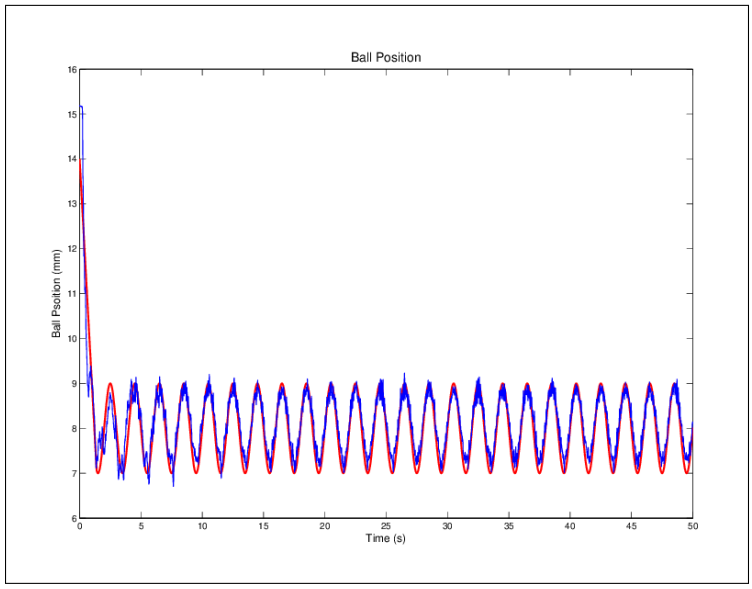


Figure 7: Ball Position Response in the first scenario

The second test is to apply a time delay less than the sampling period according to the total time delay. Fig. 8 shows the behavior of the system which maintains stability with a delay of 2 ms and a sampling period of 90 ms related to the aperiodic task. The behavior is very similar to the system without time delay.

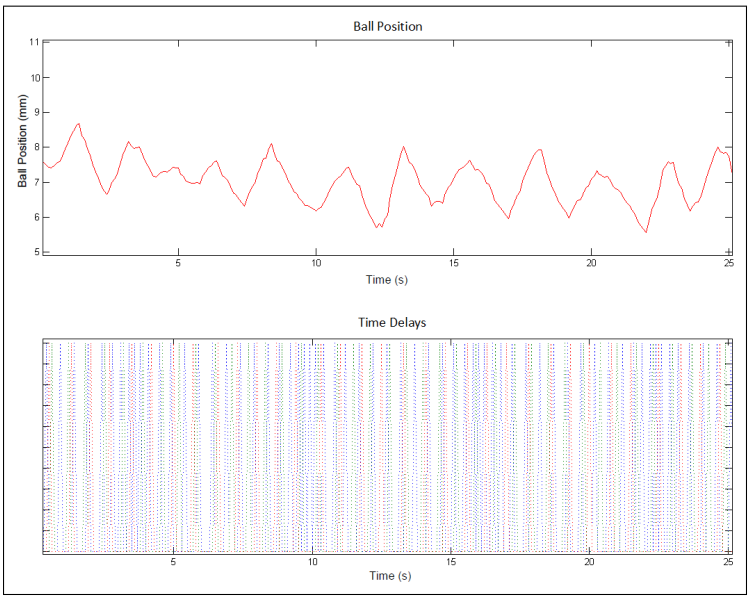


Figure 8: Ball Position Response in the second scenario

6 Conclusion

It has been established a supervisory fuzzy control to minimize the effects caused by the time delay due to communication into the network which is designed through codesign strategy. This approach introduces the time delay produced by scheduling approach named Priority Exchange Procedure. With this fuzzy model a fuzzy control is designed and the stability analysis is proposed for this controller. This approach shows that the system with a time delay smaller than sampling period but with a complex behaviour maintains the stability, the stability analysis for time varying delay and a bound for this delay remain a work in the future.

Although the example related to the time delays is fairly demonstrative it becomes challenging in terms of the dynamic scheduling approach where local time delays is pursued according to eqn. 10 in a general form and implemented through state flow tool in each node.

Acknowledgments

The authors acknowledge the support of UNAM-PAPIIT IN100813, CONACYT 176556 and PICCO 10-53.

Bibliography

[1] Almeida, L. et al (2002); The FTT-CAN protocol: why and how. *IEEE Transactions on Industrial Electronics*, 49(6):1189-1201.

[2] Benítez-Pérez, H. et al (2013); Networked Control Systems Design considering Scheduling Restrictions and Local Faults using Local State Estimation, *International Journal of Innovative Computing, Information and Control (IJICIC)*, 9(8): 3225-3239.

[3] Benítez-Pérez, H. et al (2012); Networked Control Systems design considering scheduling restrictions, *International Journal on Advanced Fuzzy Systems*, http://dx.doi.org/10.1155/2012/927878.

[4] Benítez-Pérez, H. et al (2012); Networked Control Systems Design considering Scheduling Restrictions and Local Faults, *International Journal of Innovative Computing, Information and Control (IJICIC)*, 8(10):8515-8526.

[5] Benítez-Pérez, H.; García-Nocetti, F. (2005); *Reconfigurable Distributed Control*, Springer.

[6] Buttazo, G. (2004); *Hard Real-Time Computing Systems*, Springer.

[7] Czornik, A.; Swierniak, A. (2003); *On Direct Controllability of Discrete Time Jump Linear System*, Tech. Rep., Mathematical Biosciences Institute, The Ohio State University.

[8] Eidson, J.C. et. al (2002); IEEE-1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, *IEEE Standard*, 1588-2002.

[9] Fridman, E.; Shaked, U. (2003); Delay-dependent stability and H∞ control: constant and time-varying delays, *International Journal Control*, 76(1):48-60.

[10] Lian, F. et al (2002); Network Design Consideration for Distributed Control Systems, *IEEE Transactions on Control Systems Technology*, 10(2):297-307.

[11] Liu, J. (2000); *Real-Time Systems*, Prentice Hall.

[12] Méndez-Monroy, P.E.; Benítez-Pérez, H. (2009); Supervisory Fuzzy Control for Networked Control Systems, *International Journal Innovative Computing, Information and Control Express Letters, ICIC-EL, 3-2*, 233-240.

[13] Méndez-Monroy, P.E.; Benítez-Pérez, H. (2011); Codesign Strategy Based Upon Fuzzy Control for Networked Control Systems and a Scheduling Algorithm, *IEEE International Conference on Networking, Sensing and Control (ICNSC)*, 221-226.

[14] Moarref, M.; Rodrigues, L. (2015); Piecewise Affine Networked Control Systems, *IEEE Transactions and Control of Network Systems*, DOI 10.1109/TCNS.2015.2428452.

[15] Nilsson, J. (1998); *Real-Time Control Systems with Delays*, Ph.D. Thesis, Lund Institute of Technology, Dept. of Automatic Control.

[16] Quanser Inc.(2006), *Magnetic Levitation Experiment*, Quanser Consulting.

[17] Tzes, A. et. al (2005); Development and Experimental Verification of a Mobile Client-Centric Networked Controlled System, *European Journal of Control*, 11(3): 229-241.

[18] Walsh, G. C. et. al (1999); Stability Analysis of Networked Control Systems, *American Control Conference*, 2876-2880.

[19] Wang, Y.; Sun, Z. (2007); H-inf Control of Networked Control Systems Via LMI Approach, *International Journal of Innovative Computing, Information and Control*, 3(2):343-352.

[20] Xiao, L. et al (2000); Control with Random Communication Delays via a Discrete-Time Jump System Approach, *American Control Conference*, 3:2199-2204.

[21] Yu, M. et al (2003); An LMI Approach to Networked Control Systems with Data Packet Dropout and Transmission Delays, *International Journal of Hybrid Systems*, 3(2):3545 - 3550.

[22] Zhang, W. (2001); *Stability Analysis of Networked Control Systems.* Ph.D. thesis, Case Western Reserve University, Dept. of Electrical Engineering and Computer Science.

[23] Zhu, X. et al (2008); State Feedback Control Design of Networked Control Systems with Time Delay in the Plant, *International Journal of Innovative Computing, Information and Control*, 4(2): 283-290.

[24] http://www.mathworks.com/help/robust/lmis.html (retrieved on July 20, 2015).

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):194-208, April 2016.

Energy Synchronized Transmission Control for Energy-harvesting Sensor Networks

Z. Fan, X. Liu

Zuzhi Fan*
Department of Mathematics, Jinan University
Guangzhou 510632, China
*Corresponding author: tfanzz@jnu.edu.cn

Xiaoli Liu
Computing Center, Jinan University
Guangzhou 510632, China
txlliu@jnu.edu.cn

Abstract: Energy harvesting and recharging techniques have been regarded as a promising solution to ensure sustained operations of wireless sensor networks for long-term applications. To deal with the diversity of energy harvesting and constrained energy storage capability, sensor nodes in such applications usually work in a duty-cycled mode. Consequently, the sleep latency brought by duty-cycled operation is becoming the main challenge. In this work, we study the energy synchronization control problem for such sustainable sensor networks. Intuitively, energy-rich nodes can increase their transmission power in order to improve network performance, while energy-poor nodes can lower transmission power to conserve its precious energy resource. In particular, we propose an energy synchronized transmission control scheme (ESTC) by which each node adaptively selects suitable power levels and data forwarders according to its available energy and traffic load. Based on the large-scale simulations, we validate that our design can improve system performance under different network settings comparing with common uniform transmission power control strategy. Specially, ESTC can enable the perpetual operations of nodes without sacrificing the network lifetime.
Keywords: Wireless sensor networks, Energy harvest, Transmission control.

1 Introduction

The advance of energy harvesting and recharging techniques makes it is feasible to build long-term sensor networks for cyber-physical applications [1, 2]. In such energy-harvesting networks, sensor node with extended functional units can continuously extract energy from ambient environment, such as solar power, wind energy resource, motion and wireless charging. Although energy-harvesting sensor networks can obtain renewable energy, they impose several challenges. Energy harvesting opportunities and rate are highly environmental-dependent, and usually related to the spatiotemporal distribution of sensor nodes. For example, in solar-powered networks, the harvested energy may vary significantly with node position (e.g. under the sun or shadow) or weather patterns (e.g. cloudy or sunny). Moreover, the energy storage units, such as batteries or capacitors are limited in power capacity and have been shown to be leakage-prone [8]. Therefore, it is impossible to operate the sensors at full duty-cycle even in such energy-replenishing networks.

In consideration of harvested energy, the existing power management solutions have been widely exploited in wireless sensor networks [3, 7, 9, 10]. These approaches can be classified into two categories, one is dynamic duty-cycle scheduling schemes [3, 10], which are based on assumption that the working period of nodes could be scheduled to meet the requirement of data

forwarding. However, the work schedule in many scenarios is often dependent on the application requirements, such as sensing coverage and tracking delay [11]. The other kind is focused on the energy-aware routing [7, 9, 11]. All these schemes can optimize the network performance under their supposed application scenarios.

In this paper, we study the efficient utilization of harvested energy from the perspective of transmission power control. Our motivation is quite straightforward. We observe that the increasing of transmission power can be beneficial to improving network performance, such as packet delivery ratio and delay. Meanwhile, the transmission power is directly associated with energy consumption of data transmission. Thus, we introduce the transmission power control to improve network performance while guaranteeing the sustainability in energy-harvesting sensor networks. Specifically, we propose Energy Synchronized Transmission Control scheme (ESTC), a middle layer between application and network layer. With ESTC, each node can adaptively adjust its transmission power based on its energy-harvesting capability and traffic overload. ESTC can be seamlessly integrated with current routing algorithms so that different optimization objectives can be achieved. We also present a backoff approach to avoid transmission collision among concurrent data forwarding.

Our major contributions of this work are summarized as follows. We first describes energy-harvesting sensor nodes and the duty cycle model. To balance the delivery delay and energy efficiency, we propose a distributed energy synchronized transmission control approach to find the best power level for each forwarder without sacrificing network lifetime. At last, we perform extensive large-scale simulations to validate the proposed scheme. Working with different routing policies, the simulation results show that ESTC can: i) reduce end-to-end delivery delay; ii) synchronize the energy consumption among different sensor nodes; iii) balance the traffic load to increase the node lifetime.

The rest of the paper is organized as follows: Section 2 briefly presents the related work. The concrete design of our scheme is discussed in Section 3 and the simulation results are presented in Section 4. Section 5 concludes the paper.

2 Related Work

There are two research fields related to our design: energy harvesting techniques and transmission power control.

Recently, energy harvesting and recharging technologies have been developed to ensure the sustainability of sensor networks. Many motes or platforms are designed to collect and storage these energy from environment [8]. To fully utilize the replenished energy, different power management [11] and duty-cycle based schemes [3] have been proposed. Kansal et al. [11] have proposed temporal-based approaches to adjust the duty-cycle of sensor node in order to optimize the network performance. In [10], Gu et al. first put forward the concept of energy synchronization communication, by which each node adaptively adjusts its own active instances according to the available energy budget so that the cross-delay over node can be minimized. Challen et al. [9] present IDEA, an integrated energy-aware architecture to address the energy dynamic issue of sensor nodes. In particular, they propose a holistic architecture to trade off energy objective function and other application-defined utility, such as low power listening, energy aware routing and distributed localization. In [4], Guo et al. study the joint problem of mobile data gathering and wireless charge. Differing from above schemes, their motivation is to study the mobility scheduling for efficient energy recharging and data collection. In the direction of low-duty-cycle networks, many recent works have been proposed to reduce the delivery delay for different traffic patterns. Gu et al. [17] suggest that the communication delay could be bounded with the duty-cycle adjustment of sensor nodes. Besides the delay optimization, Liu et al. [12] study the

joint routing and sleep-scheduling problem, which has been proved a non-convex problem. They transform it into equivalent sigmoidal program by relaxing the flow constraints and then solve it with iterative geometric programming.

Transmission power control techniques have been widely used to optimize network performance in wireless networks. In particular, most of them are focused on the topology control [13]. Wattenhofer et al. [13] proposed a location-based, distributed topology control algorithm to balance the network connectivity and the network lifetime. At first, each node starts a neighbor-discovery process with a lower transmission range and then gradually increases its transmission radius until either one node is found in each cone of given degree or the maximum transmission power is reached. Then, a redundant edge removal process is performed in order to reduce the nodes degree and thus increase network throughput. In [6], Cheng et al. study the throughput optimization problem with transmission power control in sensor networks. They propose algorithms in order to minimize the total transmission power and total interference. Based on various link models, both computing algorithms and heuristics are discussed for the purpose of throughput maximization. In [14], Cotuk et al. analyze the impact of varied transmission power control strategies on network lifetime. Specially, they study the effect of power levels discretization on energy consumption, which is significant for practical research because the levels of transmission power are usually discrete in reality. In [15], Fan et al. propose a delay-bounded transmission power control scheme for the performance optimization in low-duty-cycle sensor network. Under the given delay bound, a cross-layer transmission power control approach is presented so that all data delivery could be achieved with minimum energy cost. In [5], Berbakov et al. consider the similar application scenario as our design. However, their goal is to find the optimal power allocation in order to maximize the total throughput within given deadline. They also assume that the storage capacity of sensor nodes is infinite and the leakage effect is negligible.

However, none of works have considered the utilization of transmission control strategy to achieve performance improvement in sustainable sensor networks.

3 Energy Synchronized Transmission Control with Harvested Energy

In this section, we present the design of energy synchronized transmission control algorithm.

3.1 Network Model

We assume a sensor network with N energy-harvesting nodes, each of them has a fixed number of discrete transmission power levels, i.e., $p_i, (1 \leq i \leq k)$, where k is the maximum number of adjustable transmission power levels. Figure 1 illustrates the configuration of energy-harvesting node, which replenishes energy from surrounding environments, receives data packets and delivers them to the sink at possible transmission power level.

Also, we suppose that all nodes are scheduled to work in a duty-cycled mode. As shown in Figure 2, a sensor node is in either active state or a dormant state. When a node is in the active state, it can transmit or receive packets from neighboring nodes. While a node is in the dormant state, it turns off all function modules except a timer to wake itself up. For successful communication, the sender should be aware of the time slots and have to wait for its receiver to wake up before it can send a packet. We define *sleep latency*, $s_{ij}(t)$ as the time interval from the moment the sender i has a packet ready to be sent at time t to the moment that the receiver j is in the active state. Without loss of generality, we suppose T is the common working period of the whole network, which can be further divided into a number of time slots with equal length. To simplify, the length of time slot is appropriate for a round-trip transmission time, τ. Based

on such assumptions, the working schedule, Γ_i for node i can be uniquely represented as a set of active time slots, i.e., $\Gamma_i = \{t_1^i, t_2^i, ..., t_K^i\}$, where K is the number of time slots that the node is in the active state. For example, the work schedule in Figure 2 is $\{2, 6, 8\}$.

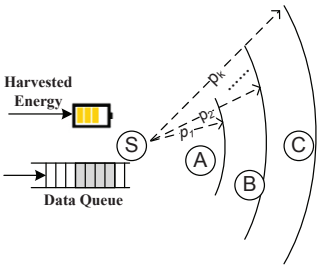


Figure 1: Energy-harvesting node.

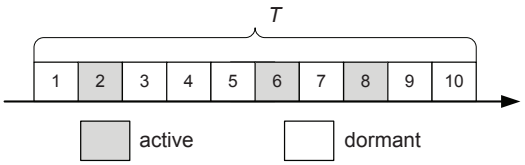


Figure 2: Working schedule.

In our design, it assumes that all node know the schedule of its one-hop neighbors. Sleep latency is the main component of delivery delay in such energy-harvesting networks.

3.2 Design Objectives

Given k available power levels, the sender have k options to relay the data packets in its data queue. We take energy harvesting into consideration and look into the following issue: *what's the optimal transmission control policy with harvested energy?* Generally, it is necessary for node to consider three aspects for such transmission decision.

- Energy budget: In general, the sender can select larger transmission power levels to reduce the sleep latency while it has enough energy budget. On the contrary, it should lower down transmission power to save energy. As shown in Figure 1, node A may increase its transmission power in order to reduce sleep latency if it has additional energy supply.
- Traffic load: Obviously, the traffic load has a significant impact on the transmission decision. More energy supply is needed when there are more data packets to be delivered in the data queue.
- Routing metric: Given transmission power, there are usually multiple potential forwarders available for current node. However, different routing metrics are designed for varied objectives. For example, the delay is the main issue to be addressed in duty-cycled sensor networks.

If there is no energy constraint, we can select the maximum available transmission power for each node in order to obtain the minimized end-to-end delay. However, such a naive and uniform transmission power control policy can waste precious energy resources and incur more transmission interference and collision. Take the dynamics of energy, traffic overload and routing strategies into consideration, our design goals include:

- **Delay Optimization.** In real world, sensor nodes are deployed to monitor or response emergency surveillance. Instead of hard deadline, our protocol provides an adaptive transmission control approach to reduce delivery delay.
- **Energy synchronization.** In energy-harvesting sensor network, each node has different energy-gathering and storage capability. Taking energy leakage into account, it is important to synchronize the demand with energy supply. In other words, we tend to consume as much energy as possible while providing the sustainability of network.

- **Balancing Traffic Overload.** Differing from wired network, the bandwidth and energy are constrained resources in sensor networks. Therefore, it is important to perform data delivery over multiple forwarding paths from source nodes to the destination. Our protocol dynamically switches forwarding among various potential forwarders according to routing metrics.
- **Localized Behavior.** It is important to keep the protocol as scalable as possible since the global coordination among hundreds of nodes may incur more energy consumption. Therefore, all behavior of our protocol are localized to achieve high scalability and low overhead.
- **Transmission Collision Avoidance.** Transmission interference and collision may happen when multiple nodes within transmission range try to send packets simultaneously. It is necessary to introduce the corresponding mechanism to reduce such collision.

3.3 Protocol Architecture

In this section, we propose an energy synchronized transmission control scheme (ESTC) which adaptively selects the optimal transmission power at transmission layer and diverts traffic overload through different forwarders at network layer in order to improve network performance with the extra harvested energy. In specific, our protocol includes the following components.

- ESTC module.
- Energy estimation module.
- Delay estimation module.

As shown in Figure 3, ESTC is the kernel module, which is responsible for the selection of approximate transmission power level and next-hop forwarder. The data queue in upper layer is holding all data packets to be relayed. Assuming all data packets have the same size, the traffic overload can be represented as the length of data queue. Energy and delay estimation are two modules that help ESTC to make the forwarding decision. In other word, ESTC will select the transmission power level and corresponding forwarder according to the available energy and feedback from neighboring nodes. The detail of these modules is discussed in the following sections.

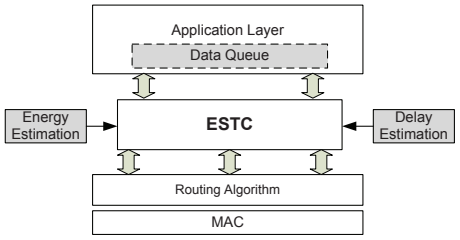


Figure 3: ESTC Architecture.

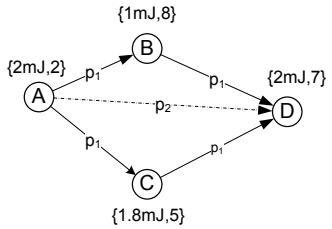


Figure 4: Example of Energy and Delay Estimation

3.4 Energy Estimation

To carry on energy synchronized communication, it is significant to know the amount of energy that can support data delivery. In sustainable sensor networks, the harvested energy is

usually unpredictable and changes significantly over time [8]. Energy estimation module traces both energy load as well as the harvesting rate on a node. Let the battery level for node s at the time t_0 be $B_s(t_0)$, a common model to estimate the available energy at time t_1 is:

$$B_s(t_1) = B_s(t_0) + \int_{t_0}^{t_1} H(t)dt - \int_{t_0}^{t_1} L(t)dt \tag{1}$$

Here, $H(t)$ and $L(t)$ are the energy harvesting rate and consuming rate at time t. Other models, such as online assess model in [16] can also be used, but they are usually focused on specific harvesting platforms, for example, the supercapacitor-powered sensor node. Instead, we assume a general in-site model by reading the energy value in the later simulations.

Notice that, the primary objective of our design is to pursue the network sustainability, which can be represented as the network lifespan. To prolong the network lifetime, it is required to balance the energy consumption among sensor nodes. In short, the issue is how much energy can be used for each node without compromising the network lifetime? Assuming the remaining energy of the network is known a prior, we can formally define the energy availability of given node.

Definition 1. ***(Energy Availability).*** Available energy of node s, B_s^a is the extra energy which can be used for data delivery but without reducing the network lifetime. Assuming the average energy of the network ($B\prime$) is known a prior, we have $B_s^a = B_i(t) - B\prime(t)$, where $B_i(t)$ is the remaining energy of node at time t.

However, it is usually inefficient to update the average energy level of the whole network from time to time. One possible solution is to approximate the average energy with the energy load of ancestor nodes. As shown in Figure 4, node A, B, C can forward their data packets to the sink via node D and their remaining energies are 2mJ, 1mJ and 1.8mJ, respectively. Accordingly, the available energy for node D is the difference between its own energy and the average value, i.e., 0.4mJ. One may argue that we can use the lowest energy level (1mJ of node B) as the baseline. However, we do not assume a fixed transmission power and forwarding path. That is, node A may switch to node C or even increase transmission power to reach node D directly for the energy-efficiency. Obviously, it is not accurate to take the minimum residual energy as the reference energy.

Maintaining the accuracy of energy estimation requires periodical exchange of these information. In practice, piggybacking on the normal data traffic can be used to reduce this control overhead. When a node receives message from its ancestor, it will recalculate the average energy value and then put it into the header of data packet. In such way, each node can trace the available energy of the network.

3.5 Fitting Routing Algorithms

With given transmission power p_i, there may be multiple available candidates for data forwarding in sensor networks. In this section, we will show how ESTC works with various routing metrics, such as link-quality-based routing (ETX) [18], delay-based routing (DESS) [19] and power-aware routing (PAR) [22]. To be scalable, it assumes that all routing decisions are made based on the local information.

Link-quality-based Routing

ETX is a link-quality-based routing algorithm, in which the expected transmission count is taken as routing metric. The one-hop ETX is the average number of transmission required to

send a packet over a link, which is usually described as the reciprocal of link quality. However, wireless link is often extremely unreliable in the real environment. The pair-wise link qualities, described as the packets reception ratio (PRR) could be very different under varied transmission power. In practice, link quality is usually evaluated by periodically broadcasting probe messages as that in [10], which is a little energy-consuming in our design due to the adjustable transmission powers.

On the other hand, reception signal strength indicator (RSSI) has a close relationship with PRR according to the empirical results [20]. Specifically, there is a clear threshold for RSSI to achieve a nearly perfect link quality. To save energy, we use the RSSI instead of PRR as the metric to filter qualified forwarding candidates. Generally, a higher transmission power tends to bring better link quality [20]. In detail, the sender periodically monitors the received signal strength from its one-hop neighbors and then evaluates their RSSIs. For given transmission power, we can select the candidate with the strongest RSSI as next hop.

Delay-based Routing

In duty-cycled sensor network, sleep latency is often in the order of seconds, while propagation delay and processing delay (in the order of milliseconds) can be ignored. Therefore, it is necessary to estimate the sleep latency. Assuming each node is assigned a predefined work schedule, we can calculate the sleep latency by the waiting time from the ready time to the moment that it is sent out. To do that, each node only needs to share its work schedule with their neighboring nodes. In a resource-constrained environment, it is energy-consuming and non-scalable to estimate the end-to-end delay among different sender-receiver pairs. Instead, we use one-hop delay value as the evaluation metric.

In ideal network with perfect link, one-hop delay can be represented as $s_{ij} = (t_j - t_i)$, where t_j, t_i are the wake-up time of transmission pair. For example, the sleep latency from node A to B in Figure 4 is 8-2=6. Notice that, the end-to-end delay is also related to the length of forwarding path. To model such parameter, we present the one-hop relative delay, d_{ij} as,

$$d_{ij} = s_{ij} * \frac{D_j}{D_i}. \tag{2}$$

Here, D_j, D_i are the distances from the sender and next-hop to the destination, respectively. Assuming two candidates wake up at the same time, the nearer one would be selected as next hop. In real scenario with unreliable links, we can evaluate delivery delay according to the model proposed in [21].

Energy-based Routing

In ESTC, the selection of transmission power is from the view of sending node. However, it is possible to integrate ESTC with power-aware routing algorithm in order to maximize energy efficiency and network lifetime [22]. To do that, each node can periodically collect energy information from its one-hop neighbors and make the forwarding decision in time. In specific, we select the metric aiming at maximizing the lifetime of all nodes. As a result, the neighboring node with maximum consumed energy for each packet would be selected as the potential forwarder. Similar as link-quality-based routing, the sender exchanges the statistical information with its neighbors, including residual energy, queue size so that the optimal forwarder could be chosen. Taking Figure 4 as an example, if the lengths of their queue are the same, node C instead of node B would be selected since it has more residual energy.

3.6 Collision Avoidance

Though there is low data traffic in duty-cycled sensor networks, it is possible that multiple transmissions among neighboring nodes are collided when transmission power is increased. Meanwhile, the concurrent transmission happens only when their receivers wake up at the same time. To resolve the conflicts, we introduce transmission-power-based backoff approach. When a node intends to begin a transmission, it first backs off for a period of time at the begin of a slot. The duration of the backoff depends on the power level used in the transmission. The higher the transmission power, the shorter the back off duration. When multiple nodes within communication range decide to send packets, they back off first before transmission and the one with highest power level starts first. Other nodes listen to the channel first after the backing-off time. Once catching the ongoing transmission, they will abort their own transmission and insert the data packet with updated timestamp into the data queue.

Suppose the backoff time bound is T_b and the maximum number of concurrent transmissions is C, we can divide T_b into C slots for different backoff durations. A sender can compute its backoff duration t_b with the following equation.

$$t_b = \lfloor C(1 - \frac{i}{k}) \rfloor \frac{T_b}{C} + X, 1 \leq i \leq k. \tag{3}$$

where i is the number of transmission power level and X is a random number generated from $[-\frac{T_b}{C}, +\frac{T_b}{C}]$ if $i \leq k$ and from $[0, +\frac{T_b}{C}]$ if $i == k$. This ensures that the backoff time is positive and within the backoff bound. The random period can reduce the chance of collisions when two or more nodes use the same power level. By using such backoff method, we avoid conflicts but also save energy since the transmission with higher power level starting early can be heard by more potential senders.

3.7 Energy Synchronized Transmission Control (ESTC) Scheme

Based on the energy estimation and routing algorithm, ESTC protocol can make the approximate forwarding decision. The detailed process of transmission power decision is described in Algorithm 5.

To efficiently synchronize the harvested energy, ESTC takes an energy adaptive strategy based on the feedback of energy estimation and real-time traffic load. If there is no additional energy budget, the current node sends the packet with minimum transmission power level. Otherwise, the sender can increase its transmission power in order to support the packet delivery in data queue. Our first step is to decide the amount of energy that can be used by the data delivery per packet. Assuming that each packet has the same length, the energy consumption for data delivery is only dependent on the used transmission power. In other word, the sender can calculate the energy consumption for given transmission power level. Given the frame size of data packet (F) and the data rate (r), we can compute the energy consumption of data transmission, E_i by given transmission power p_i with the following equation.

$$E_i = P_i * \frac{F}{r}. \tag{4}$$

Here, P_i is the power consumption for given transmission power level p_i, which is usually dependent on the wireless radio. Next, we can decide the appropriate transmission power level by comparing E_i with the available energy (See Line 1-9).

With the selected transmission power, ESTC sends packet in the order of data queue according to the given routing policy (See Line 10-16). Since data packets can only be delivered

when the next-hop forwarder wake up in duty-cycled sensor network, the data queue in upper layer is holding data packets generated by current node or received from the other nodes. Each data packet includes the following fields, (ID, TimeStamp, TransmissionTimes). To prioritize the delay, we suppose all data packets are stored in the order of their generated time. For example, it needs to forward data packets via the earliest wake-up node with DESS. If the transmission succeeds, it would update the available energy and fetch the next packet from data queue. Otherwise, ESTC inserts the failed packet into the data queue and waits for the next schedule. The packet would be dropped if it is out of the maximum allowable transmissions ($TransmissionTimes$). Notice that, the above algorithm is locally executed at the individual node, which is completely distributed.

ALGORITHM 1: Energy Synchronized Transmission Control at time t

Require: the number of packets in data queue, n;
Require: the average energy of ancestor nodes, $B\prime$;
Require: the amount of remaining energy, B_i;
Require: the maximum number of concurrent transmissions, C;
Require: the routing algorithm, $routingPolicy$;
Require: the backoff time bound, T_b;

$p_a \leftarrow p_1$;
$B_s \leftarrow B_i(t) - B\prime(t)/n$;
for all transmission power $p_i \in [p_1, p_k]$ **do**
  $E_i \leftarrow P_i * \frac{F}{r}$;
  if $(E_i > B_s^a)$ **then**
    break;
  end if
  $p_a \leftarrow p_i$;
end for
if (routingPolicy is ETX) **then**
  select next hop (n_h) according to link quality;
else if (routingPolicy is DESS) **then**
  select next hop (n_h) according to sleep latency;
else if (routingPolicy is PAR) **then**
  select next hop (n_h) according to energy level;
end if
for $i = 0$ to k **do**
  if $(i == k)$ **then**
    $X \leftarrow rand(0, +\frac{T_b}{C})$;
  else if $(i < k)$ **then**
    $X \leftarrow rand(-\frac{T_b}{C}, +\frac{T_b}{C})$;
  end if
end for
$t_b \leftarrow \lfloor C(1 - \frac{i}{k}) \rfloor \frac{T_b}{C} + X$;
Back off the time, t_b.
Fetch and send packet to n_h with power level p_a.

4 Performance Evaluation

In this section, we validate the performance of energy synchronized transmission power control scheme. In specific, we assume a data collection scenario consisting of energy-harvesting sensor nodes, which is a common communication pattern. Source nodes periodically sense and generate data packets, then deliver them to the sink node through multi-hop forwarding path. Due to energy efficiency, all nodes except the sink are presumed to work with low-duty-cycle mode.

4.1 Baseline and Selection of Routing Algorithms

To verify the effectiveness of our design, we compare ESTC scheme with those that do not use energy synchronization mechanism, i.e., the uniform transmission power control (termed UTPC later). To verify our design, the ESTC is integrated with the various routing algorithms.

- Link-quality-based: ETX [18] is proposed to minimize the expected transmission count for multi-hop data communication.
- Delay-based: DESS [19] is presented in order to minimize delivery delay for duty-cycled sensor networks.
- Power-Aware-Metric: PAR [22] is proposed to minimize the energy consumption and then prolong the network lifetime. In our experiment, we take the relative energy budget as routing metric.

Notice that, all routing algorithms can be easily integrated with our design. For given transmission power, each node selects the corresponding candidate according to the above routing metrics.

4.2 Simulation Setup

We assume that all sensor nodes are randomly deployed in a $200m \times 200m$ square field, where 40 nodes are selected as data source and the sink is located in the right corner of sensor area. The average data rate of source node is 2 packets with frame size 64B for each working period. Without otherwise specified, we set radio parameters strictly according to the CC2420 radio hardware specification [23]. In detail, we select 8 typical transmission power levels, -25dBm, -15dBm, -10dBm, -7dBm, -5dBm, -3dBm, -1dBm, 0dBm indexed from level 0 to level 7. The energy model is identical to the practical measures of CC2420, i.e., the corresponding transmission power ranges from 29.04mW to 57.42mW. The energy consumption of data reception is 62mW.

Each experiment is repeated 30 times with different deployments and working schedules generated by random seed. For each experimental setting, the result is averaged over 100 source-to-sink communications under given network size and density. To emulate the energy-harvesting environment, each node is supposed to increase its energy resource at a stochastic charging rate.

In the simulation, three performance metrics are evaluated: i) the E2E delivery delay, defined as the total time spend for the delivery of a packet; ii) the energy efficiency, defined as the standard deviation of remaining energy for all nodes within the network; iii) the network lifetime, defined as the number of time slots from the beginning to the time when first node is running out of its energy. We have to mention that it is a dynamic concept in energy-harvesting network, only representing the current status of network. With the replenishment of energy, the dying nodes can refresh and join the data forwarding again.

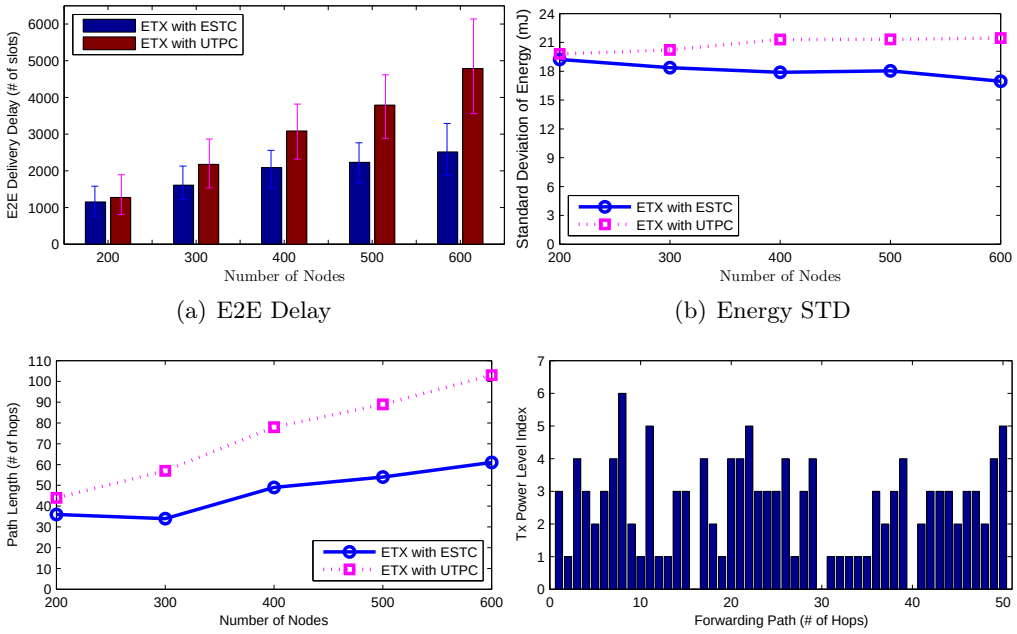
4.3 Performance Evaluation

This section evaluates the E2E delivery delay, energy consumption and network lifetime for different schemes. Moreover, we compare ESTC and UTPC scheme under varied network settings.

Working with ETX

In this section, we first study the delivery delay of both ESTC and UTPC while the number of nodes changes from 200 to 600. As can clearly be seen from Figure 5(a), ESTC has a smaller delay than UTPC under all node densities. For example, ESTC reduces the E2E delay by 47% compared with UTPC when the number of nodes is 600. It can also observed that the delay for ETX-based schemes increases with the number of nodes. More nodes are deployed in the network, more potential candidates are available. On the other hand, those candidates near to the sender are more likely selected, leading to a longer data forwarding path for ETX-based forwarding. To verify, we plot the the forwarding length of both schemes in Figure 5(c), which shows the average length of both ESTC and UTPC is increasing with nodes density. More importantly, the average length of ESTC is much shorter than UTPC in all cases. For example, the maximized length for UPTC is 103 while the corresponding length for ESTC is 61.

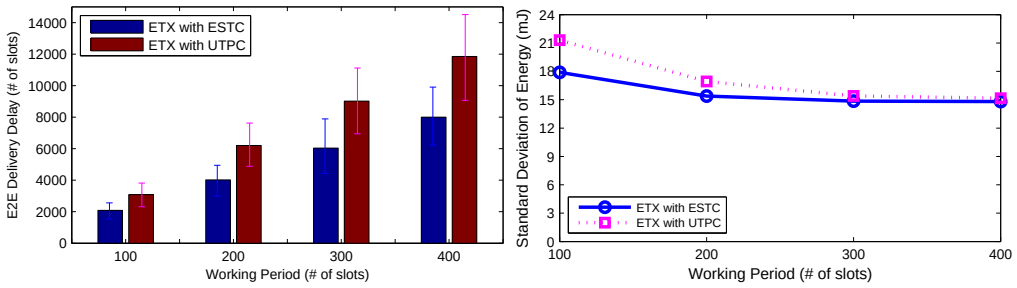
Figure 5(b) shows that ESTC has smaller standard deviation of energy than UTPC, representing that nodes with ESTC has more balanced energy consumption in the process of data collection. The reason is that ESTC tends to distribute the energy consumption among different nodes by adjusting the transmission power. We plot the transmission power levels (TPLs) used in the data forwarding process of ESTC as Figure5(d). It can be clearly seen that nodes with ESTC can adaptively select transmission power according to the availability of energy resource and link quality. From the figure, it is observed that many nodes select very low transmission power, such as level 0 or 1 in dense deployed area. Notice that, UTPC tends to select the same transmission power for all data delivery, resulting in higher energy consumption.



(a) E2E Delay (b) Energy STD

(c) Path Length (d) TPL Diversity

Figure 6: Impact of Node Density for ETX.



(a) E2E Delay (b) Energy STD

Figure 7: Impact of Duty Cycle for ETX.

Figure 6(a) shows the impact of working period on network performance, where the E2E delivery delay increases with the working period. This is because the duty cycle is reduced while the working period increases, leading to the prolong of sleep latency per hop. For energy efficiency, we observe the similar result that the energy dissipation among nodes is more balanced for ESTC than UTPC as Figure 6(b).

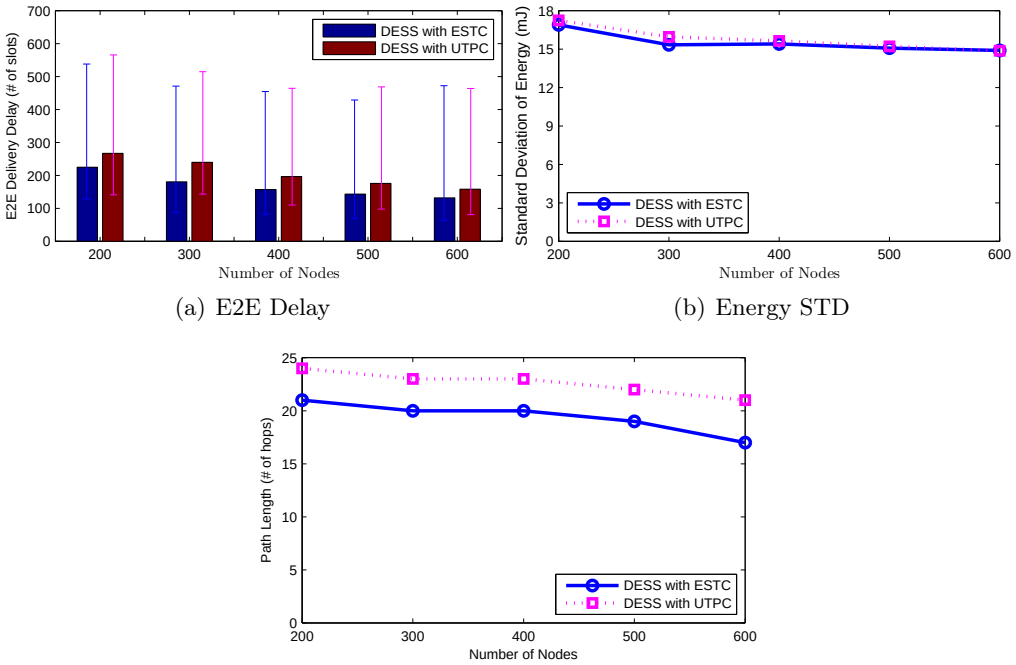
Working with DESS

In this section, we study the network performance of both ESTC and UTPC working with DESS. Again, ESTC has a smaller delay than UTPC for DESS under all node densities. Figure 7(a) shows that the delay for DESS-based scheme decreases as node density, in which more potential candidates with earlier wake-up schedule are provided. Moreover, DESS-based schemes have much lower delays compared with those ETX-based schemes. For instance, the average E2E delay for DESS is around 200 under all node densities. On the contrary, the value for ETX is more than 1000 as shown in Figure 5(a). The rationale behind is that DESS prioritizes the delivery delay since each node always selects the earliest wake-up neighbor to forward data packets. The other reason is that the path length for DESS routing is much shorter than ETX as shown in Figure 7(c). Similarly, ESTC outperforms UTPC on energy efficiency due to the adjusting of transmission power levels as shown in Figure 7(b).

Figure 8(a) shows the average delay under different working periods. We can see that the delivery delay increases with the working period of nodes. For example, the average E2E delay increases from 156 to 573 for ESTC. Totally, with energy synchronization, ESTC can reduce delivery delay by 20% than UTPC under all working periods. Figure 8(b) shows the energy efficiency under varied duty cycles, which proves the slight superiority of ESTC over UTPC. The main reason is that the schedule of nodes is assumed to be fixed in the whole lifetime so that the same node is usually selected as the forwarder in multiple times. On the other hand, the total energy consumption for DESS-based approach is much less than ETX-based scheme due to shorter forwarding path.

Working with PAR

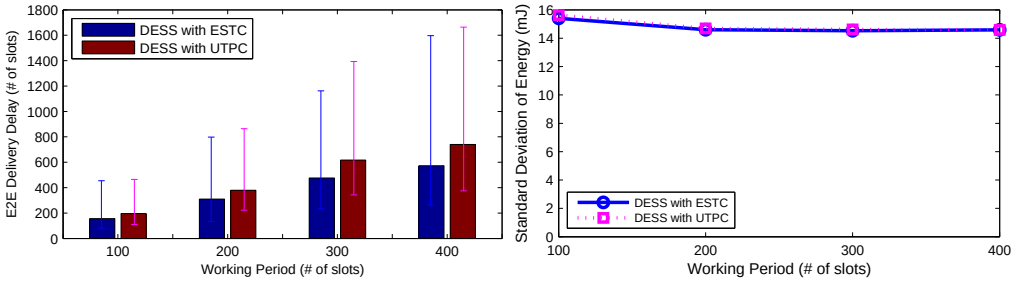
In this section, we study the network performance of proposed schemes with power-aware routing algorithm. In the simulation, we assume that each node can harvest energy within predefined period and then measure the network lifetime. In special, the average charging rate of nodes is set from 0 to 3. Figure 9(a) shows that ESTC can maintain longer lifetime than UTPC under all energy charging rates. Notice that, the network lifetime with energy-harvesting capability is much longer than that of static sensor network. For example, the network lifetime lasts 3.7 million of slots with average charge rate 3, around 9 times of the lifespan when there is



(a) E2E Delay (b) Energy STD

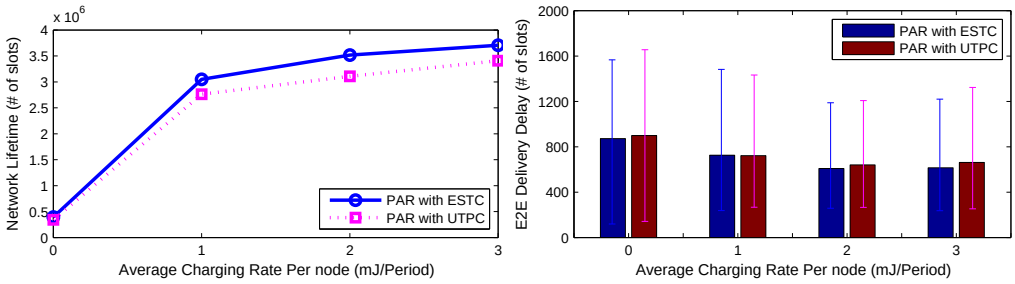
(c) Path Length

Figure 8: Impact of Node Density for DESS.



(a) E2E Delay (b) Energy STD

Figure 9: Impact of Duty Cycle for DESS.



(a) Network Lifetime (b) E2E Delay

Figure 10: Impact of Charging Rate for PAR.

no extra harvested energy. Also, we deploy the E2E delay for both UTPC and ESTC in Figure 9(b), which demonstrates ESTC can still reduce the delivery delay even working with the power aware routing policy. In fact, when the packet is delivered along with an energy-rich path, it is more likely to be transmitted at high power level, leading to a lower sleep latency.

5 Conclusion

Harvesting energy technique provides opportunities for the substantiality of resource-constrained sensor networks. To efficiently utilize the harvested energy, we propose energy synchronization transmission control scheme which can work together with different routing strategies. In specific, ESTC adaptively selects the suitable transmission power according to the energy availability and traffic load in order to reduce delay and balance the energy consumption. We verify the effectiveness of our design by conducting large-scale simulations, showing that ESTC can reduce delivery delay and energy consumption without compromising network lifespan compared with uniform transmission power control design.

Acknowledgment

This work was supported in part by NSFC under grant No. 61373125, 61572233 and "the Fundamental Research Funds for the Central Universities" under grant No. 21615442.

Bibliography

[1] M. Li et al (2009), Canopy Closure Estimates with GreenOrbs: Sustainable Sensing in the Forest, *ACM Sensys, Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, 99-112.

[2] V. Dyo et al (2010), Evolution and Sustainability of a Wildlife Monitoring Sensor Network, *ACM Sensys, Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, 127-140.

[3] Vigorito, Christopher M et al (2007), Adaptive control of duty cycling in energy-harvesting wireless sensor networks, *IEEE SECON*, 21-30.

[4] ST Guo et al (2013), Mobile data gathering with wireless energy replenishment in rechargeable sensor networks, *IEEE Infocom*, 1932-1940.

[5] Berbakov, Lazar et al (2014), Joint optimization of transmission policies for collaborative beamforming with energy harvesting sensors, *IEEE Transactions on Wireless Communications*, 13(7):3496-3509.

[6] Cheng, Maggie X., et al. (2011); Cross-layer throughput optimization with power control in sensor networks, *IEEE Transactions on Vehicular Technology*, 60(7): 3300-3308.

[7] D. J. Vergados et al (2008), Energy-Efficient Route Selection Strategies for Wireless Sensor Networks, *Mobile Network and Applications*, 12:285-296.

[8] T. Zhu et al (2009), Leakage-Aware Energy Synchronization for Wireless Sensor Networks, *MobiSys09, Proceedings of the 7th international conference on Mobile systems, applications, and services*, 319-332.

[9] G.W. Challen et al (2010), IDEA: Integrated Distributed Energy Awareness for Wireless Sensor Networks, *MobiSys10, Proceedings of the 8th international conference on Mobile systems, applications, and services*, 35-48.

[10] Y. Gu et al (2014), Achieving energy-synchronized communication in energy-harvesting wireless sensor networks, *ACM Transactions on Embedded Computing Systems (TECS)*, 13(2s):68.

[11] A. Kansal et al (2007), Power management in energy harvesting sensor networks, *ACM Transactions on Embedded Computing Systems*, 6(4):32.

[12] F. Liu et al (2010), Joint routing and sleep scheduling for lifetime maximization of wireless sensor networks, *IEEE Transactions on Wireless Communications*, 9(7):2258–2267.

[13] R. Wattenhofer et al (2001), Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks, *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 3: 1388-1397.

[14] Cotuk H, Bicakci K, Tavli B, et al. (2014); The impact of transmission power control strategies on lifetime of wireless sensor networks, *IEEE Transactions on Computers*, 63(11): 2866-2879.

[15] Z. Fan et al (2015), Delay-Bounded Transmission Power Control for Low-Duty-Cycle Sensor Networks, *IEEE Transactions on Wireless Communications*, 14(6):3157–3170.

[16] Renner, Christian et al (2014), Online energy assessment with supercapacitors and energy harvesters, *Sustainable Computing: Informatics and Systems*, 4(1):10–23.

[17] Y. Gu and T. He (2010), Bounding Communication Delay in Energy Harvesting Sensor Networks, *Distributed Computing Systems (ICDCS), 2010 IEEE 30th International Conference on*, 837-847.

[18] D. S. J. D. Couto et al (2003), A HighThroughput Path Metric for MultiHop Wireless Routing, *MobiCom 03, Proceedings of the 9th annual international conference on Mobile computing and networking*, 136-146.

[19] G. Lu et al (2005), Delay Efficient Sleep Scheduling in Wireless Sensor Networks, *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, 4: 2470-2481.

[20] S. Lin et al (2006), ATPC: Adaptive transmission power control for wireless sensor networks, *SenSys06, Proceedings of the 4th international conference on Embedded networked sensor systems*, 223-236 .

[21] Z. Fan (2013), Delay-Driven Routing for Low-Duty-Cycle Sensor Networks, *International Journal of Distributed Sensor Networks*, Volume 2013, Article198283, 11 pages, http://dx.doi.org/10.1155/2013/198283.

[22] S. Singh et al (1998), Power-Aware Routing in Mobile Ad Hoc Networks, *Mobile Computing and Networking*, DOI:10.1145/288235.288286, 181–190.

[23] http://www.ti.com/

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):209-223, April 2016.

Influence Model of User Behavior Characteristics on Information Dissemination

S.C. Han, Y. Liu, H.L. Chen, Z.J. Zhang

ShaoChun Han
Beijing Jiao tong University
Beijing, 100044, China
15620009060@126.com

Yun Liu
Beijing Jiao tong University
Beijing, 100044, China
liuyun@bjtu.edu.cn

HuiLing Chen*
1. School of Pharmaceutical Science and Technology Tianjin University
2. TianJin University of Traditional Chinese Medicine
*Corresponding author: 15022613010@163.com

ZhenJiang Zhang
Beijing Jiao tong University
Beijing, 100044, China
zhjzhang1@bjtu.edu.cn

Abstract: Quantitative analysis on human behavior, especially mining and modeling temporal and spatial regularities, is a common focus of statistical physics and complexity sciences. The in-depth understanding of human behavior helps in explaining many complex socioeconomic phenomena, and in finding applications in public opinion monitoring, disease control, transportation system design, calling center services, information recommendation. In this paper,we study the impact of human activity patterns on information diffusion. Using SIR propagation model and empirical data, conduct quantitative research on the impact of user behavior on information dissemination. It is found that when the exponent is small, user behavioral characteristics have features of many new dissemination nodes, fast information dissemination, but information continued propagation time is short, with limited influence; when the exponent is big, there are fewer new dissemination nodes, but will expand the scope of information dissemination and extend information dissemination duration; it is also found that for group behaviors, the power-law characteristic a greater impact on the speed of information dissemination than individual behaviors. This study provides a reference to better understand influence of social networking user behavior characteristics on information dissemination and kinetic effect.
Keywords: SIR, behavior dynamics, scaling laws, information dissemination.

1 Introduction

The analysis target of user behavior time characteristics is the statistical regularities manifested when humans repeatedly engaged in certain things, which was firstly proposed by Poisson introduced the concept of probability in his work of case judgment management, namely Poisson distribution. When human data collection capabilities are limited, the Poisson distribution is widely used as a classic means to depict human activity patterns.

In recent years, with the emergence of high-performance processors and constant enhancement of computer parallel computing power, making the massive social network data processing

become possible. At present, through empirical analysis, research and mining user behavior characteristics of large data and use simulation technology [1]- [8], a large number of scholars make analysis of network relationships and identify potential objective law.

By analyzing massive data of various networks, more and more facts have proven that, user behavior corresponded time interval distribution has obvious heavy-tailed effect, which can be well fitted by power function [16]- [18].

At the same time, scholars use massive data of social networks, from many fields, multi-angle and multi-dimensional human behavior characteristics were studied. For example, X.Song et al. [9] analyzed the geographical distribution of Twitter users, user's neighbor nodes and the degree of correlation coefficient, and Twitter users were grouped. H.Kwak et al. [10] studied the average shortest interval and length of Twitter micro blog, posts survival time, maximum repost depth and user grouping sorting features, the text sorted Twitter users according to the number of fans and Page-Rank value, the final results of the two sorting methods are substantially the same, which is obviously different from the final sorting result obtained by users information forwarding number, indicating that there is not tight dependencies between users information forwarding number and their owned neighbor node. M.Cha et al. [11] by comparing correlation coefficient of Twitter users posts forwarded number, post reply number and the number of neighbor nodes, studied the effect of core users on information dissemination.

The article [12] conducted further analysis of Twitter posts forwarded relevant factors. The article [13] also conducted data analysis of scholarly articles downloads from an economy physics web site, and found that download rate of different papers show exponential decrease per unit time,and the average download rate f and its variance approximately satisfy $f \propto o^{\alpha}$, of which, α is located between 0.6 to 0.9.

The paper [14] extracted sina blog user's interaction data, through network degree distribution analysis, it can be found that in the sina blog, the in-degree and out-degree obeys power-law distribution, but the exponent of out-degree is larger than in-degree, which explains that part of the blog user do not add more friends, and even of users do not have friends, and studies have found that the correlation coefficient of blog network in-link and out-link degree distribution is positive; while the correlation coefficient of out-link and in-link degree distribution is negative.

In this paper, firstly carry out mathematical statistical analysis of users publish information and information reply time intervals in QQ space data set, and by means of SIR [15] model to investigate influence of time characteristics of user behavior on information dissemination process in the social network, and make comparative analysis of even stepping model.

2 Data

In this paper, the author utilized QQ spatial data set. This data set is obtained by the use of crawling program wrote by python language. The program logins QQ space by the way of simulated browser, automatically go to access to interaction information between user and his friends, and write into the corresponding xml file, then read xml into the database by using python parsing, remove user hidden QQ number and other abnormal data, to get the final data set of QQ space. Topological statistics characteristics of the data set is shown: V denotes the number of nodes(4800), E denotes border coefficient(66475), d is the network diameter(5), C is network clustering coefficient(0.423), K_{max} is the maximum node degree(854), k is average node degree(30.882).

2.1 All nodes interval features

All nodes features refers to the overall behavior features of all nodes in data set. Figure 1 is analysis of group posting behavior characteristic in each board; Figure 2 is analysis of group reply behavior characteristics in each board. It can be found from the figures that, all group behaviors are consistent with power-law distribution, and power exponent can be obtained through simulation fitting. In posting behavioral characteristics analysis,the exponent of message board is the largest, reaching 1.1223, while the exponent of talk board is only 0.816, with exponent of group posting and log board is at the average;however, in analysis of reply behavioral characteristics, the exponent of talk board is the largest, is 1.1041, and the smallest is exponent of log board; so even posting and replying behavioral characteristics are different in the same board.

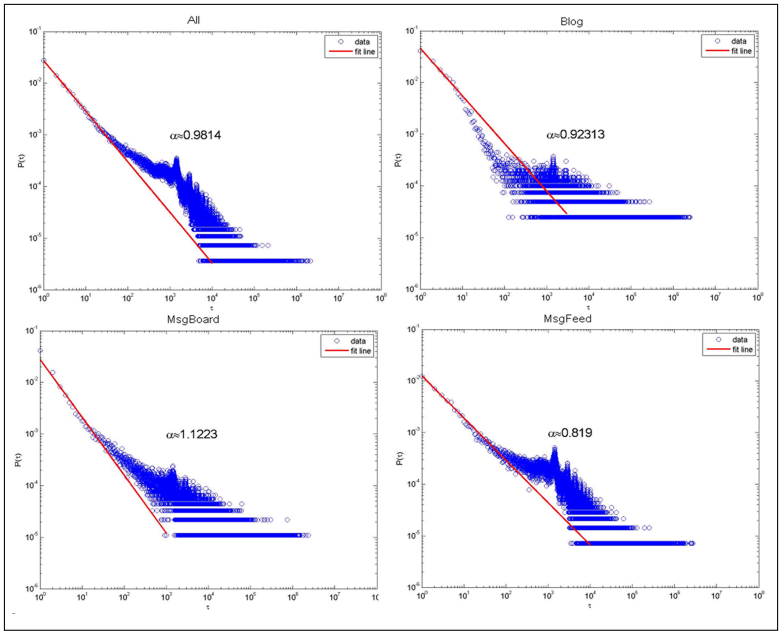


Figure 1: Analysis of group posting behavior characteristic in each board

2.2 Individual node interval features

The paper also analyzes the features of individual behavior, firstly the data sets are sorted in accordance with the number of nodes posting and the number of reply's in descending order, and then select the first rank (denoted by a), three-quarters (denoted as b), intermediate (denoted as c) and fourth (denoted as d) four-node data respectively, post and reply node behavior characteristics were analyzed.

As shown in Figure 3 and 4, the actual post and reply number are at a high level at data set node a and b, so the power index of posting and reply are relatively close, which shows that for active nodes, dealing with things usually by a specific behavior pattern, which is consistent with the literature conclusions; while at data set node c and node d, the actual post number is few but reply of a larger number, so causing power exponent of its post behavior is small, while power exponent of reply behavior is large. To explain this phenomenon, author of this paper used NC algorithm to calculate network binding targets of these four nodes, found that nodes a and b have greater binding, while the binding of node c and d are small, seen from the above exponential distribution of post and reply two types of nodes and the calculation result of NC index value, for node a and b such very active nodes, because of its dominant position in

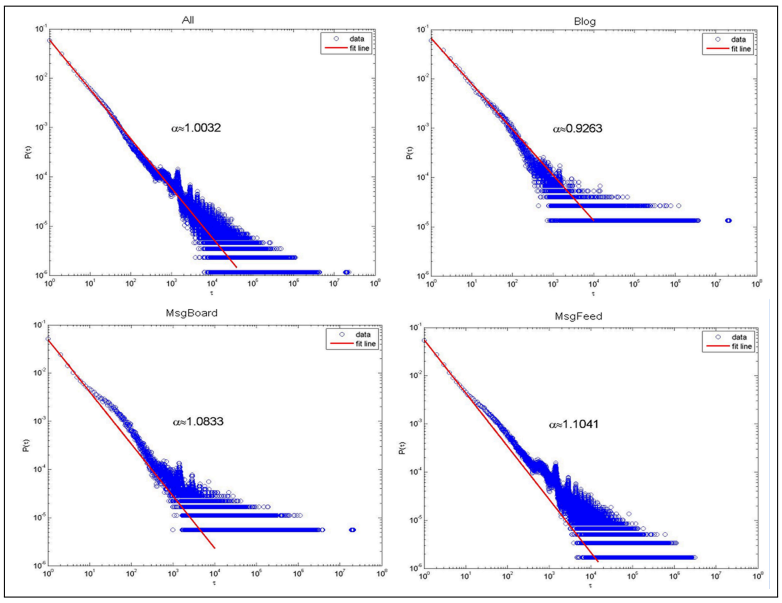


Figure 2: Analysis of group reply behavior characteristics in each board

the message, the posting content has high credibility and timeliness, resulting in follow-up reply interaction increases, while for nodes c and d, as a result of information disadvantage, it can only attract other nodes to forward information by replying behavior.

Therefore, the posting and replying behaviors of active nodes are positively related, while posting and replying behaviors of non-active nodes are negatively related.

2.3 Cluster interval features

According to the above sort results, data sets are divided into 20 equal portions in this paper, which generate 20 clusters (the higher ranking clusters have higher degree of activity), distribution exponent of each cluster posting and reply behavior interval age calculated. In Figure 5 and Figure 6, from left to right are the eighth cluster, the tenth cluster and twelfth cluster posting and reply behavior time interval distribution. It can be found that with the lower degree of cluster activity, power index of the three cluster posting and reply behaviors also decreased, Figure 7 is exponential distribution figure of 20 cluster posting behavior, from curve trend of the figure, we can draw a conclusion that cluster activity degree is positively correlated with power exponent,which is consistent with analysis results of the literature.

2.4 BM phase diagram analysis

When the time interval between incidents follows power-law distribution, means that many events will concentrated occur in a relatively short period of time, followed by a long idle period, this situation is called event paroxysmal feature. From incident time interval distribution, for system with strong paroxysmal feature, most of the time interval will be less than the average event interval, but relatively large time interval may also occur, this phenomenon means that the standard deviation of its time distribution is relatively large. Event paroxysmal feature can be measured by variation coefficient B of time interval, as shown in Formula 1:

$$B \equiv \frac{\frac{\sigma_\tau}{m_\tau} - 1}{\frac{\sigma_\tau}{m_\tau} + 1} = \frac{\sigma_\tau - m_\tau}{\sigma_\tau - m_\tau} \tag{1}$$

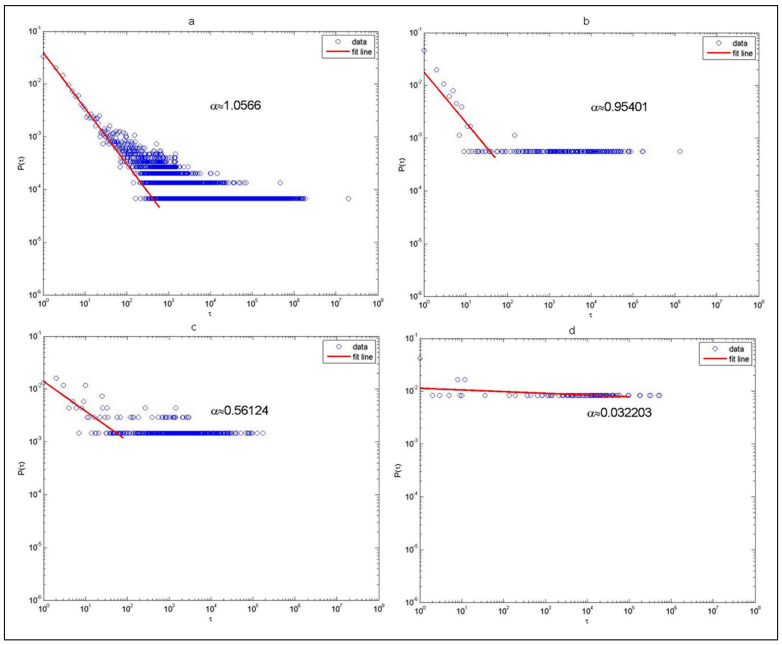


Figure 3: Characteristics of individuals posting behavior

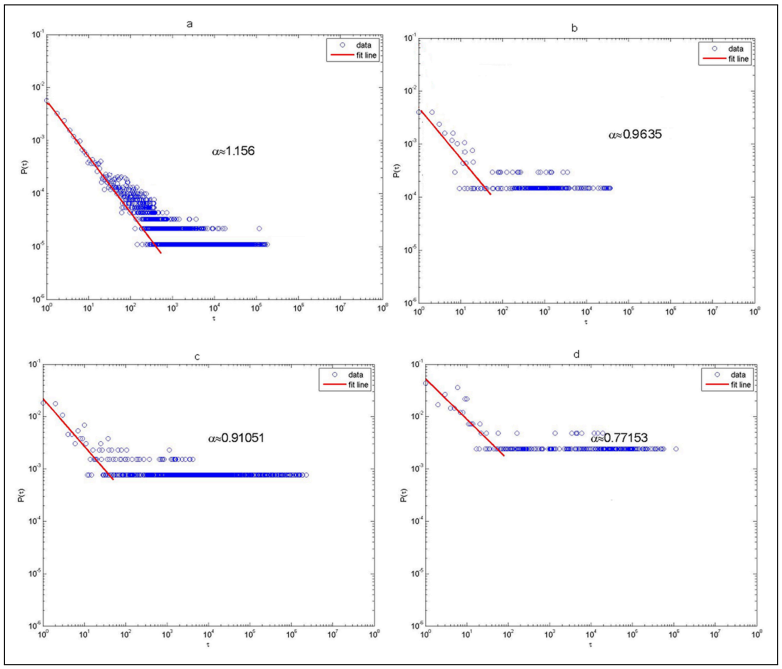


Figure 4: Characteristics of individual reply behavior

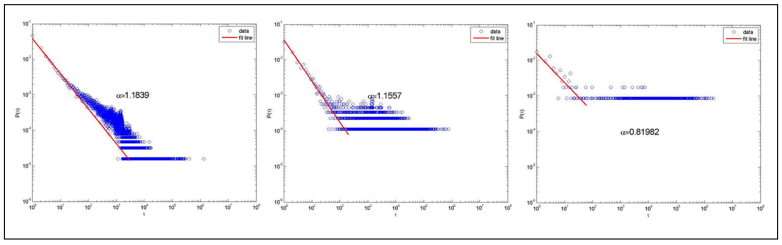


Figure 5: Analysis of three cluster post behavior characteristics

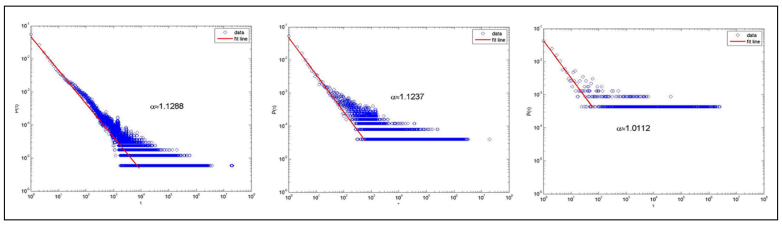


Figure 6: Analysis of three cluster reply behavior characteristics

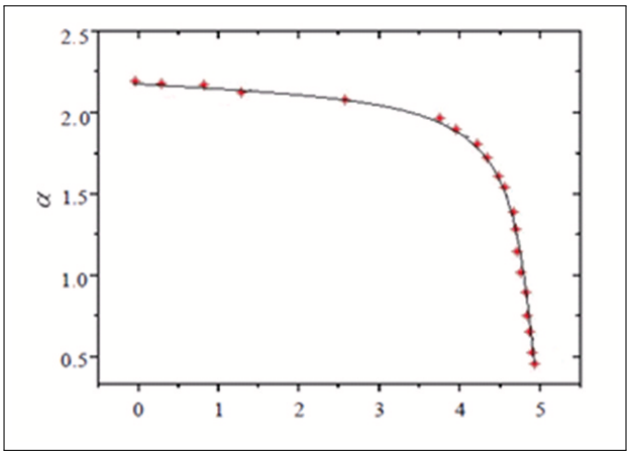


Figure 7: Relation diagram of 20 cluster exponential distribution and degree of activity

Of which σ_τ and m_τ represent standard deviation and average value of p_τ, the value range of B is (-1 to 1), with respect to the Poisson distribution, the average value and standard deviation are equal, the paroxysmal is 0 therefore, can be seen as an equilibrium point set between Goh and Barabasi; for recurring events, the time interval distribution is actually a δ function,the standard deviation is 0, B value is -1. For the power-law distribution, the standard deviation is much larger than average value, B is close to 1, that is, the closer to 1 indicates the stronger paroxysmal, close to 0 indicates a neutral, belonging to random events series, close to -1 indicates no paroxysmal, is cyclical periodic events. In addition to paroxysmal feature of events, events characteristics may also be depicted by memory description: Time sequence of events has a certain memory, a long interval is also followed by a longer time interval, and a short interval is also followed by a shorter time interval, then all of the time intervals form a sequence according to occurring time sequence (time interval sequence of two successive behaviors), assume that this sequence has n_τ elements, i.e., n_τ+1 events occurred,define the previous n_τ-1 elements constitute sequence 1, and define latter n_τ-1 elements constitute sequence two, as shown in formula 2, Pearson correlation of the two sequences can be used to measure the sequence memory.

$$M \equiv \frac{1}{n_\tau - 1} \sum_{i=1}^{n_\tau - 1} \frac{(\tau_i - m_1)(\tau_i - m_2)}{\sigma_1 \sigma_2} \tag{2}$$

m_1 and m_2 are the mean of sequence 1 and sequence2 respectively, σ_1 and σ_2 are standard deviation of sequence 1 and sequence 2. Obviously, the value range of M is also between (-1 to 1): $M > 0$ represents memory effect, $M < 0$ represents anti-memory effect. When M is close to 1, indicates a long (short) time interval is more inclined to corresponding long (short) time interval after another; when closes to 0 indicates a neutral; when closes to -1 indicates a long (short) time interval is more inclined to corresponding short (long) time interval after another. Figure 8 is

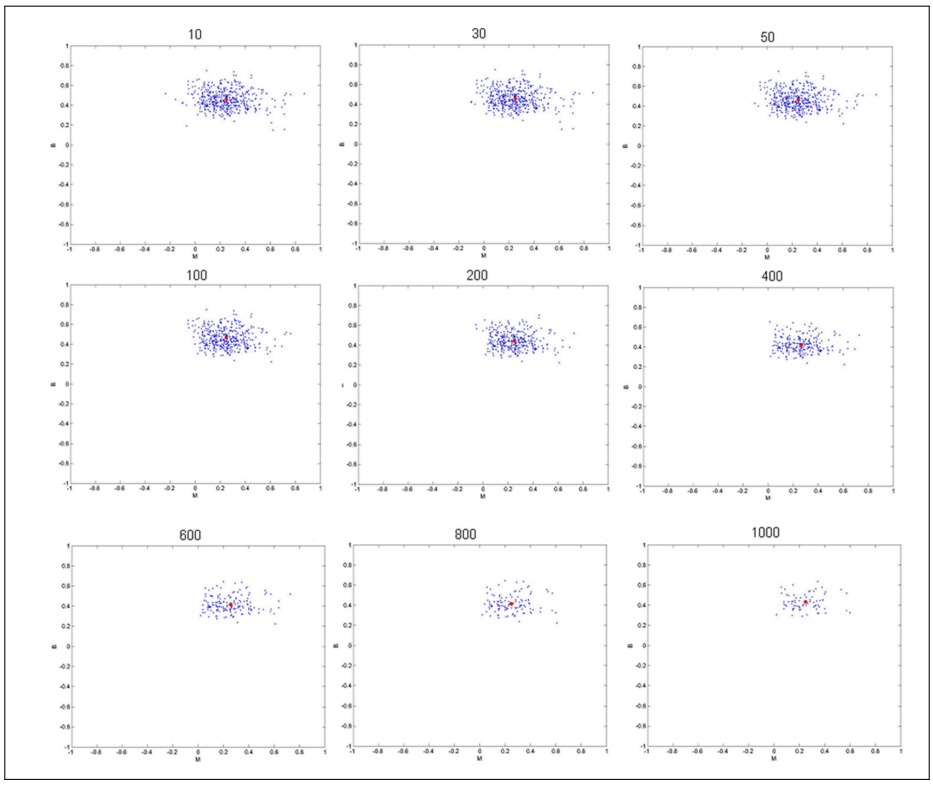


Figure 8: BM phase diagram of posting behavior

BM phase diagram of posting amount exceed 10, 30, 50, 100, 200, 400, 600, 800 and 1000 posts respectively in data set. It can be found from the figure that, although there are great variance in number of posts, their average point location in BM phase diagram are stable between at the horizontal coordinates (0.244-0.26) and vertical coordinates (0.416-0.464), indicating that in QQ space, user posting behavior has obvious paroxysmal and memory features.

3 Model building

In this paper, the information dissemination model involved utilized SIR model of infectious disease dynamics to complete. Define users of social network as nodes, friends relationship between nodes is defined as side, and nodes are divided into three categories: Stifler,Ignorant and Spreaders. Stiflers indicates that the node receives and knows contents of the information, but it does not take the initiative to disseminate this information; Ignorant indicates that the node has not received the information sent by the neighboring nodes, so it will not conduct information dissemination, but it will receive this information at a certain probability; Spreaders indicates that the node has received information from one of its neighboring nodes, at the next time-stepping, the node will take the initiative to disseminate information to its neighboring nodes. For Spreaders, in information dissemination process, if the neighbor node receiving information belongs to Ignorant, then the neighboring node may become Spreaders, and the node will conduct information dissemination at next time; if neighbor node receiving information belongs to Spreaders or Stiflers, i.e.the receiver has received the information, then the information receiving party will abandon to continue to spread this information, and if the neighbor node is Spreaders, it will change its own state into Stiflers. It can thus be seen that for social network node, its transfer between Stiflers, Ignorant and Spreaders states depends not only on the node's state at

the current time-stepping step, also relates to state of its neighbor nodes for information interaction. At the same time this model assumes: Social network structure is relatively static, that is number of nodes, sides relationship and weight between nodes do not change with time; in the social network, node send messages to its neighbor nodes at certain time-stepping,and after neighbor node receiving the information, it will be forwarded at a certain degree of probability;information will only spread among neighbors. To sum up, let the state transition rules as follows:

1. Infection probability λ : After Spreaders sending message to its neighbor nodes, Ignorant will become Spreaders at λ probability, and this information will be disseminated at the next time-stepping;
2. Stiflers probability μ: Spreaders will become Stiflers at μ probability, if and only if Spreaders contact with other neighbors Spreaders or Stiflers;
3. Attenuation probability υ: with time attenuation, Spreaders will no longer take the initiative to disseminate information, so define attenuation probability υ, namely, when Spreaders has no interaction with any neighbor nodes, it will automatically become Stiflers at υ probability.

The probability λ of Ignorant i to believe and actively disseminate information is associated with its neighbor node spread influence, so the spread probability of node i is:

$$\lambda = 1 - \prod_{j \in \phi_i} (1 - \frac{\omega_{ij}}{\sum_{m=1}^{k_i}} \omega_{ij}) \tag{3}$$

Where in ϕ_i, is defined as the set of spread nodes of neighbor nodes to node i, ω_{ij} is defined as the connection weight between node i and node j. According to SIR propagation model, it shows the entire information dissemination process:

$$\begin{aligned} Spreaders(j) + Ignorant(i) &\xrightarrow{\lambda} Spreaders(j) + Spreaders(i) \\ Spreaders(m) + Spreaders(n) &\xrightarrow{\mu} Stiflers(m) + Spreaders(n) \\ Spreaders(m) + Stiflers(k) &\xrightarrow{\mu} Stiflers(m) + Stiflers(k) \\ Spreaders(p) &\xrightarrow{\upsilon} Stiflers(p) \end{aligned} \tag{4}$$

3.1 Propagation model

According to the evolution rule of Formula 3, build mean-field differential evolution equations of information dissemination model. Define S, I and R to represents Spreaders, Ignorant and Stiflers states; for nodes of k degree, define the total number of nodes in Spreaders, Ignorant and Stiflers states as$M_{K,S}, M_{K,I}$ and $M_{K,R}$, the total number of all nodes k degree is M_K, and define:

$$M_K = M_{K,S} + M_{K,I} + M_{K,R} \tag{5}$$

Assuming in social network, the node I is in Ignorant state at t time, in [t,t+$\triangle t$] period of time, define p_{II}^i as the probability of i will remain in Ignorant state, definep_{IS}^i as the probability of i to change from Ignorant state to Spreaders state, and$p_{IS}^i = 1 - p_{II}^i$.If j remains in Ignorant state in $\triangle t$ period of time, indicating that the neighbor node of j in the Spreaders state fail to disseminate information to j. So p_{II}^ican be expressed as:

$$p_{II}^i = \prod_{m=0}^{g} (1 - \triangle t \lambda^i) \tag{6}$$

Where in g=g(t) is defined as the total number of nodes neighboring i at t time, specifically as follows:

$$\prod g, t = \phi(k,t)^g (1-\phi(k,t))^{k-g} \tag{7}$$

Where in, $\phi(k,t)$ is defined as probability of Ignorant of k degree has adjacency relationship with certain Spreaders at t time:

$$\prod(g,t) = \sum_{k\prime} P(K\prime \mid k) P(S_{k\prime} \mid I_k) \approx \sum_{k\prime} P(K\prime \mid k) \rho^S(k\prime, t) \tag{8}$$

In formula 8, $P(K\prime \mid k)$is the correlation function of degree, that is conditional probability of node of k degree and node of $K\prime$ degree have adjacency relationship; $P(S_{k\prime} \mid I_k)$ is defined when node of $K\prime$ degree has connection relationship with uninfected node of k degree, the node belongs probability of propagation state; $\rho^s(k\prime, t)$is defined as Spreaders density at t time and $K\prime$ degree. By traversing all possible g and i, the mean of p_{II}^i can be calculated, i.e. $\overline{p}_{II}(k,t)$, which is used to describe the mean field dynamics equation of the model:

$$\overline{p}_{II}(k,t) = \frac{1}{M_k} \sum_i \sum_{g=0}^{k} \prod_{m=0}^{g} (1 - \triangle t \lambda^{mi}) \phi(k,t)^g [1-\phi(k,t)]^{k-g} \tag{9}$$

Where in, M_k is the total number of all nodes of k degree in the network. In [t,t+$\triangle t$] period of time, p_{SS}^i is defined as the probability of Spreaders to maintain spread state, traverse all possible g, the average probability $\overline{p}_{II}(k,t)$ for node to maintain spread state:

$$\begin{aligned}
\overline{p}_{II}(k,t) &= \sum_{g=0}^{k} (1-\mu \triangle t)^g \phi(k,t)^g (1-\phi(k,t))^{k-g} (1-\upsilon \triangle t) \\
&= \sum_{g=0}^{k} (1-\mu \triangle t) \phi(k,t)^g (1-\phi(k,t))^{k-g} (1-\upsilon \triangle t) \\
&= (1-\mu \triangle t)\phi(k,t) + 1 - \phi(k,t)^k (1-\upsilon \triangle t) \\
&= (1-\mu \triangle t \phi(k,t)^k)(1-\upsilon \triangle t) \\
&= (1-\mu \triangle t \sum_{k\prime} P(K\prime \mid k)[\rho^S(k\prime,t) + \rho^R s(k\prime,t)])^k (1-\upsilon \triangle t)
\end{aligned} \tag{10}$$

The probability for Spreaders to become Stiflers (contact) can be expressed as $\overline{p}_{SR}(k,t) = 1 - \overline{p}_{SS}(k,t)$. On the basis of average state transition probability, according to node change rules of formula 5, changing situation of nodes of k degree in three states in[t,t+$\triangle t$]time period can be obtained, as shown in Equation 11, 12 and Equation 13:

$$\begin{aligned}
M_{K,I}(t+\triangle t) &= M_{K,I}(t) - M_{K,I}(t)(1-\overline{p}_{II}(k,t)) \\
&= M_{K,I}(t) - M_{K,I}(t)[1 - \frac{1}{M_k} \sum_i \sum_{g=0}^{k} \prod_{m=0}^{g} \\
&\quad (1-\triangle t \lambda^{mi}) \phi(k,t)^g [1-\phi(k,t)]^{k-g}]
\end{aligned} \tag{11}$$

$$\begin{aligned} M_{K,I}(t+\triangle t) &= M_{K,S}(t) + M_{K,I}(t)(1-\overline{p}_{II}(k,t)) - M_{K,S}(t)(1-\overline{p}_{SS}(k,t)) \\ &= M_{K,S}(t) + M_{K,I}(t)[1 - \frac{1}{N_k}\sum_i \sum_{g=0}^{k} \prod_{m=0}^{g}(1-\triangle t\lambda^{mi}) \\ &\phi(k,t)^g[1-\phi(k,t)]^{k-g}] - M_{K,S}(t)[1-(1-\mu \triangle t \sum_{k\prime} P(K\prime \mid k) \\ &[\rho^S(k\prime,t) + \rho^R s(k\prime,t)])^k(1-\upsilon \triangle t)] \end{aligned} \tag{12}$$

$$\begin{aligned} M_{K,R}(t+\triangle t) &= M_{K,R}(t) + M_{K,S}(t)(1-\overline{p}_{SS}(k,t)) \\ &= M_{K,R}(t) + M_{K,S}(t)[1-(1-\mu \triangle t \sum_{k\prime} P(K\prime \mid k) \\ &[\rho^S(k\prime,t) + \rho^R s(k\prime,t)])^k(1-\upsilon \triangle t)] \end{aligned} \tag{13}$$

To simplify the calculations, denote$\Phi(k,g,t) = \phi(k,t)^g(1-\phi(k,t))^{k-g}$, For Formula 11, the following variants can be realized:

$$\frac{M_{K,I}(t+\triangle t) - M_{K,R}(t)}{M_K} = \frac{M_{K,I}(t)}{M_K}[1 - \frac{1}{M_k}\sum_i \sum_{g=0}^{k} \prod_{m=0}^{g}(1-\triangle t\lambda^{mi})\Phi(g,k,t)] \tag{14}$$

Denote $\rho^I(k,t) = \frac{M_{K,I}(t)}{M_K}$. Both ends of Formula(14)take$\triangle t \to 0$.
Since $lim_{\triangle t \to 0}\prod_{m=0}^{g}(1-\triangle t\lambda^{mi}) = \sum_{m=0}^{g}(-\lambda^{mi})$, so Formula (14) can be defined as:

$$\frac{\varphi\rho^I(k,t)}{\varphi(t)} = \frac{\rho^I(k,t)}{M_K}\sum_j \sum_{m=1}^{g}\sum_{g=1}^{k}\Phi(g,k,t)\lambda^{mi} \tag{15}$$

Similarly, it can be derived from Formula (13)

$$\begin{aligned} \frac{M_{K,R}(t+\triangle t) - M_{K,R}(K,t)}{N_K} &= \frac{M_{K,S}(K,t)}{N_K}(1-\overline{p}_{SS}(K,t)) \\ &= \frac{M_{K,S}(K,t)}{m_K}[1-(1-\mu \triangle t \sum_{k\prime} P(K\prime \mid k) \\ &[\rho^S(k\prime,t) + \rho^R s(k\prime,t)])^k(1-\upsilon \triangle t)] \end{aligned} \tag{16}$$

At right end of Formula(16)

$$\begin{aligned} &(1-\mu \triangle t \sum_{k\prime} P(K\prime \mid k)[\rho^S(k\prime,t) + \rho^R s(k\prime,t)])^k \\ &= \sum_{k=0} nC(k,n)(-\mu \triangle t \sum_{k\prime} P(K\prime \mid k[\rho^S(k\prime,t) + \rho^R s(k\prime,t)])^n \\ &= C_0^K + C_1^K(-\mu \triangle t \sum_{k\prime} P(K\prime \mid k[\rho^S(k\prime,t) + \rho^R s(k\prime,t)]) \\ &= 1 - k\mu \triangle t \sum_{k\prime} P(K\prime \mid k[\rho^S(k\prime,t) + \rho^R s(k\prime,t)] \end{aligned} \tag{17}$$

Therefore, Formula(16) is varied:

$$\begin{aligned}
&\frac{M_{K,R}(t+\triangle t)-M_{K,R}(K,t)}{N_K}\\
&=\rho^S(k,t)[1-(1-k\mu\triangle t\sum_{k\prime}\\
&P(K\prime\mid k)[\rho^S(k\prime,t)+\rho^R s(k\prime,t)])(1-\upsilon\triangle t)]\\
&=\rho^S(K,t)(k\mu\triangle t\sum_{k\prime}P(K\prime\mid k)+\upsilon\triangle t+k\mu\triangle t^2\sum_{k\prime}\\
&P(K\prime\mid k))[\rho^S(k\prime,t)+\rho^R(k\prime,t)]
\end{aligned} \tag{18}$$

Both ends of Formula (18) divide $\triangle t$ and take the limit $\triangle t\to 0$.

$$\frac{\varphi\rho^R(k,t)}{\varphi(t)}=k\mu\rho^S(k,t)\sum_{k\prime}[\rho^S(k\prime,t)+\rho^R(k\prime,t)]P(k\prime\mid k)+\upsilon\rho^S(k,t) \tag{19}$$

Since $\frac{\varphi\rho^I(\alpha,k,t)}{\varphi(t)}+\frac{\varphi\rho^S(\alpha,k,t)}{\varphi(t)}+\frac{\varphi\rho^R(\alpha,k,t)}{\varphi(t)}=0$,so it is easy to obtain:

$$\begin{aligned}
\frac{\varphi\rho^S(k,t)}{\varphi(t)}&=\frac{\rho^I(k,t)}{M_k}\sum_j\sum_{m=1}^{g}\sum_{g=1}^{k}\Phi(g,k,t)\lambda^{mi}\\
&-k\mu\rho^S(k,t)\sum_{k\prime}[\rho^S(k\prime,t)+\rho^R(k\prime,t)]P(k\prime\mid k)-\upsilon\rho^S(k,t)
\end{aligned} \tag{20}$$

By Formula (15), (19) and (20) simultaneous,obtain the dynamical evolution equations of information dissemination in social network for depicting changes in the relationship between Spreaders,Ignorant and Stiflers density over time.

4 Simulation analysis

Figure 9 is the propagation and evolution diagram of Stiflers, Ignorant and Spreaders density under different power exponent α and even time-stepping($\alpha=0$), and from left to right are Spreads density evolution diagram, Ignorant evolution diagram and Stiflers evolution diagram. It can be seen from Spreads density evolution diagram that, with the decrease of power exponent α, peak of wave of Spreads density continues to decline, but the final information dissemination duration has been prolonged, indicating that the smaller heterogeneity of temporal characteristics of user behaviors, breadth of information dissemination may be affected, but the final information dissemination duration will be extended. It can be seen from Ignorant evolution diagram and Stiflers evolution diagram that, for the former, with the decrease of power exponent α, the greater proportion of Ignorant; while for the latter, with the decrease of power exponent α, the smaller proportion of Stiflers.

Figure 10 is respectively represent evolution diagram of the proportion of new Spreaders number per time step and proportion of cumulative Spreaders number, which also confirmed from edge-wise that the power-law characteristic of user activity time distribution will have a huge impact on information dissemination. When the power exponent is small, although there are many new Spreaders, the information dissemination comes and goes fast, which may not have a greater impact; but when the power exponent is large, although the number of people implementing information dissemination is small, it lasts longer, it will produce more lasting influence, which explains the phenomenon why some information gets spread again after a long silence.

Relaxation time refers to the time required for model to start from evolution to tended to be

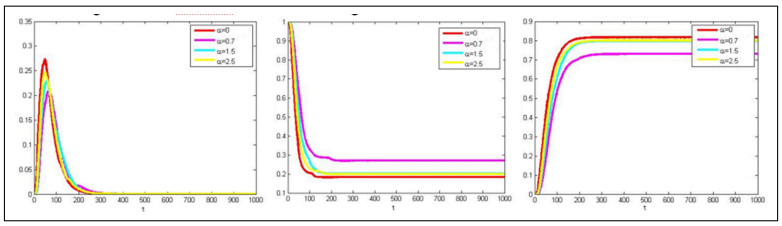


Figure 9: Influence of time-order character on propagation

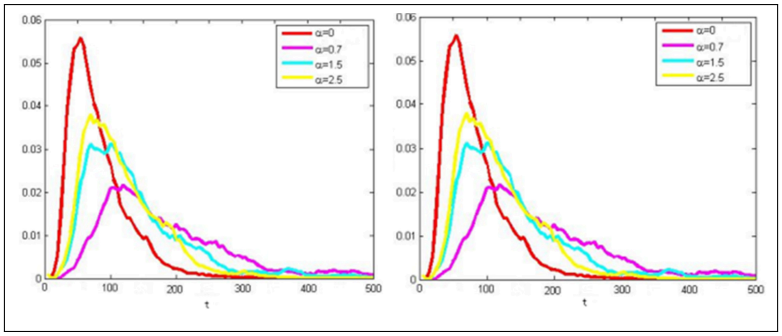


Figure 10: Left:Evolution diagram of proportion of new Spreaders number per time step. Right: Evolution diagram of proportion of cumulative Spreaders number

stable, Figure 11 are the relationship between power exponent and the relaxation time and respectively represent when the number of Spreads goes over half of time t, It can be seen from the figure that, as the power exponent increases, the relaxation time shows obvious linear downward trend. Since the power-law distribution of user behavior can be expressed in group and individual nodes, so the paper conducted simulation analysis of impact of power-law characteristic at groups and individual-level on information dissemination process. The so-called power-law distribution at group level refers to the behavior time distribution sequence of each node is regular, namely the time interval of node behavior remains unchanged,but the degree of activity between nodes varies greatly, and meets power-law distribution. Power-law distribution at individual-level refers to the time interval distribution of individual's own behavior meets the power-law characteristic, but the degree of activity and time intervals satisfied different distribution between individuals are the same.

The center and right respectively represent when the number of Spreads goes over half of time t, impact of power-law distribution characteristics group and individual level on speed of information dissemination relationship diagram. It can be seen from the above figures that, at the group level, the power-law distribution has greater impact on the dissemination of information, while power-law distribution at individual level has less effect on the dissemination of information.

Figure 12 respectively represent relation diagram between the maximum propagation range of information dissemination C_{max} and the maximum node propagation density S_{max} and power exponent, as it can be seen from the figure,with continued exponential increases, the maximum propagation range of information dissemination C_{max} and S_{max} the maximum node propagation density simultaneously increase, but when the power index $\alpha > 1.5$, the maximum propagation range of information dissemination has been stabilized, while it will still have some impact on the maximum node propagation density S_{max}.

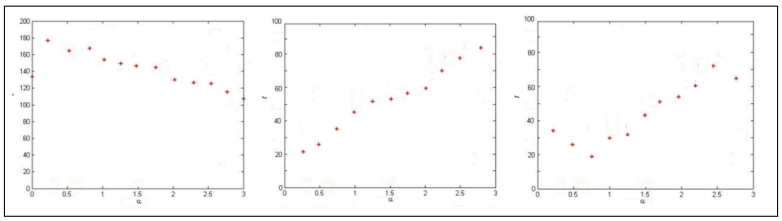


Figure 11: Left:Relationship between power exponent α and relaxation time. Center:Impactof power-law distribution characteristics group-level on speed of information dissemination. Right: Impact of power-law distribution characteristics individual-level on speed of information dissemination

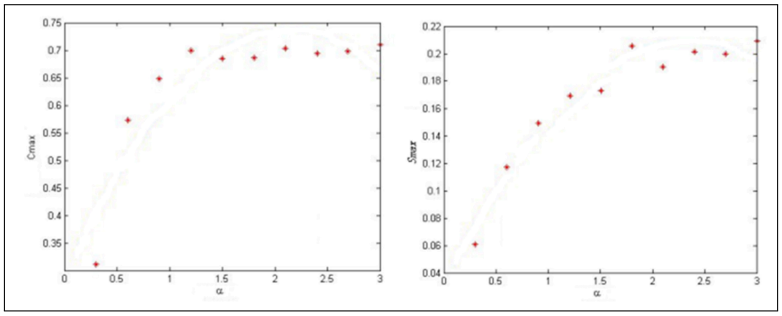


Figure 12: Left:Relation diagram between the maximum propagation range of information dissemination and power exponent. Right: Relation diagram between the maximum node propagation density and power exponent

5 Conclusion

With empirical data of social networks, here conduct quantitative analysis of user behavior time interval characteristics, which is conducive to explain many complex networks information propagation phenomena, and can generate social benefits and application value in the public opinion monitoring, disease prevention, information recommendation and other aspects. Firstly, the paper makes use of real social network user behavior data, to analyze information posting and reply behaviors of networking groups, network individuals and network groups respectively, then use the BM phase diagram to analyze paroxysmal and memory characteristics of user's information posting and reply behaviors. By means of SIR propagation model and empirical data, implement quantitative study of impact of user behavior time interval characteristic on information dissemination process, and found that user behavior time interval meets the characteristics of power-law distribution, although it will slow speed of information dissemination to a great extent, it will also extend the duration of the information dissemination and can increase the ultimate scale of information dissemination, and thus more likely to have a greater impact on the society; It also found that at the group level, power-law characteristic has a greater impact on speed of information dissemination, while at individual level, the speed of information dissemination is less affected by power-law characteristic.

Acknowledgment

This research is supported by:
1. National Natural Science Foundation of China (No.61271308);
2. Beijing Natural Science Foundation (No.4112045);

3. Beijing City Science and Technology Project (No.Z121100000312024);
4. Specialized Research Fund for the Doctoral Program of Higher Education of China (No. W11C100030).

Bibliography

[1] Zhou, T.; Han, X.P.et al(2013); Statistical Mechanics on Temporal and Spatial Activities of Human, *Journal of University of Electronic Science and Technology of China*, ISSN 1674-862X, 4(4):481-540.

[2] Zhang, H.P.(2015); An agent-based simulation model for supply chain collaborative technological innovation diffusion, *International Journal of Simulation Modelling*, ISSN 1726-4529, 14(2):313-324.

[3] Liu, S.; Gong,D.(2014); Modelling and simulation on recycling of electric vehicle batteries-using agent approach, *International Journal of Simulation Modelling*, ISSN 1726-4529, 13(1):79-92.

[4] Pasztor, A.(2014); Gathering simulation of real robot swarm, *Technical Gazette*, ISSN 1848-6339, 21(5):1073-1080.

[5] Shang, Y.l.(2013); Measuring degree-dependent failure in scale-free networks of bipartite structure, *International Journal of Simulation & Process Modelling*, ISSN 1740-2131, 8(1):74-78.

[6] Lerher, T.; Ekren, Y.B.; Sari,Z.;Rosi,B.(2015); Simulation Analysis of Shuttle Based Storage and Retrieval Systems, *International Journal of Simulation Modelling*, ISSN 1726-4529, 14(1):48-59.

[7] Cho, Y.C.(2015); A novel approach of adaptive socially aware routing algorithm in delay tolerant networks, *Technical Gazette*, ISSN 1848-6339, 22(1):61-70.

[8] Xue, Y.G.et al(2014); Determination of statistical homogeneity by comprehensively considering the discontinuity information, *Technical Gazette*, ISSN 1848-6339, 21(5),971-977.

[9] Java, A.; Song, X.; Finin, T.; Tseng,B.(2007); WebKDD/SNAKDD 2007:web mining and social network analysis post-workshop report, *Acm Sigkdd Explorations Newsletter*, 9(2):87-92.

[10] Kwak, H.; Lee, C.; Park, H.(2010); What is Twitter,a Social Network or a News Media,*International conference on World wide web*,591-600.

[11] Cha, M.; Haddadi, H.et al(2010); Measuring user influence in Twitter: the million follower fallacy, *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, 23-26.

[12] Suh, B.; Hong, L.; Pirolli, P.; Chi, E.H.(2010); Want to be Retweeted? Large Scale Analytics on Factors Impacting Retweet in Twitter Network, *2010 IEEE Second International Conference on Social Computing*, 177-184.

[13] Han, D.D.et al(2008); Fluctuation of the Download Network, *Chinese Physics Letters*, ISSN 0256-307X, 25(2):765-768.

[14] Fu, F.; Liu, L.H.; Wang, L.(2008); Empirical analysis of online social networks in the age of Web 2.0, *Physica A*, ISSN 0378-4371, 387(2):675-684.

[15] Wang, Z. et al(2015); Coupled disease-behavior dynamics on complex networks: A review, *Physics of Life Reviews*, ISSN 1571-0645, 15(1):30-31.

[16] Alessandro, A.; Laura, B.; George, L.(2015); Privacy and human behavior in the age of information, *Science*, 347(6221):509-14.

[17] Freitas, C.R.D.(2015); Weather and place-based human behavior: recreational preferences and sensitivity, *International Journal of Biometeorology*, ISSN 0020-7128, 59(1):55-63.

[18] Medina, J.R.; Lorenz,T.; Hirche, S.(2015); Synthesizing Anticipatory Haptic Assistance Considering Human Behavior Uncertainty, *Robotics IEEE Transactions on*, 31(1):180-190.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):224-232, April 2016.

A Taboo Search Optimization of the Control Law of Nonlinear Systems with Bounded Uncertainties

A. Gharbi, M. Benrejeb, P. Borne

Amira Gharbi*, Mohamed Benrejeb, Pierre Borne
LARA, Ecole Nationale d'Ingénieurs de Tunis
Tunisie, BP 37, Le Belvédére 1002 Tunis
and
CRIStAL, Ecole Centrale de Lille
France, Cité scientifique BP 48-59651, Villeneuve d'Ascq Cedex
merkarim@gmail.com, mohamed.benrejeb@enit.rnu.tn, pierre.borne@ec-lille.fr
*Corresponding author: merkarim@gmail.com

Abstract: The aim of this paper is to propose a method to determine among the eligible controls of a nonlinear system, with bounded perturbations, the one which minimizes the final error. The approach is based on the implementation of aggregation techniques using vector norms in order to determine a comparison system used to calculate an attractor in view of its minimization by implementation of metaheuristics.
Keywords: Attractor, aggregation technique, vector norm, optimization, Taboo search.

1 Introduction

In the presence of uncertainties in modeling, that increase the complexity of the stability study [1], it is not always possible to obtain a control law ensuring the stability of the process with respect to a chosen objective. It is then necessary to estimate the maximum deviation from this target, an operation which can be performed by determining an attractor [2]- [4] corresponding to the vicinity of the target for which the local stability cannot be guaranteed, [5], [7], [6], [8], [9], [10], [11], [12]. In case of uncertain or poorly defined problems, possibly subject to random perturbations or for which the search for solutions might evolve towards the combinatorial explosion, the exact methods are very unlikely to provide solutions in an acceptable period of time. The method presented in this paper corresponds to a law finding, if we do not obtain the optimal solution of the problem, we obtain at least a good solution in an acceptable run time. The heuristic methods that can be implemented on a computer are referred to metaheuristics. They rely on the following basic principle: the search for optimum is simulating either the behaviour of a biologic system or the evolution of a natural phenomenon, including an intrinsic optimization mechanism. For this reason, a new optimization branch has been developed in the past 20 years, inspired by nature. Almost all numerical algorithms designed as metaheuristics are included into this class of optimization techniques [13]. In general, all metaheuristics are using a pseudo-random engine to select some parameters or operations that yield to the estimation of an optimal solution. The procedures to generate pseudo-random (numerical) sequences of optimization are crucial in metaheuristics design. We have two classes of metaheuristic approaches: global approaches and local approaches, such as the Taboo search which is one of the easiest to implement. In this paper, the determination of the attractor, when the process is submitted to uncertainties, is achieved by using aggregation techniques and the Borne-Gentina stability criteria, with the use of vector norms and of comparison systems [14], [15]. In the following section 2, we propose the determination of the control law of a nonlinear process submitted to bounded uncertainties with a view to minimize the effect of these uncertainties. In section 3 we use the taboo search to realize the optimization. An application is presented in section 4 to illustrate the proposed method.

2 Attractor determination

Let us consider the system (S) whose evolution is described by the following state equation

$$\dot{x} = f(x,.) + g(x,.)u + \delta(.) \tag{1}$$

$$y = h(x) \tag{2}$$

x is the state vector and y is the output, $x \in R^n, y \in R^m, u \in R^l$
$\delta \in R^n$ characterizes the disturbances and/or perturbations acting on the system and u is the control law:

$$u = u(x,\theta) \tag{3}$$

where $\theta \in R^\nu$ is a vector of the adjustable parameters of the control law. A new representation of system (S) characterized by (1) and (3) can be defined by

$$\dot{x} = A(x,\theta,.)x + \delta(.) \tag{4}$$

with

$$|\delta(.)| \leq \delta_M \tag{5}$$

$$A = f(x,.) + g(x,.)u(x,\theta) \tag{6}$$

and a comparison system of this system can be determined using the vector norm $p(x)$ defined by

$$p(x) = [|x_1|, |x_2|, \ldots, |x_n|]^T \tag{7}$$

By noting $M(A(x,\theta,.))$ an overvaluing matrix of $A(x,\theta,.)$ related to the vector norm $p(x)$ it comes

$$\frac{d}{dt}p(x) \leq M(A(x,\theta,.))p(x) + N(.) \tag{8}$$

Let us denote:

$$A(.) = \{a_{ij}(.)\} \tag{9}$$

and $M(\theta) = \{m_{ij}(\theta)\}$ the matrix such that:

$$\begin{cases} m_{ii}(\theta) = \max a_{ii}(x,\theta,.) & \forall i = 1,2,\ldots n \\ m_{ij}(\theta) = \max |a_{ij}(x,\theta,.)| & \forall i \neq j \end{cases} \tag{10}$$

We can define a comparison system by:

$$z \in {}^n/\dot{z}(t) = M(\theta)z(t) + \delta_M \tag{11}$$

If $M(\theta)$ is the opposite of an M-matrix, it exists an attractor D_θ asymptotically stable such that

$$D_\theta = \left\{x \in R^n; p(x) \leq -M^{-1}(\theta)\delta_M = p_M(\theta)\right\} \tag{12}$$

3 Taboo search optimization

3.1 Principe of Taboo search

The metaheuristic described in this section belongs to greedy descent local methods. For this type of methods, the search starts from an admissible solution θ_i of $\boldsymbol{S}$. The strategy is then to focus on a local vicinity $V(\theta_i)$, in order to find another solution θ_j that can improve the criterion current performance. The vicinity $V(\theta_i)$ corresponds to the set of all accessible solutions after applying a single admissible movement, displacement or transition from θ_i . Usually, this set is a hyper-cube or a hyper-sphere including the current solution θ_i.

3.2 Taboo search method

Based on the principle of local search for minimizing a criterion, by this method, there is the possibility to jump out from the capturing vicinity and to explore a different zone of the research area. Here after, the term of movement stands for any modification allowing the search to focus on vicinity in the neighborhood of the current vicinity. As usual, the search starts from some initial point (solution), θ_i in the vicinity $V(\theta_i)$ but it is permitted to relocate the exploitation around another point (solution) $\theta_j \in V(\theta_i)$, even if this choice degrades the criterion to optimize. This actually is a movement towards another zone of interest. However, in order to avoid infinite search loops, once a solution is focused on, it will never be focused on again in the future iterations. Thus, the N_T last focused solutions belonging to a Taboo list T_{ki} become untouchable, "taboo" [16], [17]. Starting from the solution θ_i, a set of possible movements, say $M_{k,j}$, can be built, during the k- th iteration. Let $\delta\theta \in M_{k,j}$ be such a movement. By convention,$\theta_i \xrightarrow{\delta\theta} \theta_j$ stands for the transition from solution θ_i to a new point θ_j as result of movement $\delta\theta$. Then

$$V_k(\theta_i) = \left\{\theta_j \in V(\theta_i)/\exists\ \delta\theta \in M_{kj}, \theta_i \xrightarrow{\delta\theta} \theta_j\ \&\ \theta_j \notin T_{ki}\right\} \tag{13}$$

The new solution which is the best non taboo one is added to the last taboo list and the oldest one is removed from it. The chosen criterion is for this problem the minimisation of a scalar norm of $p_M(\theta)$ The optimization of the control law consists to determine the value of θ minimizing a scalar norm of p_M. In the following we use the Euclidian norm $\|p_M\|$. The optimisation algorithm corresponds in this paper to the taboo search with N_T number of elements of the taboo list and N_S the maximum number of iterations without improvement of the solution to stop the research.

4 Application to a second order system

Let us consider the nonlinear system of second order with uncertainties such that

$$\dot{x} = A(x,t)x + B(x)u(x,\theta) + \delta(.) \tag{14}$$

$$y = h(x) \tag{15}$$

with

$$u(\theta, x) = -(\theta_1 y + \theta_2 x_2) \tag{16}$$

and

$$h(x) = x_1 + (1 - e^{-x_1^2})x_2 \tag{17}$$

with $x\,(t) \in R^2, B(.) \in R^2, A(.)$ a 2×2 matrix and $\theta \in R^2$ such that

$$A(x,t) = \begin{bmatrix} -2 + \cos t + \cos x_1 & 4 - e^{-x_1^2}(1 + \sin x_1) \\ 4 + \cos x_2 & -8 + \sin x_1 + e^{-x_1^2} \end{bmatrix} \tag{18}$$

$$B(x) = \begin{bmatrix} 3 + 0.5\cos x_1 \\ 2 \end{bmatrix} \tag{19}$$

we can write (14) as

$$\dot{x} = \mathbf{A}(x,t,\theta)x + \delta(.) \tag{20}$$

with

$$\mathbf{A}(x,\mathrm{t},\theta) = A(x,t) - B(x)[\theta_1, \theta_1((1 - e^{-x_1^2})) + \theta_2] \tag{21}$$

it comes

$$\mathbf{A}(x,t,\theta) = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{22}$$

with

$$\begin{aligned} a_{11} &= -2 + \cos t \cos x_1 - \theta_1(3 + 0.5\cos x_1) \\ a_{12} &= 4 - e^{x_1^2}(1 + \sin x_1) - (3 + 0.5\cos x_1)(\theta_1(1 - e^{x_1^2}) + \theta_2) \\ a_{21} &= 4 + \cos x_2 - 2\theta_1 \\ a_{22} &= -8 + e^{x_1^2} + \sin x_1 - 2[\theta_1(1 - e^{x_1^2}) + \theta_2] \end{aligned} \tag{23}$$

4.1 Determination of a comparison system

For the vector norm $p(x) = [|x_1|, |x_2|]^T$, we obtain the overvaluation defined by

$$\frac{d}{dt}p(x) \leq M(\mathbf{A}(x,\theta,.))p(x) + N(.) \tag{24}$$

$$z \in R^n / \dot{z}(t) = M(.)z(t) + N(.) \tag{25}$$

with

$$M(\mathbf{A}(x(t))) = \begin{bmatrix} a_{11} & |a_{12}| \\ |a_{21}| & a_{21} \end{bmatrix} \tag{26}$$

and

$$|N(.)| \leq \delta_M \tag{27}$$

In our example $\delta(.)$ is assumed to be by bounded by

$$\delta_1 = \begin{bmatrix} -0.2 \\ 0.3 \end{bmatrix} \leq \delta(.) \leq \delta_2 = \begin{bmatrix} 0.1 \\ 0.5 \end{bmatrix} \tag{28}$$

then

$$\delta_M = \begin{bmatrix} 0.2 \\ 0.5 \end{bmatrix} \tag{29}$$

and by overvaluation, for the process without feedback, for $\theta = (\theta_1, \theta_2) = (0, 0)$ we obtain the linear comparison system $\dot{z} = Mz + N$

$$\dot{z} = \begin{bmatrix} 0 & 2 \\ 5 & -6 \end{bmatrix} z + \begin{bmatrix} 0.2 \\ 0.5 \end{bmatrix} \tag{30}$$

after application of stability conditions we have

$$\det(M) < 0 \tag{31}$$

it appears that M is not stable and so is not the opposite of an M-matrix which needs the determination of a suitable feedback optimized in order to limit the influence of the uncertainties.

4.2 Attractor optimization with taboo search

For this taboo search we choose $N_T = N_S = 4$.
Starting from the solution $\theta_1 = 2$ and $\theta_2 = 0$ a set possible movements, can be built, during the k-th iteration. Let $\delta\theta_l \in M_{k,j}$ be such a movement, with $|\delta\theta_1| = 0.2$ and $|\delta\theta_2| = 0.1$. By convention, $\theta_{i1} \overset{\delta\theta_1}{\rightarrow} \theta_{j1}$, $\theta_{i2} \overset{\delta\theta_2}{\rightarrow} \theta_{j2}$ stands for the transition from solution θ_{li} to a new point θ_{lj} with $l = \{1, 2\}$ as result of movement $\delta\theta_l$. Then for $\theta_1 = 2$ and $\theta_2 = 0$ the overvaluing system for the vector norm $p(x) = [|x_1|, |x_2|]^T$ is defined by (22) with

$$M(x,t,2,0) = \left[\begin{array}{c|c} -8 + \cos t & -2 - e^{-x_1^2}(-5 + \sin x_1 - \cos x_1) \\ & -\cos x_1 \\ \hline |\cos x_2| & -12 + \sin x_1 + 5e^{-x_1^2} \end{array}\right] \tag{32}$$

and

$$N = \left[\begin{array}{c} 0.2 \\ 0.5 \end{array}\right] \tag{33}$$

then the linear comparison system is the following

$$\dot{z} = \left[\begin{array}{cc} -7 & 3 \\ 1 & -7 \end{array}\right] z + \left[\begin{array}{c} 0.2 \\ 0.5 \end{array}\right] \tag{34}$$

The stability conditions for matrix M can be written

$$\left\{\begin{array}{l} -7 < 0 \\ \left(-1^2\right) \det(M) > 0 \end{array}\right. \tag{35}$$

as M is the opposite of M-matrix, we have

$$p(x) \leq -M^{-1}N = \left[\begin{array}{c} 0.0630 \\ 0.0804 \end{array}\right] = p_M(2,0) \tag{36}$$

The strategy is then to focus on a local vicinity $V(\theta_i)$ in order to find the best non taboo solution θ_i the chosen criterion being the Euclidian norm of $p_M(\theta)$.
For this, eight solutions ★ will be tested starting from $\theta = (2, 0)$

$$\theta = (\theta_1, \theta_2 + \delta\theta_2), \theta = (\theta_1, \theta_2 - \delta\theta_2), \theta = (\theta_1 + \delta\theta_1, \theta_2), \theta = (\theta_1 - \delta\theta_1, \theta_2), \theta = (\theta_1 + \delta\theta_1, \theta_2 + \delta\theta_2),$$
$$\theta = (\theta_1 + \delta\theta_1, \theta_2 - \delta\theta_2), \theta = (\theta_1 - \delta\theta_1, \theta_2 + \delta\theta_2), \theta = (\theta_1 - \delta\theta_1, \theta_2 - \delta\theta_2)$$

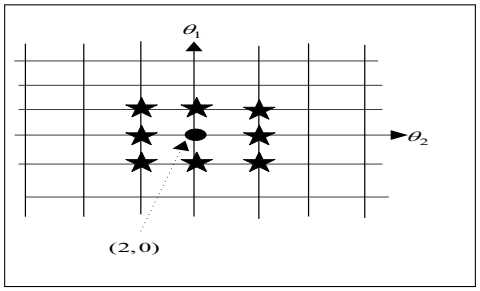


Figure 1: The vicinity of $\theta = (2, 0)$ solution

for

$$\theta = (2, 0.1) \Rightarrow p(x) \leq p_M(2, 0.1) = \begin{bmatrix} 0.0579 \\ 0.0775 \end{bmatrix},$$
$$\theta = (2, -0.1) \Rightarrow p(x) \leq p_M(2, -0.1) = \begin{bmatrix} 0.0668 \\ 0.0787 \end{bmatrix},$$
$$\theta = (2.2, 0) \Rightarrow p(x) \leq p_M(2.2, 0) = \begin{bmatrix} 0.0572 \\ 0.0763 \end{bmatrix},$$
$$\theta = (1.8, 0) \Rightarrow p(x) \leq p_M(1.8, 0) = \begin{bmatrix} 0.0727 \\ 0.0860 \end{bmatrix},$$
$$\theta = (2.2, 0.1) \Rightarrow p(x) \leq p_M(2.2, 0.1) = \begin{bmatrix} 0.0528 \\ 0.0738 \end{bmatrix},$$
$$\theta = (2.2, -0.1) \Rightarrow p(x) \leq p_M(2.2, -0.1) = \begin{bmatrix} 0.0621 \\ 0.0790 \end{bmatrix},$$
$$\theta = (1.8, 0.1) \Rightarrow p(x) \leq p_M(1.8, 0.1) = \begin{bmatrix} 0.0664 \\ 0.0824 \end{bmatrix},$$
$$\theta = (1.8, -0.1) \Rightarrow p(x) \leq p_M(1.8, -0.1) = \begin{bmatrix} 0.0796 \\ 0.0899 \end{bmatrix},$$

The best non taboo solution minimizing $\|p(x)\| : p_M(2.2, 0.1)$ is obtained for $\theta = (2.2, 0.1)$, and the solution for $\theta = (2, 0)$ becomes "taboo".
Now the strategy is then to focus on a local vicinity of this solution in order to find the best one which does not belong to the taboo list. So, we test other solutions that are neighbouring the current one's

$\theta = (2.2, 0), \theta = (2.2, 0.2), \theta = (2, 0.1), \theta = (2, 0.2),$
$\theta = (2.4, 0), \theta = (2.4, 0.1), \theta = (2.4, 0.2).$

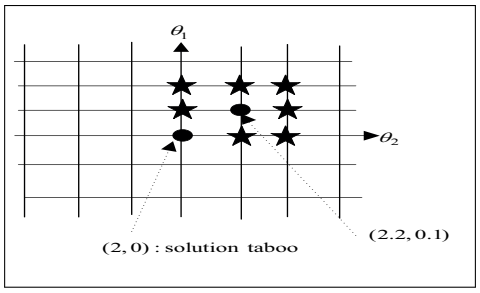


Figure 2: The vicinity of $\theta = (2.2, 0.1)$ solution

$\theta = (2.4, 0) \Rightarrow p(x) \leq p_M(2.4, 0) = \begin{bmatrix} 0.0523 \\ 0.0729 \end{bmatrix}$, $\theta = (2.4, 0.1) \Rightarrow p(x) \leq p_M(2.4, 0.1) = \begin{bmatrix} 0.0528 \\ 0.0727 \end{bmatrix}$, $\theta = (2.4, 0.2) \Rightarrow p(x) \leq p_M(2.4, 0.2) = \begin{bmatrix} 0.0540 \\ 0.0690 \end{bmatrix}$, $\theta = (2, 0.2) \Rightarrow p(x) \leq p_M(2, 0.2) = \begin{bmatrix} 0.0531 \\ 0.0747 \end{bmatrix}$,

$$\theta = (2.2, 0.2) \Rightarrow p(x) \leq p_M(2.2, 0.2) = \begin{bmatrix} 0.0536 \\ 0.0719 \end{bmatrix},$$

The best non taboo solution minimizing $\|p(x)\| : \begin{bmatrix} 0.0540 & 0.0690 \end{bmatrix}^T$ is obtained for $\theta =$

$(2.4, 0.2)$ then the solution $\theta = (2.2, 0.1)$ becomes "taboo".
Now we continue the iteration starting from this new solution $\theta = (2.2, 0.2), \theta = (2.2, 0.3)$, $\theta = (2.4, 0.1), \theta = (2.4, 0.3), \theta = (2.6, 0.1), \theta = (2.6, 0.2), \theta = (2.6, 0.3)$.
The best non taboo solution minimizing $\|p(x)\| : \begin{bmatrix} 0.0558 & 0.0673 \end{bmatrix}^T$ is obtained for $\theta = (2.4, 0.3)$, then the solution $\theta = (2.4, 0.2)$ becomes "taboo". Now we will test the solutions in the neighbourhood of $\theta = (2.4, 0.3)$
The best non taboo solution minimizing $\|p(x)\| : \begin{bmatrix} 0.0575 & 0.0656 \end{bmatrix}^T$ is obtained for $\theta = (2.4, 0.4)$ then the solution $\theta = (2.4, 0.3)$ becomes "taboo". Now we will test the vicinity of this solution
The best non taboo solution minimizing $\|p(x)\| : \begin{bmatrix} 0.059 & 0.064 \end{bmatrix}^T$ is obtained for $\theta = (2.4, 0.5)$, then the solution becomes "taboo".

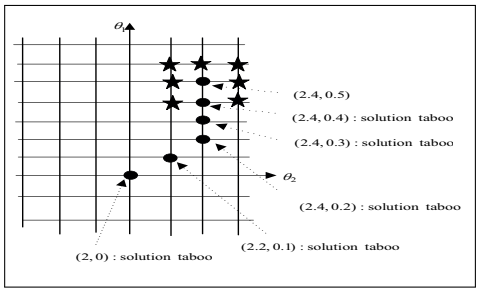


Figure 3: The vicinity of $\theta = (2.4, 0.5)$ solution

At the next iteration the best non taboo solution minimizing $\|p(x)\| : \begin{bmatrix} 0.0605 & 0.0625 \end{bmatrix}^T$ is obtained for $\theta = (2.4, 0.6)$. For the two following iterations the best non-taboo solutions correspond to $p_M(2.4, 0.7)$ and $p_M(2.4, 0.8)$, but $\|p_M(2.4, 0.4)\| = \|p_M(2.4, 0.5)\| = \|p_M(2.4, 0.6)\| = \|p_M(2.4, 0.7)\| = \|p_M(2.4, 0.8)\| = 0.870$, so as we have had 4 iterations without improvement we can stop the research. The control law defined by $\theta = (2.4, 0.4)$, corresponds to the best solution. Hence the evolution of the state vector, and its evolution of the state vector in the attractor defined in figure 4.

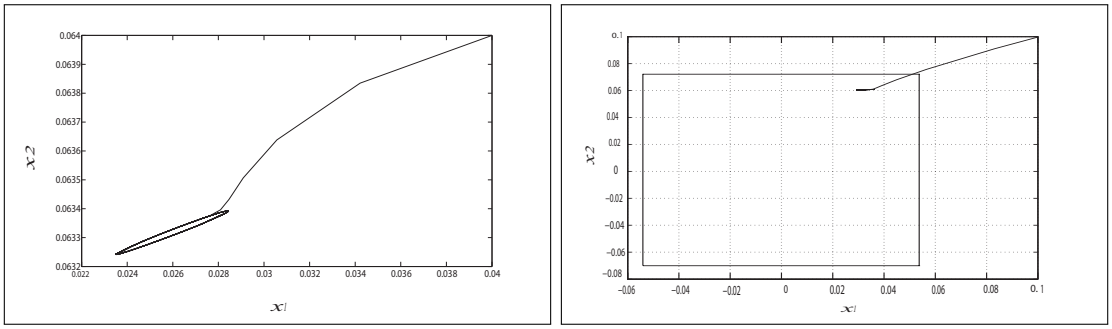


Figure 4: Evolution of the state vector in the attractor

5 Conclusion

The approach proposed here consists, having defined the attractor of the process for a control law depending of parameters to minimize the size of this attractor by implementation of a metaheuristic to determine the optimal values of these parameters. The method presented in this paper is applied, with success, for a second order nonlinear complex system using the concept

of vector norm for the determination of the comparison system. The minimization of the norm of the vector defining the limits of the attractor is realized by using a taboo search method.

Bibliography

[1] Benrejeb, M. ; Borne, P. (1978); On an algebraic stability criterion for non-linear processe, *Interpretation in the frequency domain, Measurement and Control International Symposium MECO*, Athens: 678-682.

[2] Gharbi, A.; Benrejeb, M. ; Borne, P. (2013); On nested attractors of complex continuous systems determination, *Proceedings of the Romanian Academy, Series A*, 14(2):259-265.

[3] Gharbi, A.; Benrejeb, M. ; Borne, P. (2013); New Approach for the Control and the Determination of Attractors for Nonlinear Systems , *2nd International Conference on Systems and Computer Science (ICSCS)*, Villeneuve d'Ascq, France, August 26-27.

[4] Gharbi, A.; Benrejeb, M. ; Borne, P. (2014); Tracking error estimation of uncertain Lur'e Postnikov systems, *2nd International Conference on Control, Decision and Information Technologies (CoDIT'14)* Metz, France, November 3-5.

[5] Benrejeb, M. (2010); Stability study of two level hierarchical nonlinear systems. *Large Scale Complex Systems Theory and Applications IFAC Symposium*, Plenerylecture,Lille, 9(1), 30-41.

[6] Borne, P.; Benrejeb, M. (2008);On the representation and the stability study of large scale systems, *International Journal of Computers Communications and Control*, 3(5): 55-66.

[7] Borne, P. (1987); Nonlinear system stability. Vector norm approach, System and Control Encyclopedia,*Pergamon Press*, Lille, France, 5:3402-3406.

[8] Gentina, J.C.; Borne, P.; Burgat, C.; Bernussou, J., : Grujic, L.T. (1979). Sur la stabilite des systmes de grande dimension. Normes vectorielles, 13(1):57-75.

[9] Gentina, J.C.; Borne, P.; Laurent, F. (1972a). Stabilite des systmes continus non linéaires de grande dimension,*RAIRO*, Aôut:69-77.

[10] Gentina, J.C.; Borne, P. (1972b), Sur une condition d'application du critcre de stabilité linéaire certaines classes de systmes continus non linéaires, *CRAS*, Paris, T. 275: 401-404.

[11] Grujic, L.T.; Gentina, J.C.; Borne, P.; Burgat, C.; Bernussou, J. (1978). Sur la stabilité des systmes de grande dimension. Fonctions de Lyapunov vectorielles,*RAIRO*, 12(4):319-348.

[12] Grujic, L.T.; Gentina, J.C.; Borne, P. (1976). General aggregation of large scale systems by vector Lyapunov functions and vector norms,*Inernational.Journal.of Control*, 24(4): 29-550.

[13] Stefanoiu, D.; Borne , P.; Popescu, D.; Filip, F. G. ; El Kamel, A.(2014); Optimization in engineering sciences. Metaheuristics, Stochastic method and Decision support, *ISTE, Wiley*:20-39.

[14] Siljac, D. D. (1972); Stability of large scale systems under structural perturbations.IEEETrans. *On Syst. Manand Cyber*, 2(5).

[15] Borne, P.; Richard, J.P.; Radhy, N.E. (1996). Stability, stabilization, regulation using vector norms, Nonlinear Systems, 2, Stability and Stabilization,*Chapman and Hall*, Chapter 2; 45-90.

[16] Ghédira K., (2007); Optimisation combinatoire par métaheuristiques, Editions TECHNIP, France.

[17] Ennigron M.; Ghédira K (2004); Flexible Job-Shop Scheduling with Multi-Agent System and Taboo Search,*Journal Européen des Systmes Automatisés JESA*, 38: 7-8.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):233-247, April 2016.

Content Based Model Transformations: Solutions to Existing Issues with Application in Information Security

J. Janulevičius, S. Ramanauskaitė, N. Goranin, A. Čenys

Justinas Janulevičius*, Simona Ramanauskaitė, Nikolaj Goranin, Antanas Čenys
Vilnius Gediminas Technical University,
Lithuania, LT-10223 Vilnius, Sauletekio al. 11
justinas.janulevicius@vgtu.lt, simona.ramanauskaite@vgtu.lt, nikolaj.goranin@vgtu.lt, antanas.cenys@vgtu.lt
*Corresponding author: justinas.janulevicius@vgtu.lt

Abstract: Model-Driven Engineering uses models in various stages of the software engineering. To reduce the cost of modelling and production, models are reused by transforming. Therefore the accuracy of model transformations plays a key role in ensuring the quality of the process. However, problems exist when trying to transform a very abstract and content dependent model. This paper describes the issues arising from such transformations. Solutions to solve problems in content based model transformation are proposed as well. The usage of proposed solutions allowing realization of semi-automatic transformations was integrated into a tool, designed for OPC/XML drawing file transformations to CySeMoL models. The accuracy of transformations in this tool has been analyzed and presented in this paper to acquire data on the proposed solutions influence to the accuracy in content based model transformation.
Keywords: Cyber Security Modeling Language; Model Transformation; Model Driven Engineering.

1 Introduction

Model-Driven Engineering (MDE) [1] uses models as a reference in various phases of software engineering. The model is created in the early stages and reused later for a number of purposes. Since most of the processes and aspects can be formalized and represented as a model - they are commonly used for their commodity. To obtain a certain output from different type of models is vital for MDE and a variety of solutions has been proposed by the research community, spanning from experimental approaches [2] to frameworks [3]. Model transformation is a very actual problem in practice as well as research as new types of models appear and more accuracy is needed.

The aim of this paper is to simplify transformation of abstract, content based model transformations. Content based models have very abstract structure. It can be a benefit as it increases the meta-model adaptation area, but one of the main drawbacks is that model transformations have to be done in content rather than structure level. Two main problems with content based model transformations are presented in this paper along with the solutions. To analyze the effectiveness proposed solutions, they are integrated into a tool for OPC/XML drawing file to CySeMoL model transformation. The accuracy results of the transformation are presented in this paper as well.

2 Related Works

Numerous research approaches have been carried out on model transformations, as it is a very useful process that not only leads to automation of processes [1], ease of migrating data [2] and

at the same time liberating the systems from legacy components [3], but also, most importantly, from the economic point of view - reducing costs by reusing the existing data [4]. Methodologies have been developed to manage the correctness of data, stored as model attributes in the process of transformation. Of which, the triple graph grammar case offers a methodology for attribute handling for bidirectional model transformations [5].

Dedicated model transformations for information security modeling is a relatively new yet very important area for research. Model-driven security is a growing trend with an expanding list of tools and methodologies for the subject [6]. Approaches, such as SecureUML model transformation semantics and analysis [7] as well as transformations between SecureUML and UMLsec [8] exist. However, new information security assessment tools require a more flexible approach with an ability to acquire data from less formalized model structures as information security modeling typically involves representing the analyzed infrastructure in a formal way. Architectural modeling languages are typically used in this case. They include SySML [9], Business Process Modeling Notation (BPMN) [10] that enable representation of information system architecture and system environment through diagrams that can be used for various forms of analysis, one of which is security. Some of them also offer extensions for Industrial Control System Security Analysis [11]. However, the aforementioned modeling languages do not offer the reasoning process. Some solutions that offer modeling capabilities along with the reasoning based on the systemized expert knowledge base exist. One of them is OpenMADS [12], the other is Cyber Security Modeling Language (CySeMoL) [13].

2.1 Model to Model Transformations

Model transformation enables information reuse preserving consistency between the two models [14]. In this case preservation of relationship between the source and target models as well as heterogeneity of the transformed data comes as a challenge [15]. Model transformation is facing two issues: impedance mismatch and heterogeneity [16]. Heterogeneity forces to deal with different data models and encodings of values. Impedance mismatches are caused by the difference between logical schemas required by the applications and the ones exposed by data sources. These issues support the idea that data consistency between the models by adjusting the level of abstraction is the main task in order to avoid data loss along the transformation process [17].

Model transformation patterns are obtained by using the Formal Concept Analysis [18], where relations and element meta-classes of target and source models are linked together based on model classification group links that have similarities between them.

2.2 Content Dependent Model Transformations

Some languages are equipped with an abstract meta-model. The content of the model is provided in text based form as the label or property value of an element (see Fig. 1). This type of meta-model is very common in general purpose systems. The abstract meta-model allows presentation of wider, not predefined content.

According to Taxonomy of Model Transformations [19] this type of transformation is considered to be exogenous, vertical transformation. Typically it is used as synthesis of a higher-level, more abstract, specification into a lower-level, more specific one.

Transformation of such model is very content dependent. Therefore the definition of transformation rules is time consuming due to these reasons:

- Every model component and property label has to be listed in order to write a transformation rule. As labels are human generated, the list is infinite or very long as all components and properties can have multiple synonyms.

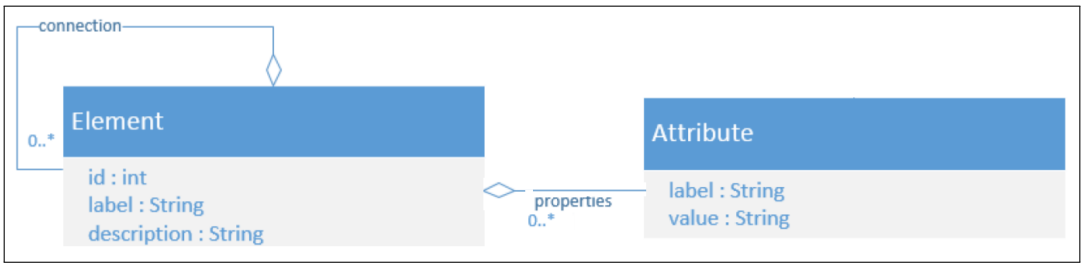


Figure 1: Example of general purpose source meta-model

- All component synonyms have to be taken into the account for the transformation rules. Therefore multiple rules are required for one target concept or rule.

These reasons cause higher resource consumption compared to discrete formal models. There is also a level of uncertainty, as some of the synonyms or concepts can be missed out of the model transformation rules and the process will not be able to transform the elements into the target model. An example of a content dependent source model definition is presented in Fig. 2.

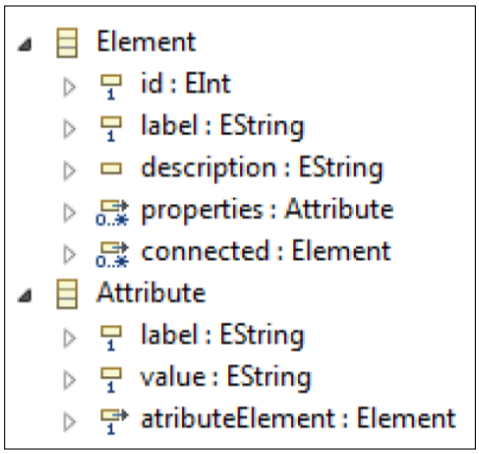


Figure 2: Example of content dependent source model definition in ECore file

An example of transformation rules of such model in ATL is presented in Fig. 3.

The provided example in figure 3 only has five synonyms, however, the list of synonyms increases by taking different languages, dialects and situations into account. Therefore, it would be difficult to modify the list of synonyms if the rule is hardcoded into the source code of software product. A solution for easy synonym integration is a valuable improvement.

The source model element type identification in content dependent models complicates when abstract element does not have a name nor a description. In such situation the information is not enough. Therefore element identification can be performed according to the structure of the element. However this task in content dependent models is complicated as well as there is no predefined specific element structure for different content source elements. Therefore, to identify the type of source model, rules can to be used to check if the containing attributes match the ones expected in the target model (see. Fig. 4).

Since the source model is abstract, the transformation is facing some complications as well:

- The attribute set for each element has to be defined individually as there is no list of attribute labels and values in the meta-model. The complications are amplified if the attribute labels are hardcoded in the software source code.

```
rule Element2Network {
  from  //define which component to take from the source
    s:Abstract!Element in IN (s.label = 'Network' or
      s.label = 'Net' or s.label = 'Internet' or
      s.label = 'LAN' or s.label = 'WAN')
    to
      //define how the source element have to be transformed
        n1:Cysemol!NetworkZone(  //creating NetworkZone element
          id <- s.id,       //with appropriate properties
          name <- s.label,
          originalConnection <- s.connected.id,
          interface <- n2
        ), //creating NetworkInterface element to connect NetworkZone
       n2:Cysemol!NetworkInterface(
          network <- n1
        )
}
```

Figure 3: Example of content dependent element transformation rule in ATL for element, associated to NetworkZone in target model

```
rule Element2Computer{
  from //searches for elements with needed attributes
    s:Abstract!Element in IN (
          s.attribute->collect(l | l.label)->
          includesAll(Set{'cpu', 'ram', 'hdd'})
        )
  to
    n1:Cysemol!OperatingSystem(
          name <- 'Computer with '+s.label
        )
}
```

Figure 4: Example of content dependent element transformation according to the obtained parameters rule in ATL

- A decision has to be made on which of the attributes provide a better definition of the element, at the same time which ones are unimportant and may be discarded. In case too many attributes are compared in the transformation rule, a missed attribute in the source model would make the rule worthless. On the contrary, if not enough attributes are be used in the transformation rule, element can be inconclusively (multiple possibilities) identified.
- Attribute labels and values are content based. Therefore multiple labels and values can be linked to the same content. Knowing all possible values is nearly impossible and it increases the complexity of transformation rules.
- Source element identification according to its structure element label, attribute labels and attribute values can be the crucial element. There is no unified methodology for measurement of the significance of the element identity from list of possible cases.

All these reasons make the source model element difficult to identify using only static rules.

3 Assumptions for Model to Model Transformation Improvement

Existing model transformation methodologies seem to have drawbacks when dealing with specific situations or have to be applied in dynamic situations [20]. Therefore new solutions are proposed to improve the process and provide an alternative method that improves the efficiency and accuracy of the transformations. In this chapter ideas on how current situation in specific situations can be improved using advanced techniques, such as grammar-based model transformations [21] and model transformation by-example [22–25] element identification are presented.

3.1 Dictionary Based Element Identification

A context analysis is a compex task as some words can have different meaning, synonyms for most of words exists etc. One of ways to implement context analysis is synonym based analysis. This aproach is used in web serach engine optimization [26] and user review analysis [27] during the last two years and shows promissing results. Therefore dictionary based element identification approach on model to model transformation is proposed for simplification of the transformation of content dependent models. The main idea is to use a synonym database for each target meta-model element. This is done by providing additional dictionary meta-model (see Fig. 5) and input of synonyms for each of the target model elements.

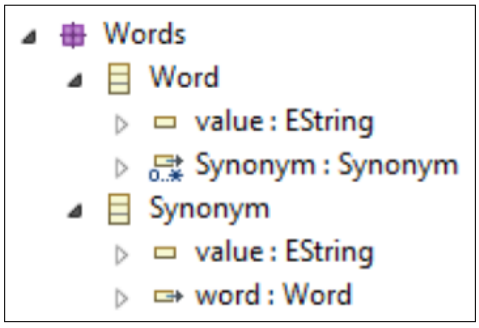


Figure 5: Dictionary meta-model for element identification

The condition for element identification in source model is simplified and achieved using only one condition rather than a list of conditions. An example of synonym search in dictionary model and its usage are provided in Fig. 6 and Fig. 7.

```
helper def : getBySyninym(sn : String) : String =
  if ( //looking for a synonym in the dictionary model
    Dictionary!Synonym.allInstances()->
       select(e | e.value.toLowerCase().startsWith(sn.toLowerCase()))
  ).isEmpty()
  then //if there is no synonym - we take the same value
    '<<'+sn+'>>'
  else //if there is a synonym - we take the word
    Distionary!Synonym.allInstances()->
       select(e | e.value.toLowerCase() = sn.toLowerCase())->
       collect(e | e).first().word.value
  endif;
```

Figure 6: Example code for search of an element name by comparing it to existing synonyms

```
...
  from
    s:Abstract!Element in IN (
          //helper usage to get synonyms from the dictionary
          thisModule.getBySynonym(s.label) = 'network'
        )
...
```

Figure 7: Simplified situation of Fig. 6 used to identify element type of source model

The proposed solution is more flexible as the list of synonyms for target model elements can be provided as input file and modified at any given moment. These changes do not require source code to be changed, so the dictionary file can vary depending on the target metamodel, language and other factors.

3.2 Example Based Element Identification

Example based model transformation is well known strategy for transform one model to another and other tasks. This technology is used for images and videos color transformation [28], semantic data analysis from a give string [29] etc. Therefore we propose to use example based model transformation in order to simplify the transformation of content dependent models where elements are defined by structure only. In this case a database of target meta-model element examples is used. For each target meta-model element one example is stored in the database of source meta-model. To simplify the ATL code a new meta-model was created as a copy of source meta-model (see Fig. 8).

When example target elements of source meta-model are presented, each source element is compared to the one stored in the example database to find the most similar element based on its structure. Similarity estimation q is calculated using as follows:

$$q = \frac{m}{s} + \frac{m}{e} \tag{1}$$

In (1) m is the number of matched labels between source and example elements; s is the number of attributes in the source element; e is the number of attributes in the example element.

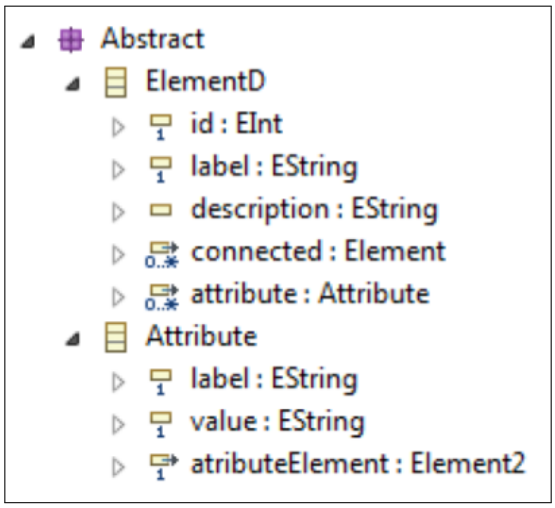


Figure 8: Modified source meta-model structure for target element descriptionl

Sum of these two proportions takes both attribute redundancy and shortage into account. The implementation of this method is presented in Fig. 9.

There is some space for the improvement of this example by optimizing the code, adding the comparison and attribute labels. Example based element identification should be executed after dictionary based element and attribute transformation as the labels and values of source model are content dependent and dictionary usage leads to formalization.

4 Case Analysis: OPC /XML drawing file transformation to CySeMoL

OPC is a container file standardized format [30]. An OPC format for storing graphical notation has an extension of .vsdx. The structure of the OPC/XML drawing file is presented in Fig. 10. The information about the element layout of the pages are stored in separate XML format files in sub-directory "visio/pages" (marked red in Fig. 10). In this case object and relationship information is extracted from files stored in this directory.

For this model transformation specific tags of the XML files are used. They are:

- Shapes - describes a shape array;
- Shape - describes a shape and its' identification number, name, type and master template;
- Cell - it is a versatile tag, containing information about name and value of many properties of cells under Shape and Section tags;
- Text - gives text output, most commonly an object of instance, visible graphically;
- Section - contains attribute information under it;
- Row - stores attribute information;
- Connects - describes array of connections;
- Connect - defines a connector between instances, specifying sheets, cells and parts connected.

```
helper def : calculateValue(
  a:Integer, b:Integer, c:Integer) :
  //calculates the similarity value q
  Integer = ((m/a + m/b)*100).floor();

helper def : calculateSimilarity(
  a:Abstract!Element, b:Abstract2!ElementD) :
  Integer = thisModule.calculateValue(
    //calculate a value (number of attributes in source)
    a.attribute->collect(l | l.label)->size(),
        //calculate b value (number of attributes in target)
    b.attribute->collect(l | l.label)->size(),
        //calculate m value (number of maching attributes)
    ((a.attribute->collect(l | l.label).asSet().
      intersection(b.attribute->collect(e | e.label).asSet()))->
    size())
  );

helper context Abstract!Element def : getByStructure() :
  String = let sk : String = self.getByExample2() in
    //skipping first letters, which indicates similarity
        //as the most similar element label is presented at the end
    sk.substring(5, sk->size());

helper context Abstract!Element def : getByExample2() :
  String = let elem : Sequence(Abstract2!ElementD) =
    Abstract2!ElementD.allInstances()->asSequence() in
    elem->iterate(p; label : String = '000' |
          //looking for the maximum q value
          if thisModule.calculateSimilarity(self, p) >
            label.substring(1, 3).toInteger()
          then let numb : Integer =
            //returning 3 digit value and label of the element
            thisModule.calculateSimilarity(self, p) in
        ('000'+numb).substring(('000'+numb)->size()-2,
        ('000'+numb)->size())+' '+p.label
      else label //otherwise returning the same value
      endif
        );
```

Figure 9: Example code for search of element name by comparing it to example structure

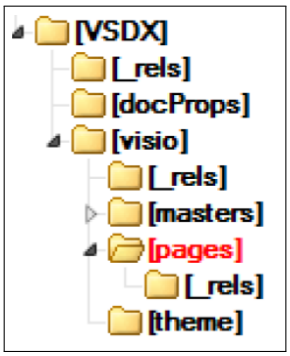


Figure 10: The structure of an OPC /XML drawing file

The proposed transformations were implemented as a tool to convert .vsdx file data into CySeMoL meta-model. The proposed transformation methods use synonym database for target meta-model. The current version is constructed for English language only. It stores over 6000 synonyms for most common CySeMoL classes and attributes. An additional integrated database for connection comparison is built as well. This database has up to 1000 possible connections between CySeMoL elements and serves as an alternative to the example based content dependent model transformation. The ideas of model transformations based on triple graph grammars are integrated [31] as well as class identification using missing elements based on the target model connection example database. This allowed a more accurate class mapping.

4.1 Transformation Accuracy Estimation Experiment

An experiment has been carried out to estimate the accuracy of the proposed model transformation methods. This experiment includes estimation of the results provided by a group of 48 Informatics Engineering senior year students. They were assigned to draw two diagrams in Microsoft Visio 2013 tool: one to present basic SMEs local network and one - basic web server diagram. The diagram type, diagram elements, description, and detailing level were entirely a matter of choice. The only constraint was to use English language exclusively. The experiment resulted in 86 different diagrams. The most common examples are presented in Fig. 11 and Fig. 12.

All provided diagrams were transformed to a CySeMoL model. The transformed models were analyzed and compared to expert prepared CySeMoL model in the EAAT tool. The EAAT tool allows graphical representation of cybersecurity area as well conforms to the model requirements for CySeMoL. Automated formal comparison as the results were not compliant to any formalization. Therefore multiple output results were generated. This fact required to analyze every situation individually by experts.

During the experiment most CySeMoL models had more elements in comparison to the source model file data. This is due to some additional elements had to be added as interfaces (see Fig. 13 and Fig. 14 as results of Fig. 11 and Fig. 12 in CySeMoL).

4.2 Results of Transformation Accuracy Estimation Experiment

The network and Web server diagrams use different Microsoft Visio diagram templates and elements, therefore they are analyzed separately. Diagram description level categorization was to the following categories: no diagram element descriptions; defined diagram element name; defined associated diagram element properties. These categories are used for assessment of usefulness of diagram name and property descriptions.

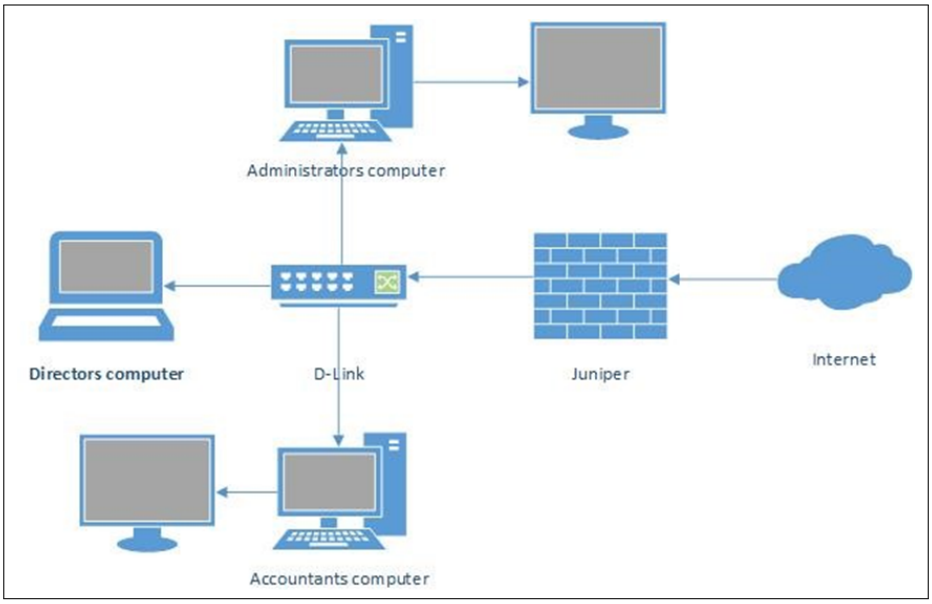


Figure 11: Typical result diagram for basic SMEs network

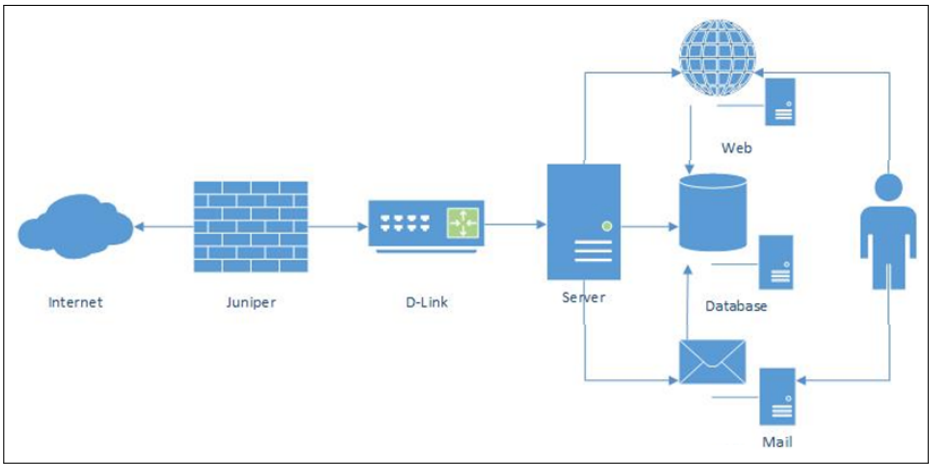


Figure 12: Typical result diagram for basic web server

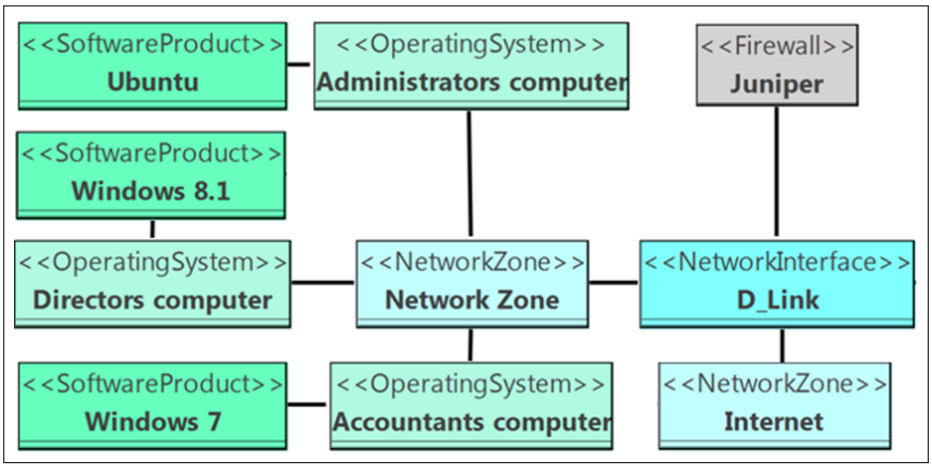


Figure 13: Typical result diagram for basic SMEs network in CySeMoL

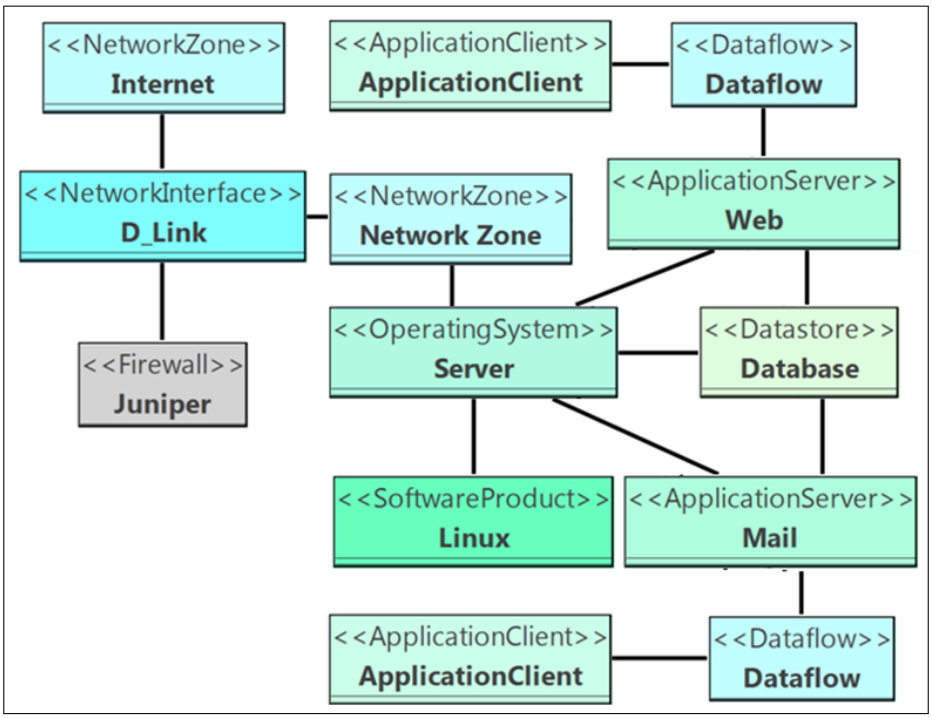


Figure 14: Typical result diagram for basic web server

The summary of analyzed file data and transformation accuracy for different diagram type and description level is presented in Tables 1. The Table 1 shows the number of property level detailed SMEs network OPC/XML files are bigger comparing to Web server diagrams as well as property level detailed diagrams usually have less components, comparing to less detailed Microsoft Visio diagrams. The most important - the generated CySeMoL model has a bigger number of elements, comparing to the source files as existing CySeMoL meta-model requires more elements to evaluate the risk.

Transformation accuracy analysis shows that the proposed transformation method is capable of generating CySeMoL models from more abstract OPC/XML drawing files - 94% of generated CySeMoL model elements are correctly identified; 88% of CySeMoL objects are transformed to template level (with defined default values); 87% of connections between CySeMoL objects are added as expected (see Table 1).

As seen in table 1 the accuracy is dependent on used diagram content and detailing level. OPC/XML drawing files have a predefined attribute list, however not all diagram elements are covered. Therefore some elements cannot be detailed by defining their attribute values. Moreover, model transformation might fail due to incorrect diagram element description, using modified terms. This requires maintenance of synonym database, keeping it up to date with human language and technology improvement changes.

Table 1: Summary of OPC/XML files and CySeMoL model data and transformation accuracy

	SME networm situation			WEB server situation			**Total**
	Components and links with no descriptions	Component name added with no properties	Component name and properties added	Components and links with no descriptions	Component name added with no properties	Component name and properties added	
Number of files	6	18	22	21	15	4	**86**
Number of .vsdx elements	42	121	145	169	127	22	**626**
Number of CySeMoL elements	130	349	435	569	412	105	**1976**
objects	62	184	231	268	193	50	**988**
connections	68	165	204	301	219	58	**1015**
Correctly identified element %	95%	100%	100%	96%	100%	100%	**98%**
Correctly identified connection %	81%	98%	100%	64%	98%	97%	**87%**
Total	**88%**	**99%**	**100%**	**79%**	**99%**	**98%**	**94%**

5 Conclusions

The proposed model transformation methods offer a semi-automatic abstract relationship-based model transformation into more detailed, domain specific template-based model. As this is a content dependent situation - detailed knowledge databases are required to extract knowledge and identify model elements according to text based names and descriptions.

Textual dictionary based analysis is used for element identification, however further reasoning is required for definition of the source model element relation to destination metamodel. Element identification in source model is one of the most important steps in model to model transformation. The combination of dictionary association, structure comparison and relationship similarities provided a 94% accuracy in this model transformation. For further improvements it requires a detailed list of attributes in order to increase the model transformation accuracy.

Bibliography

[1] L. Levi, M. Amrani, J. Dingel, L. Lambers, R. Salay, G. M. K. Selim, E. Syriani and M. Wimmer (2014), Model Transformation Intents and their Properties, *Software Systems & Modeling*, 1-38.

[2] L. M. Rose, M. Hermannsdoerfer, S. Mazanek, P. V. Gorp, S. Buchwald, T. Horn and E. Kalnina (2014), Graph and Model Transformation Tools for Model Migration, *Software & Systems Modeling*, 13(1): 323-359.

[3] G. M. K. Selim, S. Wang, J. R. Cordy and J. Dingel (2012), Model Transformations for Migrating Legacy Models: An Industrial Case Study, *ECMFA, LNCS 7349*, 90-101.

[4] S. Sen, N. Moha, V. Mahe, O. Barais, B. Baudry, J. M. Jezequel (2012), Reusable Model Transformations, *Software & Systems Modeling*, 11(1): 111-125.

[5] L. Lambers, S. Hildebrandt, H. Giese and F. Orejas (2012), Attribute Handling for Bidirectional Model Transformations: the Triple Graph Grammar Case, *Electronic Communications of the EASST*, 49: 1-16.

[6] D. Basin, M. Clavel and M. Egea (2011), A Decade of Model-driven Security, *SACMAT11 Proceedings of the 16th ACM symposium on Access control models and technologies*, 1-16.

[7] A. D. Brucker, J. Doser and B. Wolff (2006), A Model Transformation Semantics and Analysis Methodology for SecureUML, *Lecture notes in computer science*, Berlin, Springer, 306-320.

[8] R. Matulevicius and M. Dumas (2011), Towards Model Transformation Between SecureUML and UMLsec for Role-based Access Control, *Databases and Information Systems*, 6: 1-14.

[9] S. Friedenthal, A. Moore and R. Steiner (2014), *A Practical Guide to SysML*, Waltham: Elsevier, 2014.

[10] M. Chinosi and A. Trombetta (2012), BPMN: An introduction to the standard, *Computer Standards & Interfaces*, 34: 124-134.

[11] L. Lemaire and J. Lapon (2014), A SysML Extension for Security Analysis of Industrial Control Systems, *2nd International Symposium for ICS & SCADA Cyber Security Research 2014 (ICS-CSR 2014)*,1-9.

[12] E. C. Andrade, M. Alves, R. Matos, B. Silva and P. Maciel (2013), OpenMADS: An Open Source Tool for Modeling and Analysis of Distributed Systems, *Computer Safety, Reliability, and Security*, Lecture Notes in Computer Science, 8153: 277-284.

[13] T. Sommestad, M. Ekstedt and H. Holm (2013), The Cyber Security Modeling Language: A Tool for Assessing the Vulnerability of Enterprise System Architectures, *Systems Journal*, 7: 363-373.

[14] M. Biehl (2010), *Literature study on model transformations*, Royal Institute of Technology, Stockholm, 2010.

[15] K. Czarnecki and S. Helsen (2003), Classification of model transformation approaches, *2nd OOPSLA Workshop on Generative Techniques in the Context of the Model Driven Architecture*, 2003.

[16] P. A. Bernstein and S. Melnik (2007), Model Management 2.0: Manipulating Richer Mappings, *SIGMOD07, Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, 1-12.

[17] T. Frisendal (2012), Business Concept Mapping, *Concept Maps: Theory, Methodology, Technology Proc. of the Fifth Int. Conference on Concept Mapping*, 1-4.

[18] B. Ganter and R. Wille (2012), *Formal Concept Analysis: Mathematical Foundations*, Berlin: Springer Science & Business Media, 2012.

[19] T. Mens and P. V. Gorp (2015), A Taxonomy of Model Transformation, *Electronic Notes in Theoretical Computer Science*, 152: 125-142.

[20] K. Czarnecki and S. Helsen (2006), Feature-Based Survey of Model Transformation Approaches, *IBM Syst. J.*, 45(3): 621-645.

[21] G. Besova, D. Steenken, and H. Wehrheim (2015), Grammar-based model transformations, *Comput. Lang. Syst. Struct.*, 43(C): 116-138.

[22] H. Saada, X. Dolques, M. Huchard, C. Nebut and H. Sahraoui (2012), Generation of operational transformation rules from examples of model transformations, *Model Driven Engineering Languages and Systems. Lecture Notes in Computer Science*, 7590: 546-561.

[23] M. Wimmer, M. Strommer, H. Kargl and G Kramler (2007), Towards Model Transformation Generation By-Example, *HICSS 2007, 40th Annual Hawaii International Conference on System Sciences*, 285b.

[24] G. Kappel, P. Langer, W. Retschitzegger, W. Schwinge and M. Wimmer (2012), Model Transformation By-Example: A Survey of the First Wave, *Conceptual Modelling and Its Theoretical Foundations*, 197-215.

[25] D. Varro (2006), Model Transformation by Example, *Model Driven Engineering Languages and Systems*, 410-424.

[26] P. Arora and T. Bhalla (2014), A Synonym Based Approach of Data Mining in Search Engine Optimization, *International Journal of Computer Trends and Technology (IJCTT)*, 12(4): 201-205.

[27] B. Ma, Z. Dongsong, Z. Yan and T. Kim (2013), An LDA and Synonym Lexicon Based Approach to Product Feature Extraction from Online Consumer Product Review, *Journal of Electronic Commerce Research*, 14(4): 304-314

[28] Y. Chang, S. Saito and M. Nakajima (2005), Example-based color transformation for image and video, *Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia (GRAPHITE05). ACM*, New York, NY, USA, 347-353.

[29] R. Singh and S. Gulwani (2012), Learning semantic string transformations from examples, *Proc. VLDB Endow.*, 5(87): 740-751.

[30] International Organization for Standartization (2012), ISO 29500-2:2012 Information technology - Document description and processing languages - Office Open XML File Formats - Part 2: Open Packaging Conventions. Third Edition, Geneva: International Organization for Standartization.

[31] F. Hermann, H. Ehrig, F. Orejas and U. Golas (2010), Formal Analysis of Functional Behaviour for Model Transformations Based on Triple Graph Grammars, *Graph Transformations*, Berlin, Springer, 155-170.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):248-258, April 2016.

Sparse Online Learning for Collaborative Filtering

F. Lin, X. Zhou, W.H. Zeng

Fan Lin*
Software School, Xiamen University
China, 308B of General Office at Haiyun Campus, Xiamen University, Xiamen, 361009
*Corresponding author: iamafan@xmu.edu.cn

Xiuze Zhou
Department of Automation, Xiamen University
China, General Office at Haiyun Campus, Xiamen University, Xiamen, 361009
zhouxiuze@foxmail.com

Wenhua Zeng
Software School, Xiamen University
China, 502 of General Office at Haiyun Campus, Xiamen University, Xiamen, 361009
whzeng@xmu.edu.cn

Abstract:
With the rapid growth of Internet information, our individual processing capacity has become over-whelming. Thus, we really need recommender systems to provide us with items online in real time. In reality, a user's interest and an item's popularity are always changing over time. Therefore, recommendation approaches should take such changes into consideration. In this paper, we propose two approaches, i.e., First Order Sparse Collaborative Filtering (SOCFI) and Second Order Sparse Online Collaborative Filtering (SOCFII), to deal with the user-item ratings for online collaborative filtering. We conduct some experiments on such real data sets as MovieLens100K and MovieLens1M, to evaluate our proposed methods. The results show that, our proposed approach is able to effectively online update the recommendation model from a sequence of rating observation. And in terms of RMSE, our proposed approach outperforms other baseline methods.
Keywords: Recommender systems, Collaborative Filtering, Online learning, SOCFI, SOCFII

1 Introduction

With the prosperity of such large-scale online commercial websites and online shopping websites as Amazon [1], Barnes, Netflix, eBay, etc, users are continuously exposed into increasing amount of items. Consequently, the information flow which is increasingly complex and huge makes the user lost, and thereby be tired of the inefficient search. In order to deal with this problem, and also to predict the user's unknown preferences based on some user's preferences we have studied [2,3], a modern technique named Collaborative Filtering(CF) is put forward.CF has become widely used as one of the most successful learning techniques to build real-world recommendation systems.

Consider online e-commerce applications where a user wishes to watch a movie or buy a product, the system offers recommendations using CF techniques in exploiting one's previous preference and that of others. A good recommendation system is extremely beneficial to users in accurately predicting their preferences and providing satisfactory recommendations, and consequently benefiting the company [1]. The fundamental assumption of CF is that if two users rate many items similarly, they will be likely to rate other items similarly [2].

Despite the successful application in such many fields as book [6], music [7], news [8], etc, all these traditional CF approaches share a common but critical drawback that these approaches have to be re-trained completely from scratch whenever new training data arrives, which is clearly non-scalable for large real recommendation systems in which user's rating data often arrives sequentially and frequently [5, 9]. It takes quite a long time to learn a new model when large numbers of parameters need to be estimated. The popularity of items and the interests of users are always changing over time [10, 11]. Therefore, Recommendation approaches shall take these changes into consideration.

Traditional CF approaches work well on the user-item rating data when the latent factors number k is very small. However, when the k become large, these approaches fail to well deal with the user-item rating data, since the user-item rating data is sparse and of large scale, and many approaches need a large k to get accurate results. Although the batch algorithm of matrix factorization has a high accuracy, we can't stand the high memory cost and time complexity in the real-world recommender system.

Nowadays, the data recommendation system which has a enormous amount of data is characterized as followed [12]: (1) high volume, system need to deal with huge amount of training data; (2) high velocity, new data often arrives very rapidly and sequentially; (3) high dimensionality, the data from users has a large number of features; (4) high sparsity, many feature elements are zero.

To tackle the above challenges, recent years have witnessed some studies for online collaborative filtering [3, 4]. The state-of-the-art Online Collaborative Filtering (OCF) approach avoids the highly expensive re-training cost of traditional batch matrix factorization algorithms by applying the simple online gradient descent (OGD) algorithms to solve the matrix factorization task [3]. Muqeet Ali et al. proposed a parallel collaborative filtering for streaming data by using distributed stochastic gradient descent algorithm [4].

Unfortunately, these methods are generally based on the first order optimization framework (e.g., online gradient descent) to find the optimal solutions of low-rank matrix factorization. The ignorance of second order information results in the slow convergence of these approaches. Besides, the latent factors number is actually quite likely to be very large which is a difficulty for the first order optimization framework even the framework which has already taken the second order information. To tackle this issue, we propose to solve the following sparse collaborative filtering problem. To address the weakness of these first order or second order online CF approaches and reduce data storage space and increase computing speed, we propose such Sparse Online Collaborative Filtering (SOCF) as First Order Sparse Collaborative Filtering (SOCFI) and Second Order Sparse Online Collaborative Filtering (SOCFII). Our proposed approach is able to effectively online update the recommendation model from a sequence of rating observation. The Sparse Online Collaborative Filtering (SOCF) takes consider the latent factors of the low rank matrix and online second order optimization method. The key idea of SOCF is to not only update the user and item weight vectors at each round, but also estimate their distribution and take full account of large latent factors. Because of full account of this case, SOCF converges significantly faster and thus achieves much lower values of RMSE and MAE than those of the regular first order algorithms when receiving the same amount of rating observations.

The rest of the paper is organized as follows. Section 2 introduces the background information and presents the problem formulation. Section 3 exhibits the proposed Sparse Collaborative Filtering algorithm, which takes first order and second order information into consideration. Section 4 presents our experimental results and analysis. Section 5 draws conclusions and discusses the future work.

2 Background Information and Related Work

In this section, we introduce some background information, related works and the problem we're going to solve.

Collaborative filter (CF) and content-based filtering are two strategies widely used in recommendation systems for recommending items for users. CF makes prediction by using only the user-item interaction indormation without additional information or domain knowledge, so it has a wider application. The key idea of the CF is that users who have similar preferences in the past are likely to have similar preferences in the future.

Memory-based algorithms show good performance on accuracy, but they cannot handle scalability and sparsity problems of data. All these CF algorithms achieved good results without using additional information. In order to solve the data sparsity problems, many model-based CF methods have been proposed. Model-based CF techniques aim at building a model to represent user rating data, and use that model to predict user preference for a specific item. For example, the Singular Value Decompositon (SVD) obtains the main factors to reduce dimensimality. Hofmann converts the Latent semantic model from information retrieval to collaborative filtering. These models not only reduce the dimensions of the user-item matrix and smooth out the noise information, but also help the algorithm to alleviate scalability of data.

Recommendation systems provide an effective way for information filtering to discover useful information according to the historical preferences expressed by users [15]. At present, CF approaches, as the most widerly used method in recommendation system, can be generally grouped into two types: model-based CF and memory-based CF. Model-based CF approaches provide item with recommendation by first developing a model of user ratings, and then predicting user's preference for a specific item through reained model [13–16]. While memory-based approaches predict rating of users according to all user ratings [1, 7, 20]. Generally, the Model-based CF approaches is more accurate than memory-based approaches [21].

Matrix factorization is one of the most popular and the state-of-the-art methods of model-based CF approaches, which was used by the winner of the Netflix prize [22–25]. SVD puts the items that are highly relevant and apparent together as a Singular factor, and breaks up the vector into a small order approximation matrix. One user with one item represents a vector in this space and the rating that a user assigns to an item is the dot product of their feature vectors [26]. The key idea of latent factor model assumes that the similarity between users and items is discovered by the lower-dimension data. The system minimizes the regularized squared error on the set of known ratings to learn the factor vectors [27]. Low rank matrix factorization is considered to be a very effective method and achieves good results in practice [20, 28].

We will base our matrix factorization study on the collaborative filtering, which is traditionally defined as:

$$\sum_{(a,b)\in A} l(r_{a,b}, U_a, V_b) + \frac{\lambda}{2}\left(\sum_{a=1}^{m}\|U_a\|^2 + \sum_{b=1}^{n}\|V_b\|^2\right)$$

Where $A \subseteq \{(a,b)|r_{a,b} \text{ is known}\}$, $l(r,U,V) = (r - U^T V)^2$, $U_a, V_b \in R^k$ and λ is a regularization parameter. A is the user-item rating set we have known.

Although the matrix factorization technique can obtain high accuracy, we cannot stand with it for a long time to run. Stochastic gradient descent algorithm requires an iterative many times until convergence. In practical, the model-based approaches have to be retrained completely for new records when new users' rating data arrives sequentially and frequently [6]. In contrast to traditional collaborative filtering algorithms, online learning promptly update the predictive model and able to avoid expensive re-training cost when a new instance appears [7, 8]. In online algorithms, these models need to be retrained when each new data arrives. The online

algorithms only need a single iteration which processes the events in the order of time over the training data [30]. Although the online learning algorithm loses some accuracy, it avoids high time complexity and memory cost.

3 Sparse Collaborative Filtering Algorithm

In this section, we introduce our proposed sparse collaborative filtering algorithms, including First Sparse Order Collaborative Filtering (SOCFI) algorithm and Second Order Online Sparse Collaborative Filtering (SOCFII) algorithm.

3.1 First Online Sparse Collaborative Filtering Algorithm

With N users and M items in the user-item rating matrix is broken up into two low rank matrixes. The user-item rating matrix $R \in R^{N*M}$ is broken down into two low rank matrixes. U_a is the a-th row from the user matrix $U \in R^{N*K}$, and V_b is the b-th row form item matrix $V \in R^{M*K}$. The rank of $K \ll \min\{N, M\}$. $r_{a,b}$ is the movie b rated by user a. The predicted score is the dot product of U_a and V_b, i.e., $\widehat{r_{a,b}} = U_a^T V_b$. $|C|$ represents the number of observed ratings. In general, one can define different type of loss function for different purposes. For example, for the Root Mean Square Error (RMSE), i.e., , we define the loss by the square error function as:

$$RMSE = \sqrt{\frac{1}{|C|} \sum_{(a,b) \in C} (r_{a,b} - \widehat{r_{a,b}})^2}$$

We define the loss by the square error function as:

$$l(U_a, V_b, r_{a,b}) = (r_{a,b} - U_a V_b^T)^2$$

And for the Mean Absolute Error (MAE), i.e., $MAE = \frac{1}{|C|} \sum_{(a,b) \in C} |r_{a,b} - \widehat{r}_{a,b}|$, we define the absolute loss function as:

$$l(U_a, V_b, r_{a,b}) = |r_{a,b} - U_a V_b^T|$$

Traditionally, k is treated as the latent factors number. When k is set as a small value, the traditional algorithm will work well. However, when k is set as a large number, the algorithm will fail, which implies the traditional algorithm cannot tackle tasks with large factors number. However, the latent factors number is quite likely to be very large in reality. To tackle this issue, we propose to solve the following sparse collaborative filtering problem,

$$\sum_{(a,b) \in A} l(r_{a,b}, u_a, v_b) + \frac{\lambda}{2} \left(\sum_{a=1}^{m} \|u_a\|^2 + \sum_{b=1}^{n} \|v_b\|^2 \right) + \tau \left(\sum_{a=1}^{m} \|u_a\|_1 + \sum_{b=1}^{n} \|v_b\|_1 \right)$$

Furthermore, since the data usually comes one by one, we propose to solve the sparse collaborative filtering problems through online learning techniques. Specifically, we will update the two vectors as follows:

$$\boldsymbol{u}_a \leftarrow arg \min_{\boldsymbol{u}} \langle \partial_u l(r_{a,b}, \boldsymbol{u}_a, \boldsymbol{v}_b) + \lambda \boldsymbol{u}_a, \boldsymbol{u} \rangle + \tau \|\boldsymbol{u}\|_1 + \frac{1}{2\eta_t} \|\boldsymbol{u} - \boldsymbol{u}_a\|^2$$

And

$$\boldsymbol{v}_b \leftarrow arg \min_{\boldsymbol{v}} \langle \partial_v l(r_{a,b}, \boldsymbol{u}_a, \boldsymbol{v}_b) + \lambda \boldsymbol{v}_b, \boldsymbol{v} \rangle + \tau \|\boldsymbol{v}\|_1 + \frac{1}{2\eta_t} \|\boldsymbol{v} - \boldsymbol{v}_b\|^2$$

These two updates enjoys closed-form solutions:

$$\boldsymbol{u}_a \leftarrow ST_{\tau\eta t}[(1-\eta_t\lambda)\boldsymbol{u}_a - \eta_t \partial_{\boldsymbol{u}} l(r_{a,b}, \boldsymbol{u}_a, \boldsymbol{v}_b)] \tag{1}$$

And

$$\boldsymbol{v}_b \leftarrow ST_{\tau\eta t}[(1-\eta_t\lambda)\boldsymbol{v}_b - \eta_t \partial_{\boldsymbol{v}} l(r_{a,b}, \boldsymbol{u}_a, \boldsymbol{v}_b)] \tag{2}$$

Where $ST_v(w) = sign(w) \odot [|w| - v]_+$ and $\odot$ denotes element-wise product.

3.2 Second Online Sparse Collaborative Filtering Algorithm

In order to improve the convergence speed, we propose a second order online collaborative filtering algorithm. The key idea of SOCFII is to not only update the user and item weight vectors at each round, but also estimate their distribution, i.e., mean and covariance matrix. In the second order online collaborative filtering, where u_a and v_b are assumed satisfy Gaussian distributions. The objective functions are

$$D_{KL}\left(N(\mu_{\boldsymbol{u}_a}, \sum\nolimits_{\boldsymbol{u}_a}) \| N(\mu_{\boldsymbol{u}_a,t}, \sum\nolimits_{\boldsymbol{u}_a,t})\right) + \eta l(r_{a,b}, \boldsymbol{u}_a, \boldsymbol{v}_b) + \frac{\lambda}{2}\boldsymbol{v}_b^T \sum\nolimits_{\boldsymbol{u}_a} \boldsymbol{v}_b$$

And

$$D_{KL}\left(N(\mu_{\boldsymbol{v}}, \sum\nolimits_{\boldsymbol{v}_b}) \| N(\mu_{\boldsymbol{v}_a,t}, \sum\nolimits_{\boldsymbol{v}_b,t})\right) + \eta l(r_{a,b}, \boldsymbol{u}_a, \boldsymbol{v}_b) + \frac{\lambda}{2}\boldsymbol{u}_a^T \sum\nolimits_{\boldsymbol{v}_b} \boldsymbol{u}_a$$

Where K_{KL} is KL divergence.

In this way, the algorithm significantly outperform first order algorithm. However, the latent number k has to be set as a small value. To solve this issue we proposed the following two online objective functions:

$$\begin{aligned} C_{\boldsymbol{u}_a}(\mu_{\boldsymbol{u}_a}, \sum\nolimits_{\boldsymbol{u}_a}) = {} & D_{KL}\left(N(\mu_{\boldsymbol{u}_a}, \sum\nolimits_{\boldsymbol{u}_a}) \| N(\mu_{\boldsymbol{u}_a,t}, \sum\nolimits_{\boldsymbol{u}_a,t})\right) + \eta\langle \partial_{\boldsymbol{u}} l(r_{a,b}, \mu_{\boldsymbol{u}_a,t}, \boldsymbol{v}_b), \mu_{\boldsymbol{u}_a}\rangle \\ & + \frac{\lambda}{2}\boldsymbol{v}_b^T \sum\nolimits_{\boldsymbol{u}_a} \boldsymbol{v}_b + \eta\tau\|\mu_{\boldsymbol{u}_a}\|_1 \end{aligned}$$

And

$$\begin{aligned} C_{\boldsymbol{v}_b}(\mu_{\boldsymbol{v}_b}, \sum\nolimits_{\boldsymbol{v}_b}) = {} & D_{KL}\left(N(\mu_{\boldsymbol{v}_b}, \sum\nolimits_{\boldsymbol{v}_b}) \| N(\mu_{\boldsymbol{v}_b,t}, \sum\nolimits_{\boldsymbol{v}_a,t})\right) + \eta\langle \partial_{\boldsymbol{v}} l(r_{a,b}, \mu_{\boldsymbol{v}_b,t}, \boldsymbol{v}_b), \mu_{\boldsymbol{v}_b}\rangle \\ & + \frac{\lambda}{2}\boldsymbol{u}_a^T \sum\nolimits_{\boldsymbol{v}_b} \boldsymbol{u}_a + \eta\tau\|\mu_{\boldsymbol{v}_b}\|_1 \end{aligned}$$

These objectives linearize the loss functions and introduce sparsity regularization. We can solve these two objectives in two steps:

$$\begin{aligned} \mu_{\boldsymbol{u}_a,t+1} &= arg\min_{\mu_{\boldsymbol{u}_a}} C_{\boldsymbol{u}_a}(\mu_{\boldsymbol{u}_a}, \sum\nolimits_{\boldsymbol{u}_a}) \\ &= arg\min_{\mu_{\boldsymbol{u}_a}} \frac{1}{2}(\mu_{\boldsymbol{u}_a} - \mu_{\boldsymbol{u}_a}, t)^T \sum\nolimits^{-1}_{\mu_{\boldsymbol{u}_a,t}} (\mu_{\boldsymbol{u}_a} - \mu_{\boldsymbol{u}_a,t}) \\ &\quad + \eta\langle \partial_{\boldsymbol{u}} l(r_{a,b}, \mu_{\boldsymbol{u}_a,t}, \boldsymbol{v}_b), \mu_{\boldsymbol{u}_a}\rangle + \eta\tau\|\mu_{\boldsymbol{u}_a}\|_1 \end{aligned}$$

$$\begin{aligned} \sum\nolimits_{\boldsymbol{u}_a,t+1} &= arg\min_{\sum_{\boldsymbol{u}_a}} C_{\boldsymbol{u}_a}(\mu_{\boldsymbol{u}_a}, \sum\nolimits_{\boldsymbol{u}_a}) \\ &\text{i.e.,} \sum\nolimits_{\boldsymbol{u}_a,t+1} = \sum_{\boldsymbol{u}_a,t} - \frac{\sum_{\boldsymbol{u}_a,t} \boldsymbol{v}_b \boldsymbol{v}_b^T \sum_{\boldsymbol{u}_a,t}}{1/\lambda + \boldsymbol{v}_b^T \sum_{\boldsymbol{u}_a,t} \boldsymbol{v}_b} \end{aligned} \tag{3}$$

And

$$\begin{aligned}\mu_{\boldsymbol{v}_b,t+1} &= arg\min_{\mu_{\boldsymbol{v}_b}} C_{\boldsymbol{v}_b}(\mu_{\boldsymbol{v}_b}, \sum\nolimits_{\boldsymbol{v}_b}) \\ &= arg\min_{\mu_{\boldsymbol{v}_b}} \frac{1}{2}(\mu_{\boldsymbol{v}_b} - \mu_{\boldsymbol{v}_b,t})^T \sum\nolimits^{-1}_{\mu_{\boldsymbol{v}_b,t}} (\mu_{\boldsymbol{v}_b} - \mu_{\boldsymbol{v}_b,t}) \\ &= \ + \eta\langle \partial_{\boldsymbol{v}} l(r_{a,b}, \mu_{\boldsymbol{v}_b,t}, \boldsymbol{v}_b), \mu_{\boldsymbol{v}_b}\rangle + \eta\tau\|\mu_{\boldsymbol{v}_b}\|_1\end{aligned}$$

$$\begin{aligned}\sum\nolimits_{\boldsymbol{v}_b,t+1} &= arg\min_{\sum_{\boldsymbol{v}_b}} C_{\boldsymbol{v}_b}(\mu_{\boldsymbol{v}_b}, \sum\nolimits_{\boldsymbol{v}_b}) \\ \text{i.e.,} \sum\nolimits_{\boldsymbol{v}_b,t+1} &= \sum_{\boldsymbol{v}_b,t} - \frac{\sum_{\boldsymbol{v}_b,t} \boldsymbol{u}_a \boldsymbol{u}_a^T \sum_{\boldsymbol{v}_b,t}}{1/\lambda + \boldsymbol{v}_b^T \sum_{\boldsymbol{v}_b,t} \boldsymbol{v}_b}\end{aligned} \tag{4}$$

In practice, it is computationally expensive to get $\mu_{\boldsymbol{u}_a,t+1}$, $\mu_{\boldsymbol{v}_b,t+1}$ and update $\sum_{\boldsymbol{u}_a,t+1}$, $\sum_{\boldsymbol{v}_b,t+1}$. To solve this issue, we can set the covariance matrices as diagonal, which will produce the following closed-form solutions for the two mean vectors:

$$(\mu_{\boldsymbol{u}_a,t+1})_i = ST_{\eta\tau(\sum_{\boldsymbol{u}_a,t})_{i,i}}\left[(\mu_{\boldsymbol{u}_a,t})_i - \eta(\sum\nolimits_{\boldsymbol{u}_a,t})_{i,i}(\partial_{\boldsymbol{u}} l(r_{a,b}, \mu_{\boldsymbol{u}_a,t}, \boldsymbol{v}_b))_i\right] \tag{5}$$

And

$$(\mu_{\boldsymbol{v}_b,t+1})_i = ST_{\eta\tau(\sum_{\boldsymbol{v}_b,t})_{i,i}}\left[(\mu_{\boldsymbol{v}_b,t})_i - \eta(\sum\nolimits_{\boldsymbol{v}_b,t})_{i,i}(\partial_{\boldsymbol{v}} l(r_{a,b}, \mu_{\boldsymbol{v}_b,t}, \boldsymbol{v}_b))_i\right] \tag{6}$$

Algorithm 1 First Order Sparse Online Collaborative Filtering (SOCFI)

Require: a sequence of rating pairs $\{(a_t, b_t, r_{ab}), t = 1, \cdots, T\}$

1: Initialization: initialize a random matrix for user matrix $U \in R^{n\times k}$ and item matrix $V \in R^{m\times k}$
2: **for** $t = 1, 2\cdots, T$ **do**
3: Receive rating prediction request of user a_t on item b_t
4: Make prediction $\widehat{r_{a_t b_t}} = U_{a_t} V_{b_t}^T$
5: The true rating $r_{a_t b_t}$ is revealed
6: The algorithm suffers a loss $l(U_a, V_b, r_{a,b})$
7: Update U_{a_t} and V_{b_t} by (1), (2), respectively
8: **end for**

4 Experiment

In this section, we present the experimental results to evaluate the performance of our proposed methods by using online the Root Square Error (RMSE) on the data set.

Our experiments are performed on two real data sets: MovieLens100k and MovieLens1M. These two data sets are classic movie rating data sets collected by the MovieLens web site (www.movielens.umn.edu). MovieLens is publicly available data set, and it is widely used to study recommendation systems. The MovieLens100K consists of 100,000 ratings from 943 users on 1,682 movies and the MovieLens1M consists of 1,000,209 ratings from 6,040 users on 3,900 movies.

Algorithm 2 Second Order Sparse Online Collaborative Filtering (SOCFII)

Require: a sequence of rating pairs $\{(a_t, b_t, r_{ab}), t = 1, \cdots, T\}$

1: Initialization: initialize a random matrix for user matrix $U \in R^{n\times k}$ and item matrix $V \in R^{m\times k}$, and covariance matrix $\sum_U, \sum_V$ to be item I
2: **for** $t = 1, 2 \cdots, T$ **do**
3: Receive rating prediction request of user a_t on item b_t
4: Make prediction $\widehat{r_{a_t b_t}} = U_{a_t} V_{b_t}^T$
5: The true rating $r_{a_t b_t}$ is revealed
6: The algorithm suffers a loss $l(U_a, V_b, r_{a,b})$
7: Update U_{a_t} and V_{b_t} by (3), (4), respectively
8: Update $\sum_{U_a}$ and $\sum_{V_b}$ by (5), (6), respectively
9: **end for**

4.1 Compared Algorithm

We compare our methods with other two online algorithm CF algorithms as follow:

(1) OCF: Online Collaborative Filtering for learning a rank-k matrix factorization by using online gradient descent [3];

(2) DA-OCF: Dual-Averaging method of probabilistic matrix factorization for Online Collaborative Filtering by absorbing previous rating information in an approximate average gradient of the loss [9];

(3) SOCFI: First Order Sparse Online Collaborative Filtering;

(4) SOCFII: Second Order Sparse Online Collaborative Filtering by setting covariance matrices as diagonal to simplify the calculation.

Our experiments are conducted on two real data sets, i.e. MovieLens100k and MovieLens1M, which are classic movie rating data sets collected by the MovieLens web site (www.movielens.umn.edu). MovieLens, as a publicly available data set, is widely used to study recommendation systems. The MovieLens100K consists of 100,000 ratings from 943 users on 1,682 movies while the MovieLens1M consists of 1,000,209 ratings from 6,040 users on 3,900 movies.

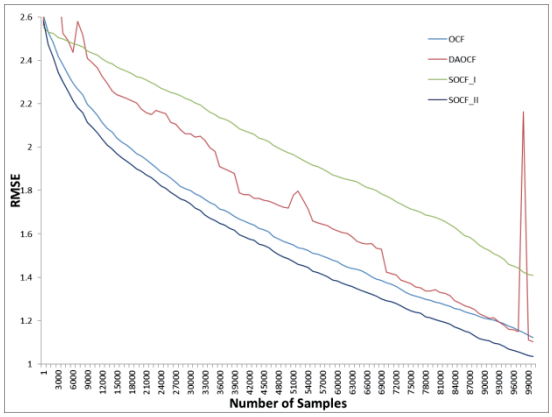
The rank parameter k of matrix u and v is set to four cases: 5, 10 and 50, respectively. After all the parameters are set, all the experiments are conducted 10 times randomly for each data set. To make a fair comparison, the learn arte r of all algorithms is set to 0.01, and λ parameter in DAOCF is set to 0.006, which was suggested to achieve the best performance according to reference [31].

4.2 Experiment and analysis

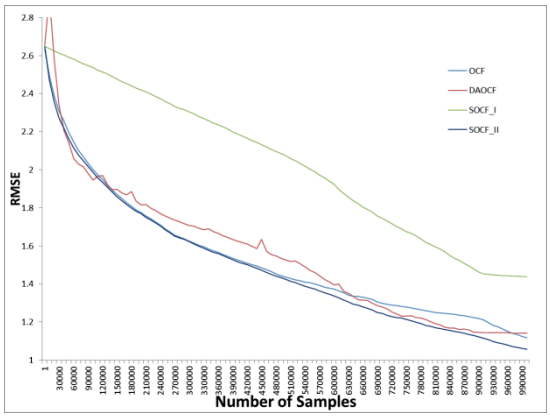
Table 1: The results of MovieLens100K

MovieLens100K	k=5	k=10	k=50
	RMSE	RMSE	RMSE
OCF	1.1218	0.9904	1.5007
DAOCF	1.1038	1.0882	**1.2654**
SOCFI	1.4083	1.2502	1.2886
SOCFII	**1.0355**	**0.9859**	1.3191

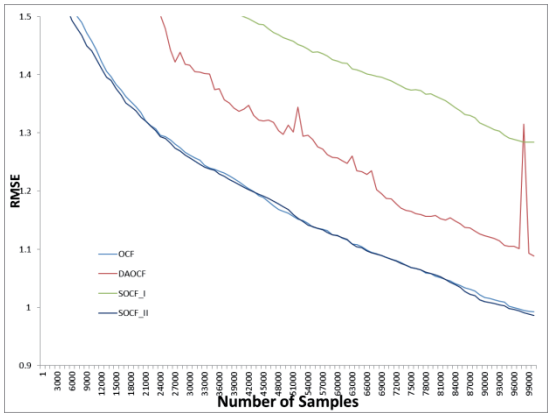
For performance metric, we evaluate the performance of online collaborative filtering algorithms by measuring their scores of online Root Square Error (RMSE) on the test set. The



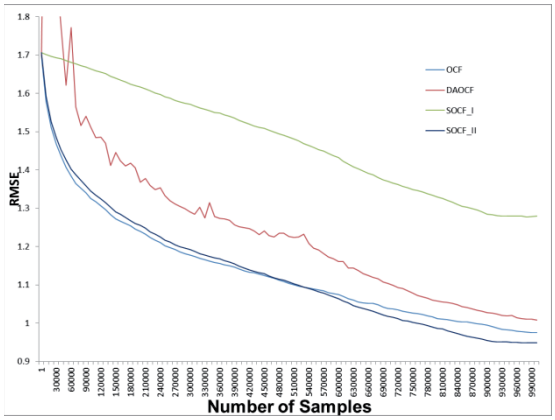
(a) MovieLens100K, K=5



(b) MovieLens1M, K=5



(c) MovieLens100K, K=10



(d) MovieLens1M, K=10

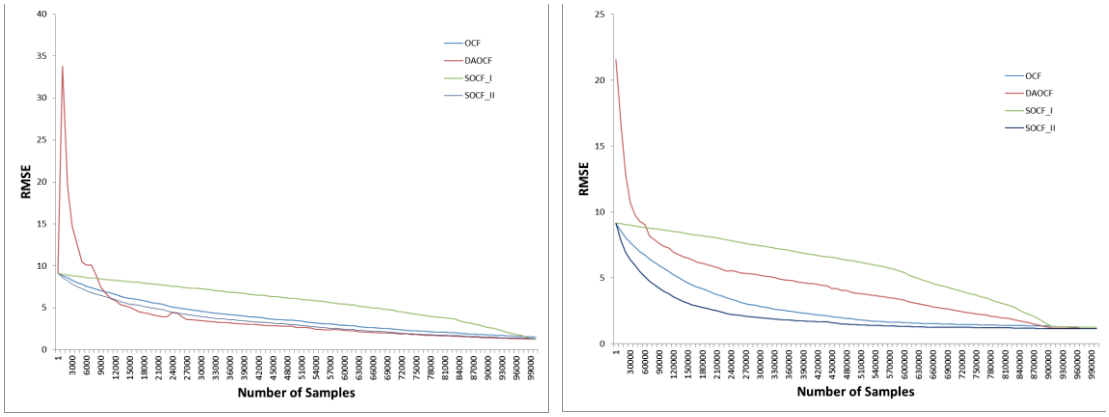
average performance of all approaches is shown in Table 1.

Table 2: The results of MovieLens1M

MovieLens1M	k=5	k=10	k=50
	RMSE	RMSE	RMSE
OCF	1.1147	0.9769	1.2345
DAOCF	**1.0136**	1.0077	1.1286
SOCFI	1.4201	1.2702	1.2577
SOCFII	**1.0573**	**0.9486**	**1.1179**

By comparison with OCF and DAOCF approaches, we can find from table 1 that the SOCFII always achieves best results on these data sets, while has smaller RSME values in all cases.

When k is set to a small value, such as 5 or 10, both of the traditional algorithm DAOCF and the SOCFII we proposed perform very well in small scale data sets and large scale data sets. In small scale data sets in which k is set to a large value, such as 50, the DAOCF works better than SOCFII. However, when the data sets become large ones, SOCFII achieves best results. Due to the SOCFI algorithm's lost on second order information, the SOCFI may not perform very well in the case of a very small k on both small scale data sets and large scale data sets. However, in reality, the latent factors number k stands a good chance to be very large. When we turn the k into a large value, adapting to the real situation, the SOCFI achieves better results on these



(e) MovieLens100k, K=50 (f) MovieLens1M, K=50

Figure 1: Performance of online collaborative filtering approaches on the MovieLens100k and MovieLens1M data sets.

data sets. This shows that our proposed method which exploits the confidence information can better handle data online and is suitable to large-scale dynamic collaborative filtering scenario.

To further evaluate our approaches for online learning, Figure 1 shows more details. We observe that the curve of the DAOCF is always fluctuant instead of smooth. This is disadvantageous for stable output, because sometimes users will be recommended with item which involve severe and numerous errors. The curves of SOCFII and OCF are very similar. However, the former is slightly better than the latter.

5 Conclusion and Future Work

In this paper, we propose two approaches to deal with the user-item ratings for online collaborative filtering. We focus on the online matrix factorization problem which consists of learning the basis set of users in order to adapt it to online CF recommender systems. A user's interest and an item's popularity are always changing over a long period of time. So, recommendation approaches should take such changes into consideration.

Then, an empirical study has been conducted on two benchmark data sets, namely, *MovieLens100K* and *MovieLens1M*. These experimental results demonstrate that our online algorithm achieves more accuracy performance than other online algorithms while dramatically boosting efficiency. Our approaches are suitable to large-scale dynamic collaborative filtering scenario.

Now, many collaborative filtering algorithms are unable to capture the latest change of user preferences over time. In the future, we will focus our works on the improving of prediction accuracy and the accelerating of the speed of our approaches.

Acknowledgment

The Project was supported by the National Natural Science Foundation of China (No. 61402386, No. 61305061 and Grant No. 61402389). And we wish to thank the anonymous reviewers who helped to improve the quality of the paper.

Bibliography

[1] G. Linden, B. Smith, and J. York (2003), Amazon. com recommendations: Item-to-item collaborative filtering, *Internet Comput. IEEE*, 7(1): 76–80.

[2] M. D. Ekstrand, R. Join T., and K. Joseph A.(2011), Collaborative Filtering Recommender Systems, *Found. Trends® Human–Computer Interact.*, 4(2): 81–173.

[3] Z. Wang and H. Lu (2014), Online Recommender System Based on Social Network Regularization, *Neural Inf. Process.*, 487–494, Nov. 2014.

[4] J. B. Schafer, J. Konstan, and J. Riedl (1999), Recommender systems in e-commerce, *Electronic Commerce*, 158–166.

[5] K. Dohyun and Y. Bong Jin (2005), Collaborative filtering based on iterative principal component analysis, *Expert Syst. Appl.*, 28(4): 823–830.

[6] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl (2004), Evaluating collaborative filtering recommender systems, *ACM Trans. Inf. Syst. TOIS*, 22(1): 5–53, 2004.

[7] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl (2001), Item-based collaborative filtering recommendation algorithms, *Proceedings of the 10th international conference on World Wide Web*, 285–295.

[8] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl (1997), GroupLens: applying collaborative filtering to Usenet news, *Commun. ACM*, 40(3): 77–87.

[9] J. Wang, S. C. H. Hoi, P. Zhao, and Z.-Y. Liu (2013), Online multi-task collaborative filtering for on-the-fly recommender systems, *Proceedings of the 7th ACM conference on Recommender systems, 2013*, 237–244.

[10] K. Yehuda and I. Haifa (2010), Collaborative filtering with temporal dynamics, *Commun. ACM*, 53(4): 89–97.

[11] J. Z. Kolter and M. Maloof (2003), Dynamic weighted majority: a new ensemble method for tracking concept drift, *ICDM 2003. Third IEEE International Conference on*, 123–130.

[12] D. Wang, P. Wu, P. Zhao, Y. Wu, C. Miao, and S. C. H. Hoi (2014), High-Dimensional Data Stream Classification via Sparse Online Learning, *Data Mining (ICDM), 2014 IEEE International Conference on*, 1007–1012.

[13] J. Abernethy, K. Canini, J. Langford, and A. Simma (2007), *Online collaborative filtering*, Univ. Calif. Berkeley Tech Rep, 2007.

[14] M. Ali, C. C. Campbell, and A. K. Tang (2011), Parallel Collaborative Filtering for Streaming Data, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.230.8613.

[15] J. Wilson, S. Chaudhury, B. Lall, and P. Kapadia (2014), Improving Collaborative Filtering based Recommenders using Topic Modelling, *Web Intelligence*, 340–346.

[16] T. Hofmann (2004), Latent semantic models for collaborative filtering, *ACM Trans. Inf. Syst.*, v 22(1): 89–115.

[17] R. Salakhutdinov, A. Mnih, and G. Hinton (2007), Restricted Boltzmann machines for collaborative filtering, *Proceedings of the 24th international conference on Machine learning*, 791–798.

[18] Li M; Wu C; Zhang L; You LN (2015), An Intuitionistic Fuzzy-Todim Method To Solve Distributor Evaluation And Selection Problem, *International Journal Of Simulation Modelling*, 14(3): 511-524.

[19] A. Mnih and R. Salakhutdinov (2007), Probabilistic matrix factorization, *Advances in neural information processing systems*, 1257–1264.

[20] G. Rainer, N. Nrik, H. Peter J., and S. Yannis (2011), Large-scale matrix factorization with distributed stochastic gradient descent, *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 69–77.

[21] Saric T; Simunovic G; Simunovic K (2013), Use Of Neural Networks In Prediction And Simulation Of Steel Surface Roughness, *International Journal Of Simulation Modelling*, 12(4): 225-236.

[22] R. M. Bell, Y. Koren, and C. Volinsky (2007), The BellKor solution to the Netflix Prize, http://www.netflixprize.com/assets/ProgressPrize2007_KorBell.pdf, 1-15.

[23] J. Bennett and S. Lanning (2007), The Netflix Prize, *KDD Cup Workshop Conjunction KDD*, 2007.

[24] Z. Qiao, P. Zhang, J. He, Y. Cao, C. Zhou, and L. Guo (2014), Combining geographical information of users and content of items for accurate rating prediction, *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, 361–362.

[25] W. Li and D. Yeung (2011), Social Relations Model for Collaborative Filtering, *Twenty-Fifth AAAI Conference on Artificial Intelligence*, 803-808.

[26] Ocevcic Hrvoje; Nenadic Kresimir; Solic Kresimir (2014), Decision Support Based On The Risk Assessment Of Information Systems And Bayesian Learning, *Tehnicki Vjesnik-Technical Gazette*, 21(3): 539-544.

[27] Y. Koren, R. Bell, and C. Volinsky (2009), Matrix factorization techniques for recommender systems, *Computer*, 8: 30–37.

[28] M. Julien, B. Francis, P. Jean, and S. Guillermo (2010), Online Learning for Matrix Factorization and Sparse Coding, *J. Mach. Learn. Res.*, 11: 19–60.

[29] S. Shalev-Shwartz (2011), Online Learning and Online Convex Optimization, *Found. Trends Mach. Learn.*, 4(2): 107–194.

[30] R. Pálovics, A. A. Benczúr, L. Kocsis, T. Kiss, and E. Frigó (2014), Exploiting temporal influence in online recommendation, 273–280.

[31] G. Ling, H. Yang, I. King, and M. R. Lyu (2012), Online Learning for Collaborative Filtering, *IEEE World Congress on Computational Intelligence*, Brisbane, Australia, 1 – 8.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):259-272, April 2016.

A Multi-Objective Approach for a Multihoming Load Balancing Scheme in WHN

C. Lozano-Garzon, M. Molina, Y. Donoso

Carlos Lozano-Garzon*
Universidad de los Andes, Bogotá, Colombia, South America
& Universitat de Girona, Girona, Spain.
*Corresponding author: calozanog@ieee.org

Miguel Molina, Yezid Donoso
Universidad de los Andes
Bogotá, Colombia, South America
mf.molina35@uniandes.edu.co, ydonoso@uniandes.edu.co

Abstract: The telco operators face up to challenges related to the need of ensuring a quality of service to the user in a planning, maintenance and resource allocation in their complex networks. These challenges are directly related with the need to ensure an user's service with a good level of quality in a highly dynamic environment in terms of changes in the radio access technologies, growth in the number of mobile users, technical requirements of the new services and applications, and the possibility to connect to different networks at the same time, among others. In this paper, we address the problem of the user's service allocation into the different feasible networks in order to reduce the network overloading. We present a multihoming load balancing scheme that allows the re-allocation of services according to their QoS requirements and the availability of network resources. We propose a multi-objective optimization model of this problem together with an evolutionary algorithm to solve it. Through simulation in different scenarios, we show that our algorithm is efficient, sensitive, scalable and provides optimal solutions.
Keywords: Heterogeneous Networks, load Balancing, multihoming, multi-objective optimization, multi-objective evolutionary algorithms, vertical handover.

1 Introduction

Given the continuous advances in network technologies, the growth in the number of mobile users and the increasing demand of the new services and applications, the mobile network operator are confronted with multiple challenges in the planning, maintenance and operation of their complex infrastructure. In many cases, these networks are composed by multiple radio access technologies, which allows to the user the access to different services by using simultaneously one or more their network interfaces.

In some cases during the network operation, it is possible that some radio access channels could be overloaded because of a traffic growth caused by the increased of the number of user's services connected to these channels. Therefore, the mobile operators need some mechanisms that allows a balanced distribution of the traffic load over the available networks. Note that this process could involve could involve the reallocation of some users. The main goal of this mechanism is the optimal use of the available network resources whilst the technical requirements needs of each service are guaranteed.

From the viewpoint of the mobile network operators, one of the most appropriate ways to achieve this goal is that the deployed infrastructure must be capable to perform the user's reallocation for one network to another, this process is called Vertical Handover (VHO). Due to standards such as IEEE 802.21 [1] only provide the framework for VHO, the decision making

algorithm that allow establish the best connection for each user is an open challenge [2]. In this algorithms, the re-allocation must be based on different metrics obtained from the mobile devices and/or from the performance parameters of the available networks [3]. Also, it is important that this network changes process should be transparent for the user.

Based on the recent advances in the mobile phones, nowadays these devices could establish connections to multiple networks in a simultaneously way, which is referred to as multihoming [4]. This characteristic facilitates a seamless VHO process while it is seamless to the user [5] and allows the simultaneous transmission of multiple services across multiple network interfaces [6]. Several research projects use the multihoming strategy over a heterogeneous environment in order to achieve a load balancing [4]; [7–9] or to make a better distribution of the bandwidth charge [10,11]. In other studies this strategy was used as a decision tool for the VHO process [12–15].

Considering the aforementioned problems, we addressed our study about the Always Best Connected (ABC) problem in heterogeneous wireless networks (HWN) in [16]. In this work we designed a proactive Vertical Handover Decision Algorithm (VHO-DA) based on user preferences, QoS requirements, and network conditions. Later in [17], we presents a load balancing optimization scheme; this scheme is composed by one mathematical model and a two-step algorithm based on the anchor-adjustment heuristic. In this paper, we also address the problem of load balancing across heterogeneous networks from the viewpoint of the operator. We present a multi-objective optimization model to solve the traffic load balancing problem into HWN using a multihoming strategy, and an evolutionary algorithm to solve it.

The remainder of this paper is structured as follows. In Section 2 we introduce the mathematical model that encodes the multi-objective function in order to obtain a global load balancing among HWN. In section 3 the load balancing algorithm based on strength pareto evolutionary algorithm is presented. The experimental results about the performance of our proposal are shown in Section 4. Finally, concluding remarks and directions for further research are given in Section 5.

2 Load Balancing Mathematical Model

As it was mentioned in [17], the load balancing is an important strategy used by the mobile operators in order to allocate, in a fair way, the available resources in a network. However, this strategy implies, in many cases, the reallocation of mobile devices; therefore, it is necessary to consider the cost of connecting services to the new networks and the energy consumption of the mobile device. Considering the above statements and the possibility of the simultaneous use of multiple network interfaces by each mobile device [6], in this section we proposed a multi-objective - multihoming mathematical model.

2.1 Decision Variable

By assuming that mobile devices are able to perform multihoming in the network, we define the decision variable x as a binary variable that specify if the service s of the mobile k is connected to the network j or not (See Figure 1).

The variable is represented as follows:

$$x_j^{k,s} = \begin{cases} 1 & \text{if the service is connected to the network} \\ 0 & \text{otherwise} \end{cases}$$

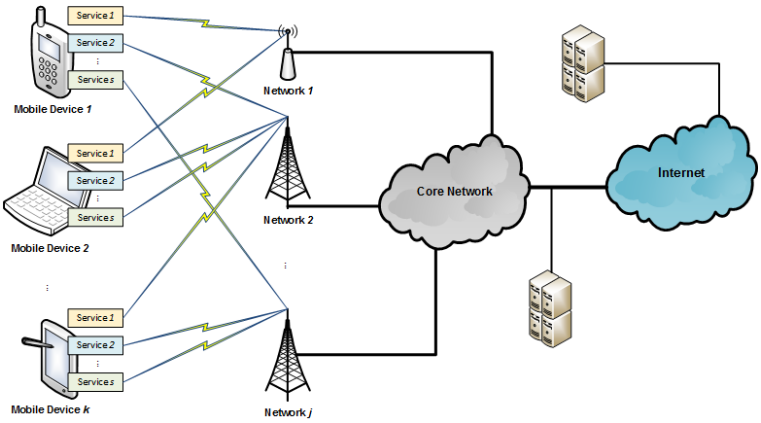


Figure 1: Multihoming Cellular System

2.2 Objective Functions

In order to design an efficient load balancing algorithm, the mathematical model is formulated under the premise of achieving an overall load balance in the wireless heterogeneous networks, whilst the connection cost and the energy consumption is also minimized. This function is expressed as: $min(\alpha, \beta, \gamma)$, where α represents the load balancing function, β the connection cost function, and γ the energy consumption function.

Load Balancing Function

The load balancing function (α) is the main function of this model. This function determines the network traffic load by considering the demand of the services of the mobile devices in relation to the theoretical available bandwidth of the network. For this model, the load function is defined as: $\alpha = max(\alpha_j), \forall j \in N$, where j represents the destination network of the mobile device, N the set of access networks and α_j the load of the network j.

We calculate α_j as the sum of demanded bandwidth (D_s) of each connected service (s), for each mobile (k) over the theoretical available bandwidth of the network (BW_j)

$$\alpha_j = \frac{\sum_k \sum_s D_s \cdot x_j^{k,s}}{BW_j}, \forall j$$

Connection Cost Function

The function determines the maximum monetary cost of the mobile devices that are connected to the network. If a mobile device has at least one service connected to the network, its cost is taken into account to access the network. The connection cost function is expressed as: $\beta = max(\beta_j), \forall j \in N$, where j represents the network and β_j the maximum cost of connected mobile devices to this network ($Cost_j$).

β_j is defined as:

$$\beta_j = \sum_k max_s(Cost_j \cdot x_j^{k,s}), \forall j$$

Energy Consumption Function

This function determines the energy consumption of the mobile devices that are connected to the network. If a mobile device has at least one service connected to the network, the consumption that generates for being connected to the network is taken into account in our model. We defined the energy function as: $\gamma = max(\gamma_j), \forall j \in N$.

Whilst γ_j is determined as the maximum consumption that is generated by mobile device k for being connected to the j_{th} network

$$\gamma_j = \sum_k max_s\{Cons(RSS_{k,j}) \cdot x_j^{k,s}\}, \forall j$$

Finally, in order to guarantee consistency in our model, we consider that when a services s is active in the device, it will generate a traffic demand D_s. Following the work presented in [18], the values of received signal strength are discretized at three levels: low, medium and high. Note that the power consumption is inversely proportional to the Received Signal Strength (RSS) and therefore, a high signal level results in a low power consumption by the mobile radio interface through which communication is established.

Because multiple network interfaces can be used in a multihoming scenario, it is a privilege to be connected to those network which you receive better signals from, i.e. where less energy is consumed for being connected. Consumption levels derived from the RSS are modeled in the $Cons(RSS_{k,j})$ function.

$$Cons(RSS_{k,j}) = \begin{cases} 1 & RSS_{k,j} > RSS_{th2} \\ 2 & RSS_{th1} \leq RSS_{k,j} < RSS_{th2} \\ 3 & 0 < RSS_{k,j} < RSS_{th1} \\ 0 & RSS_{k,j} = 0 \end{cases}$$

2.3 Model Constraints

Through the model constraints, we intend to guarantee the adjustment of this model to the real-life Telco networks. In this case, the model only allows the service connection to those networks that are in the coverage area, comply with the cost the user can assume to connect to the network, and offer enough bandwidth to meet the demand of service and an adequate power consumption according to the level of battery charge that the mobile device has.

Demand Constraint

The demand constraint states that the service can be only connected to a network that has enough bandwidth to meet its demand.

$$D_s \cdot x_j^{k,s} \leq BW_j, \forall j$$

Cost Constraint

The cost constraint states that the overall cost to access network j ($Cost_j$), i.e. the cost to connect any service to network j must be less or equal to the cost incurred by the user in the contract of the mobile device k ($Cost_k$).

$$Cost_j \cdot x_j^{k,s} \leq Cost_k, \forall k$$

Access Constraint

Through the access constraint it is ensured that each service s that is active on the mobile device k is connected to a N network and can only be connected to one network.

$$\sum_j x_j^{k,s} = y_{k,s}, \forall k, \forall s$$

Reach Constraint

This constraint ensures that only networks that exceed the defined signal strength threshold (RSS_{th}) are considered in the model assessment.

$$RSS_{k,j} \geq RSS_{th}, \forall j, \forall k$$

Power Consumption Constraint

The power consumption constraint ensures that services only can be connected to those networks that offer lower power consumption, according to the current battery level of the mobile device.

$$Cons(RSS_{k,j}) \cdot x_j^{k,s} \leq Bat_k, \forall j, \forall k$$

where Bat_k is defined as:

$$Bat_k = \begin{cases} 1 & Charge_k < Bat_{th1} \\ 2 & Bat_{th2} \leq Charge_k \leq Bat_{th1} \\ 3 & Charge_k > Bat_{th2} \end{cases}$$

A service s of a mobile device k is considered active when the constraints are satisfied at least for one of the networks; i.e. if the service can be connected to at least one of the networks available for the device. When the service s is active, it consumes the traffic demand in the network that it is connected.

3 Multi-Objective Evolutionary Algorithm

In order to solve the multi-objective model, several strategies can be used. One of them is to evaluate each objective function in the model separately (mono-objective approach). The weight sum method is one of these strategies. However, it has several disadvantages, including not finding all optimal solutions if the solution set is not convex, and the need to normalize the functions so that no one predominates over the others [7].

For this reason we propose the use of an Multi-Objective Evolutionary Algorithm (MOEA) to find the best set of solutions for all objective functions at the same time. The chosen algorithm is the elitist type evolutionary algorithm SPEA (Strength Pareto Evolutionary Algorithm) proposed by Zitzler and Thiele [7,19]. The time complexity of this algorithm is upper bounded by $O(NM^2)$ in each generation, where M is the population size and N is the number of objectives.

Algorithm 1: SPEA algorithm pseudo-code

1: Generate a random population M
2: **while** not max number of generation **do**
3: Evaluate population according objective function
4: Calculate the fitness of each of the individuals
5: Classification Population based on fitness $(M, M\prime)$
6: Generate New Population M_{t+1}
7: Apply Binary Tournament Selection
8: Apply Crossover Operator
9: Mutation Operator
10: **end while**
11: Find Pareto Optimal Set.

3.1 Chromosome Representation

As starting point for implementing the evolutionary algorithm it is necessary to define the chromosome, i.e. the data representation of the solutions in the model. In the proposed mathematical model, the solutions can be expressed in a matrix representation (See Fig. 3). The rows represent mobile devices and the columns represent services; whilst the cell values represent the network in which the service will be connected, the value of zero is given when the service is not active and it is not connected to any network.

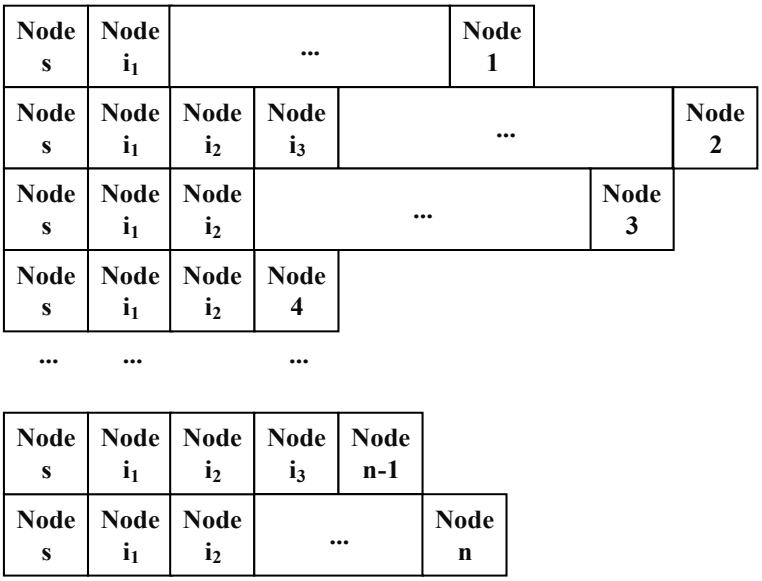
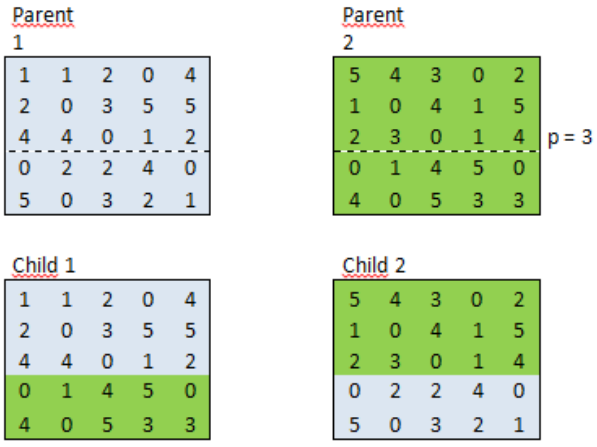


Figure 3: Chromosome representation of the solution

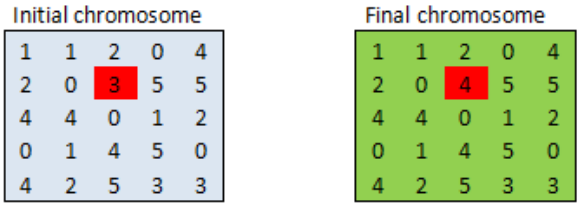
The decision variables of the mathematical model can be directly obtained from the chromosome, for example if service 2 of the mobile device 3 is connected to network 4, it means that $x_4^{3,2} = 1$ and for the remaining j networks the variable is 0.

3.2 Genetic Operators

The crossover function takes two initial solutions (i.e. chromosomes), called parent solutions, and then created a new one from them. We define a crossover function based on the well-known



(a) Crossover function



(b) Mutation function

Figure 4: Genetic Operators

single-point strategy proposed in [19]. The new solutions are generated by combining the first p rows of the first parent solution with the last $k - p$ rows of parent second parent solution, and vice-versa. The Figure 4(a) shows an example of this operator. Since the solution of each mobile device in the parent chromosomes meets the constraints of the model, the solutions of the mobile devices that are part of the child solutions also meet the constraints since each mobile device represented in a row conserves its signal strength and battery level conditions as it moves from one chromosome to another. Also note that the inactive services are the same in the different solutions so they are conserved.

The mutation strategy is based on a random function that takes a mobile device service from its current to a new one as long as the constrains are met, i.e. the network will be connected to a feasible network. The function takes into account only the services that are activated and have more than one feasible network. Figure 4(b) shows an example of the mutation strategy. In this case, the service 3 of the mobile 2 is moved from network 3 to network 4

3.3 Generation of Scenario Parameters

Once the genetic operators are defined, the algorithm starts to generate pseudo-randomly parameters for each mobile. This parameters are the maximum cost that the user can assume to connect to one network ($Cost_k$), the percentage values and battery charge levels (Bat_k and $Charge_k$), the signal strength values that each mobile perceives to each network j ($RSS_{k,j}$), with its corresponding power consumption level parameter ($Cons_{k,j}$).

When the scenario is created, we proceed to validate all constraints of model for each available network: demand, cost, reach and power consumption. After this validation, we store in a vector the networks that can be used to connect each service of each mobile device or the zero value ([0])

if the service cannot be connected to any network. The result is a feasibility matrix ($Fact_{k,s}$), this matrix is used to computed a matrix of active services $y_{k,s}$, where it is defined the probability of use for a mobile service, for each service that can be connected to at least one network. Once the active services are defined, a population of M initial solutions and an $M\prime$ elite (or external) population are randomly generated from both the feasibility and active services matrix. In our implementation, the value of M and $M\prime$ are set to 20 and 4, respectively.

4 Experimental Results

In order to verify the correct operation of the proposed algorithm, we define four different experimental scenarios. For each one of them, we execute 500 iteration of the algorithm with different number of networks, mobile devices and services.

4.1 First Experimental Scenario

The aim of this scenario is to compare the quality of the solution obtained by our algorithm with respect to the optimal solutions obtained by solving each function separately. The optimal solutions were computed using GAMS system [20]. For this purpose, the scenario is composed by five mobile devices that will connect three services in three different available access networks. Tables 1 and 2 describe the parameters of bandwidth, cost, distribution of active services and network coverage for each mobile device.

Table 1: Network parameters

Network Technology	Theoretical Bandwidth (Mbps)	$Cost_j$ (monetary units)
LTE	70	80
WiFi g	54	0
HSPA+	15	40

Table 2: Bandwidth demand parameters

Service	Voice	Video	Web
Demand (Mbps)	0.1	3	0.5

To solve a multi-objective mathematical model through a general optimization software such as GAMS [20], it is necessary to convert the optimization problem into a single objective one. In this case, each objective function was optimized separately obtaining the solutions shown in Table 3. On the other hand the proposed algorithm converges rapidly to a set of 4 optimal solutions; three of them are unique. The results are presented in Table 4. Based on the feasible solutions obtained, we plot three different graphics in order to identify the Pareto-Optimal Front. These graphics are: load (α) vs. cost (β), cost (β) vs. consumption (γ), and load (α) vs. consumption (γ), as you can see in Figure 4:(a). In the load (α) vs. cost (β) graph, the optimal Pareto front can be seen, because as the load decreases the cost grows. The cost (β) vs. consumption (γ) graph shows that solutions move in cost values of 80, 120 and 160, and zero-cost solution is quite atypical. Finally, in the load (α) vs. consumption (γ) graph can be seen that some optimal solutions are on value 4 and others on power consumption value 7.

Making a comparison between the two sets of solutions obtained (Table 3 and 4), the proposed algorithm found several intermediate solutions belonging to the Pareto optimal front. These solutions cannot be found under mono-objective approaches because they are clearly not a linear

Table 3: Mathematical model solutions obtained using GAMS

	LTE (load)	WiFi g (load)	HSPA+ (load)	α	β	γ
Solution 1	0.086	0.080	0.080	0.086	160	4
Solution 2	0	0.213	0	0.213	0	7
Solution 3	0.103	0.080	0	0.103	160	4

Table 4: Solution results for the first study case

	LTE (load)	WiFi g (load)	HSPA+ (load)	α	β	γ
Solution 1	0.051	0.115	0.113	0.115	120	7
Solution 2	0.051	0.124	0.080	0.124	80	4
Solution 3	0.086	0.080	0.080	0.086	160	4

combination of the objective functions. By looking at the solutions obtained by GAMS, it is possible to see that the solution 3 is dominated by solution 1, so that the solution 3 is not part of the Pareto optimal front. Moreover, it is highly improbable that the zero solution cost (found with GAMS) can be found by our algorithm, because this solution must connect all the services to the same network that goes against the main objective of our proposal.

Finally, in order to evaluate the quality of the solutions found by the algorithm, a performance metric called spacing was calculated. This metric, as its name implies, analyze the distribution of the solutions in a Pareto Front.

$$S = \sqrt{\frac{1}{|Q|} \cdot \sum_{i=1}^{|Q|} (d_i - \overline{d})^2}$$

where:

$$d_i = min_{k \in Q \wedge k \wedge \neq i} \sum_{m=1}^{M} |f_m^i - f_m^k|$$

$$\overline{d} = \sum_{i=1}^{|Q|} \frac{d_i}{|Q|}$$

It is important to note that d_i are the distance measure, $\overline{d}$ is the mean value of the above distance measure, and f_m^k is the m_{th} objective function value of the k_{th} member of the population [7].

For our implementation the value of S obtained was 21.505. This value was influenced in a strongest way by the difference in the cost values.

4.2 Second Experimental Scenario

We proceed to perform of the algorithm with the same external and elite population size $M = 20$, $M\prime = 4$; but changing to 20 mobile devices, 5 services for each device, and 7 access networks. This scenario was randomly generated according to the characteristics presented in Table 5 and 6. The feasible solutions obtained are presented in Table 7.

The solutions obtained show that the services present a tendency to be allocated into the networks with the highest capacity restricted only by the cost that each mobile could pay. The

Table 5: Network parameters

Network Technology	LTE	WiFi n	WiFi g	WiMAX	HSPA+	HSDPA	UMTS
Theoretical Bandwidth (Mbps)	70	300	54	15	15	2	0.3
$Cost_j$ (monetary units)	80	0	0	60	40	20	10

Table 6: Bandwidth demand of services for second study case

Service	Voice	Video	Web	Game	Chat
Demand (Mbps)	0.1	3	0.5	2	0.2

Table 7: Network parameters

	LTE	WiFi n	WiFi g	WiMAX	HSPA+	HSDPA	UMTS	α	β	γ
Solution 1	0.057	0.205	0.652	0.527	0.447	0.4	0.667	0.667	160	29
Solution 2	0.086	0.205	0.585	0.587	0.493	0.467	0.587	0.587	240	27

solutions obtained by our algorithm show that if the model constrains are met, then the algorithm trends to allocate the mobile services in those networks with the highest capacity and low cost

In addition, we plot three same graphics as the previous scenario (see Figure 5:(a)). In the first two graphs it is observed the optimal Pareto fronts. These first two graphs shows that in order to achieve a better load distribution across networks and to reach a lower power consumption of mobile devices, the services should be grouped into the higher-cost networks. However, it can be said that the load variations are little in comparison with the cost difference that can be obtained; so that the operator may prefer the solution which means lower cost. The value of the spacing metric (S) was 35.542, despite having three repeated solutions. The difference in the values of cost and load shows that the solutions are not equally distant.

4.3 Third Experimental Scenario

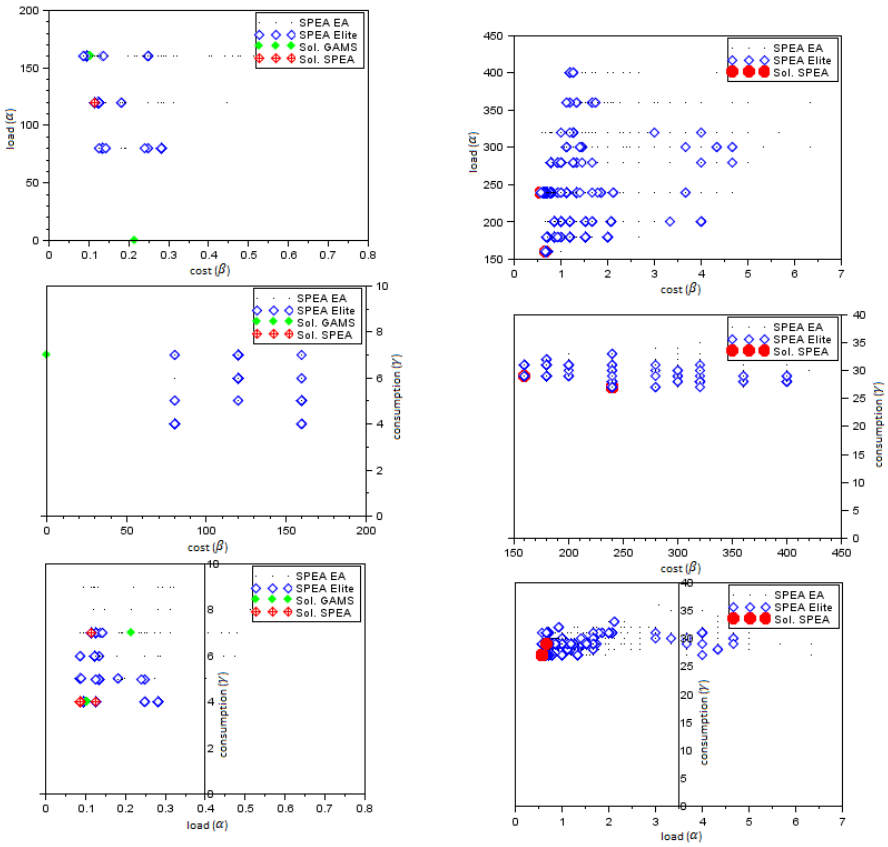
For this scenario, we only change the number of mobile devices to 500 in comparison with the latest scenario. The solutions obtained are shown in Table 8. Based on the achieved solutions for this scenario we can assert that our algorithm present a good grade of sensitivity. We can observe that small changes in the main parameters are reflected in the services allocation, which mean changes in the set of optimal solutions.

In Figure 6:(a) you can see, well defined, the optimal Pareto fronts in the first two graphs. This scenario also shows how small changes in load balancing produce appreciable changes in the cost function. The value of S for this scenario was 30.168, again due to the wide differences in cost values.

4.4 Fourth Experimental Scenario

In this last scenario we increased the number of mobile devices to 10000 in order to validate the scalability of the algorithm. The other scenario parameters are kept as the scenario 2. The feasible solutions are shown in Table 9.

Figure 6:(b) shows the optimal solutions found by our algorithm. It is possible to see that solutions trend to maintain the the same power consumption in both graphs, cost (β) vs. consumption (γ) and load (α) vs. consumption (γ). The S metric value was 1.0, so that the solutions are fairly well spaced, being an unique solutions. The algorithm is fully adaptable to any situation within the proposed mathematical model designed.



(a) First Study Case (b) Second Study Case

Figure 5: 2D perspectives of solution distribution

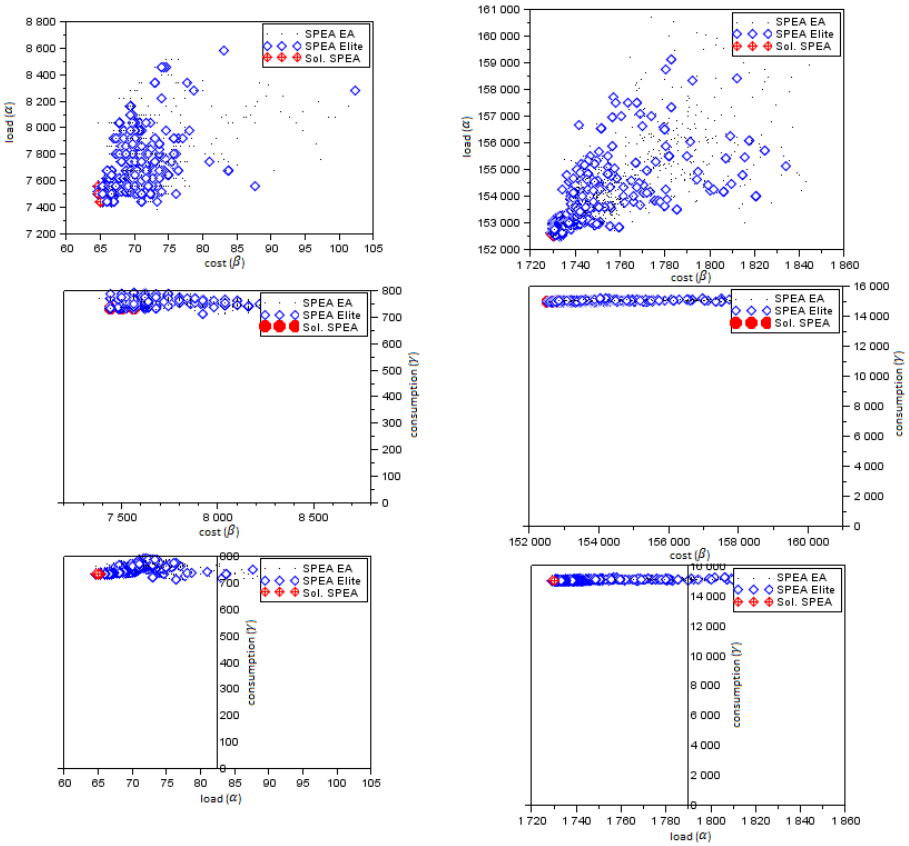
Table 8: Solution results for the third study case

	Solution 1	Solution 2	Solution 3	Solution 4
LTE	1.531	1.517	1.544	1.544
WiFi n	3.161	3.163	3.153	3.165
WiFi g	18.113	18.219	18.302	18.237
WiMAX	16.62	16.58	16.533	16.547
HSPA+	27.247	26.947	26.747	26.747
HSDPA	48.6	48.467	48.6	48.533
UMTS	64.667	64.667	65.333	65
α	64.667	64.667	65.333	65
β	7560	7500	7440	7440
γ	732	734	732	733

After running the algorithm on this scenario, we prove that the algorithm maintains its characteristics of high sensitivity in the searching for optimal solutions regardless of the number of mobile devices on the problem. That means our algorithm is scalable.

Table 9: Solution results for the fourth study case

	Solution 1	Solution 2	Solution 3	Solution 4
LTE	30.46	30.26	30.263	30.463
WiFi n	65.56	65.63	65.62	65.57
WiFi g	358.61	358.64	358.61	358.55
WiMAX	329.43	328.86	328.69	329.6
HSPA+	537.59	537.41	537.79	537.34
HSDPA	934.93	936.67	937.13	934.53
UMTS	1730	1730.33	1729	1731.33
α	1730	1730.33	1729	1731.33
β	152760	152520	152600	152680
γ	15024	15027	15027	15022



(a) First Study Case (b) Fourth Study Case

Figure 6: 2D perspectives of solution distribution

5 Conclusions and Future Work

We have presented a Multihoming Load Balancing Model in Heterogeneous Wireless Networks based on a multi-objective approach. In this model we take as objective functions the network load, connection cost, and energy consumption; with the aim of performing an efficient use of the capacity resources in the available networks.

Based on this model it was designed a Vertical Handover Algorithm (VHO) using evolutionary algorithms, specifically the Strength Pareto Evolutionary Algorithm (SPEA). Through the proposed environments we validate the correct operation of our algorithm; in the first scenario we validate the exactitude of the feasible solutions obtained by our algorithm in comparison to the solutions of the mathematical model using GAMS. In the other scenarios, we validate the sensibility and scalability of our evolutionary algorithm. The results obtained by our proposal were satisfactory and provided a starting point for the mobile network operator to run a VHO processes in their networks. With this process they could get an efficient use of their network resources, reduce the connection costs, and extend the battery life of mobile devices.

Due to we proposes a multi-objective optimization algorithm, the model is opens up to incorporate additional parameters as objective functions; these parameters could be obtained from both the available access networks and the mobile devices. As future work we propose to continue this research, we want to introduce the concept of fairness in the load balance optimization and also include the concept of quality of experience (QoE) in the objective functions.

6 Acknowledgment

The authors would like to thank to Administrative Department of Science, Technology and Innovation (COLCIENCIAS) for the financial support to Carlos Lozano-Garzon through the 528 - 2011 National Call for Doctoral Studies in Colombia

Bibliography

[1] IEEE Computer Society (2008); IEEE 802.21: Media Independent Handover Services, *IEEE-SA Standards Board* available at `https://standards.ieee.org/getieee802/download/802.21-2008.pdf`.

[2] Yan, X.; Şekercioğlu, Y. A.; Narayanan, S. (2010); An overview of vertical handover techniques: Algorithms, protocols and tools, *Comput. Netw.*, ISSN 1389-1286, 54(11):1848-1863.

[3] Marquez-Barja, J.; Calafate, C.T.; Cano, J.C.; Manzoni, P. (2011); A Survey of Vertical Handover Decision Algorithms in Fourth Generation Heterogeneous Wireless Networks, *Computer Communications*, ISSN 0140-3664, 34(8):985-997.

[4] Sousa, B.M.; Pentikousis, K.; Curado, M. (2011); Multihoming Management for Future Networks, *Mob. Netw. Appl.*, ISSN 1383-469X, 16(4):505-517.

[5] Paasch, C.; Detal, G.; Duchene, F.; Raiciu, C.; Bonaventure, O. (2012); Exploring Mobile/WiFi Handover with Multipath TCP, *Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design*, New York, NY, USA: ACM, 31-36 (available at `http://doi.acm.org/10.1145/2342468.2342476`).

[6] Hyun-Dong, C.; Hyunjung, K.; Seung-Joon, S. (2013); Flow based 3G/WLAN vertical handover scheme using MIH model, *2013 International Conference on Information Networking (ICOIN)*, 658-663.

[7] Donoso, Y.; Fabregat, R. (2007); *Multi-Objective Optimization in Computer Networks Using Metaheuristics*, Auerbach Publications, Boston, MA, USA.

[8] Capela, N.; Sargento, S. (2012); Optimizing network performance with multihoming and network coding, *2012 IEEE Globecom Workshops (GC Wkshps)*, pp. 210-215.

[9] Yang, R.; Chang, Y.; Sun, J.; Yang, D. (2012); Traffic Split Scheme Based on Common Radio Resource Management in an Integrated LTE and HSDPA Networks, *2012 IEEE Vehicular Technology Conference (VTC Fall)*, 1-5.

[10] Sungwook, K.; Varshney, P.K. (2002); An adaptive bandwidth reservation algorithm for QoS sensitive multimedia cellular networks, *2002 IEEE Vehicular Technology Conference (VTC Fall)*, 1475-1479.

[11] Sungwook, K.; Varshney, P.K. (2003); Adaptive load balancing with preemption for multimedia cellular networks, *2003 IEEE Wireless Communications and Networking (WCNC)*, 1680-1684.

[12] Li, M.; Fei, Y.; Leung, V.; Randhawa, T. (2003); A new method to support UMTS/WLAN vertical handover using SCTP, *2002 IEEE Vehicular Technology Conference (VTC Fall)*, 1788-1792.

[13] Bin, L.; Boukhatem, N.; Martins, P.; Bertin, P. (2010); Multihoming at layer-2 for inter-RAT handover, *2010 IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, 1173-1178.

[14] Eun, K.P.; Si, Y.H.; Hanlim, K.; Jong-Sam, J.; Seong-Choon, L.; Sang-Hong, L. (2008); Seamless Vertical Handover Using Multihomed Mobile Access Point, *IEEE Global Telecommunications Conference, 2008 (IEEE GLOBECOM 2008)*, 1-4.

[15] Folstad, E.L.; Helvik, B.E. (2009); Managing availability in wireless inter domain access, *International Conference on Ultra Modern Telecommunications Workshops, 2009. (ICUMT '09)*, pp. 1-6.

[16] Lozano-Garzon, C.; Ortiz-Gonzalez, N.; Donoso, Y. (2013); Mobile Network A Proactive VHD Algorithm in Heterogeneous Wireless Networks for Critical Services, *International Journal of Computers, Communications & Control*, ISSN 1841-9836, 8(3):425-431.

[17] Donoso, Y.; Lozano-Garzon, C.; Camelo, M.; Vila, P. (2014); A Fairness Load Balancing Algorithm in Heterogeneous Wireless Networks using a Multihoming Strategy, *International Journal of Computers, Communications & Control*, ISSN 1841-9836, 9(5):555-569.

[18] Mittal, R.; Kansal, A.; Chandra, R. (2012); Empowering Developers to Estimate App Energy Consumption, *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, 317-328.

[19] Deb, K. (2001); *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA.

[20] Rosenthal, R. E. (2015); GAMS a user's guide, *GAMS Development Corporation.*, available at `http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf`.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):273-281, April 2016.

Fuzzy b-Metric Spaces

S. Nădăban

Sorin Nădăban
Department of Mathematics and Computer Science
Aurel Vlaicu University of Arad,
Elena Drăgoi 2, RO-310330 Arad, Romania
snadaban@gmail.com

Abstract: Metric spaces and their various generalizations occur frequently in computer science applications. This is the reason why, in this paper, we introduced and studied the concept of fuzzy b-metric space, generalizing, in this way, both the notion of fuzzy metric space introduced by I. Kramosil and J. Michálek and the concept of b-metric space. On the other hand, we introduced the concept of fuzzy quasi-b-metric space, extending the notion of fuzzy quasi metric space recently introduced by V. Gregori and S. Romaguera. Finally, a decomposition theorem for a fuzzy quasi-pseudo-b-metric into an ascending family of quasi-pseudo-b-metrics is established. The use of fuzzy b-metric spaces and fuzzy quasi-b-metric spaces in the study of denotational semantics and their applications in control theory will be an important next step.
Keywords: Fuzzy b-metric spaces, fuzzy quasi-b-metric, fuzzy quasi-pseudo-b-metric, b-metric space.

1 Introduction and preliminaries

The concept of *b-metric space* was introduced by I.A. Bakhtin [5] and exensively used by S. Czerwic [10, 11].

Definition 1. [10] Let X be a nonempty set and $k \geq 1$ be a given real number. A function $d : X \times X \to [0, \infty)$ is a b-metric on X if, for all $x, y, z \in X$, the following conditions hold:

(b1) $d(x, y) = 0$ if and only if $x = y$;

(b2) $d(x, y) = d(y, x)$;

(b3) $d(x, z) \leq k[d(x, y) + d(y, z)]$.

The triple (X, d, k) will be called b-metric space.

Some examples of b-metric spaces and some fixed point theorems in b-metric spaces can be found in [6–8, 21]. We also note that the class of b-metric spaces is larger than that of metric spaces, since every b-metric is a metric when $k = 1$. In [22] an example of a b-metric space which is not a metric space, is given.

Recently, M.A. Alghamdi, N. Hussain, P. Salimi [1] introduced the notion of *b-metric-like space*, which is an interesting generalization of metric-like space (introduced by A. Amini-Harandi [2]) and partial metric space (introduced by S.G. Matthews [17]). In paper [14], N. Hussain and M.H. Shah introduced the notion of *cone b-metric space*, generalizing both notions of b-metric spaces and cone metric spaces.

The concept of *quasi-b-metric space* was introduced by M.H. Shah and N. Hussain [20] in 2012. In this paper we adopt a slight modification of their definition.

Definition 2. Let X be a nonempty set. A real valued function $d : X \times X \to [0, \infty)$ is said to be a quasi-b-metric with constant $k \geq 1$ if the following conditions are satisfied:

(qb1) $d(x,y) = d(y,x) = 0$ if and only if $x = y$;

(qb3) $d(x,z) \leq k[d(x,y) + d(y,z)], (\forall)x, y, z \in X$.

The triple (X, d, k) will be called quasi-b-metric space.

On the other hand, after L.A. Zadeh has introduced in his famous paper [23] the concept of fuzzy set, one of the important problems is to obtain an adequate notion of *fuzzy metric space*. I. Kramosil and J. Michálek [16] reformulated successfully the notion of probabilistic metric space, introduced by K. Menger in 1942, in fuzzy context.

Definition 3. [19] A binary operation

$$* : [0,1] \times [0,1] \to [0,1]$$

is called triangular norm (t-norm) if it satisfies the following condition:

1. $a * b = b * a, (\forall)a, b \in [0,1]$;
2. $a * 1 = a, (\forall)a \in [0,1]$;
3. $(a * b) * c = a * (b * c), (\forall)a, b, c \in [0,1]$;
4. If $a \leq c$ and $b \leq d$, with $a, b, c, d \in [0,1]$, then $a * b \leq c * d$.

Example 4. Three basic examples of continuous t-norms are $\wedge, \cdot, *_L$, which are defined by $a \wedge b = \min\{a, b\}$, $a \cdot b = ab$ (usual multiplication in $[0,1]$) and $a *_L b = \max\{a+b-1, 0\}$ (the Lukasiewicz t-norm).

Definition 5. [16] The triple $(X, M, *)$ is said to be a fuzzy metric space if X is an arbitrary set, $*$ is a continuous t-norm and M is a fuzzy metric, i.e. a fuzzy set in $X \times X \times [0, \infty)$ such that for all $x, y, z \in X$ we have:

(M1) $M(x, y, 0) = 0$;

(M2) $[M(x, y, t) = 1, (\forall)t > 0]$ if and only if $x = y$;

(M3) $M(x, y, t) = M(y, x, t), (\forall)t \geq 0$;

(M4) $M(x, z, t + s) \geq M(x, y, t) * M(y, z, s), (\forall)t, s \geq 0$;

(M5) $M(x, y, \cdot) : [0, \infty) \to [0, 1]$ is left continuous and $\lim\limits_{t \to \infty} M(x, y, t) = 1$.

We note that A. George and P. Veeramani [12] modified the concept of fuzzy metric space introduced by I. Kramosil and J. Michálek and defined a Hausdorff topology on this fuzzy space. Another approach for fuzzy metric spaces was introduced by O. Kaleva and S. Seikkala in paper [15], by setting the distance between two points to be a non-negative, upper semicontinuous, normal and convex fuzzy number.

In recent years, different types of fuzzy generalized metric spaces was considered by different authors in different approaches. Thus, V. Gregori and S. Romaguera introduced in paper [13] the concept of *fuzzy quasi-metric space*, generalizing in this way the notions of fuzzy metric introduced by I. Kramosil and J. Michálek and by A. George and P. Veeramani to the quasi-metric setting.

On the other hand, the idea of *fuzzy cone metric space* has been introduced in [3] and some basic properties and fixed point theorems for different types of contraction mappings have been developed in fuzzy cone metric spaces. In paper [4], T. Bag introduced the concept of *fuzzy*

cone b-metric space and some fixed point theorems are established in such spaces for contraction mappings. We must note that Bag's definitions for fuzzy cone metric space and for fuzzy cone b-metric spaces generalized the notion of fuzzy metric introduced by Kaleva and Seikkala.

In this paper we introduced and studied the concept of fuzzy b-metric space, generalizing, in this way, both the notion of fuzzy metric space introduced by I. Kramosil and J. Michálek and the concept of b-metric space. On the other hand, we introduced the concept of fuzzy quasi-b-metric space, extending the notion of fuzzy quasi-metric space recently introduced by V. Gregori and S. Romaguera. Finally, a decomposition theorem for a fuzzy quasi-pseudo-b-metric into an ascending family of quasi-pseudo-b-metrics is established.

2 Fuzzy b-metric spaces

Definition 6. Let X be a nonempty set, let $k \geq 1$ be a given real number and $*$ be a continuous t-norm. A fuzzy set M in $X \times X \times [0, \infty)$ is called fuzzy b-metric if, for all $x, y, z \in X$, the following conditions hold:

(bM1) $M(x, y, 0) = 0$;

(bM2) $[M(x, y, t) = 1, (\forall)t > 0]$ if and only if $x = y$;

(bM3) $M(x, y, t) = M(y, x, t), (\forall)t \geq 0$;

(bM4) $M(x, z, k(t + s)) \geq M(x, y, t) * M(y, z, s), (\forall)t, s \geq 0$;

(bM5) $M(x, y, \cdot) : [0, \infty) \to [0, 1]$ is left continuous and $\lim_{t\to\infty} M(x, y, t) = 1$.

The quadruple $(X, M, *, k)$ is said to be a fuzzy b-metric space.

Remark 7. The class of fuzzy b-metric spaces is larger than the class of fuzzy metric spaces, since a fuzzy b-metric space is a fuzzy metric space when $k = 1$.

Example 8. Let (X, d, k) be a b-metric space. Let

$$M_d : X \times X \times [0, \infty) \to [0, 1], M_d(x, y, t) = \begin{cases} \frac{t}{t+d(x,y)} & \text{if} \quad t > 0 \\ 0 & \text{if} \quad t = 0 \end{cases}.$$

Then $(X, M_d, \wedge, k)$ is a fuzzy b-metric space. M_d will be called standard fuzzy b-metric.

Proof: We check only $(bM4)$, because verifying the other conditions is standard.

Let $x, y, z \in X$ and $t, s > 0$. Without restraining the generality we assume that $M_d(x, y, t) \leq M_d(y, z, s)$. Thus $\frac{t}{t+d(x,y)} \leq \frac{s}{s+d(y,z)}$, i.e. $td(y, z) \leq sd(x, y)$.

On the other hand

$$M_d(x, z, k(t + s)) = \frac{k(t + s)}{k(t + s) + d(x, z)} \geq$$

$$\geq \frac{k(t + s)}{k(t + s) + k[d(x, y) + d(y, z)]} = \frac{t + s}{t + s + d(x, y) + d(y, z)}.$$

We will prove that

$$\frac{t + s}{t + s + d(x, y) + d(y, z)} \geq \frac{t}{t + d(x, y)}.$$

Hence we will obtain that $M_d(x, z, k(t + s)) \geq M_d(x, y, t) = M_d(x, y, t) \wedge M_d(y, z, s)$, what had to be verified. We remark that

$$\frac{t + s}{t + s + d(x, y) + d(y, z)} \geq \frac{t}{t + d(x, y)} \Leftrightarrow$$

$$t^2 + st + td(x,y) + sd(x,y) \geq t^2 + st + td(x,y) + td(y,z) \Leftrightarrow sd(x,y) \geq td(y,z) \ ,$$

which is true. □

Definition 9. Let $k \geq 1$ be a real given number. A function $f : \mathbb{R} \to \mathbb{R}$ will be called k-nondecreasing if for $t < s$ we will have that $f(t) \leq f(ks)$.

Proposition 10. For all $x, y \in X$ the mapping $M(x, y, \cdot) : [0, \infty) \to [0, 1]$ is k-nondecreasing.

Proof: For $0 < t < s$ we have

$$M(x, y, ks) \geq M(x, x, s - t) * M(x, y, t) = 1 * M(x, y, t) = M(x, y, t) \ .$$

□

Theorem 2.1. Let $(X, M, *, k)$ be a fuzzy b-metric space. For $x \in X, r \in (0, 1)$, $t > 0$ we define the open ball

$$B(x, r, t) := \{y \in X \ : \ M(x, y, t) > 1 - r\} \ .$$

Then

$$\mathcal{T}_M := \{T \subset X \ : \ x \in T \text{ iff } (\exists)t > 0, r \in (0, 1) \ : \ B(x, r, t) \subseteq T\}$$

is a topology on X.

Proof: It is obvious that $\emptyset$ and X belong to $\mathcal{T}_M$.

Let $\{T_i\}_{i \in I} \subseteq \mathcal{T}_M$ and $T = \bigcup\limits_{i \in I} T_i$. We will show that $T \in \mathcal{T}_M$. Let $x \in T$. Then there exists $i_0 \in I$ such that $x \in T_{i_0}$. As $T_{i_0} \in \mathcal{T}_M$, there exist $t > 0$, $r \in (0, 1)$ such that $B(x, r, t) \subseteq T_{i_0}$. Thus $B(x, r, t) \subseteq \bigcup\limits_{i \in I} T_i = T$.

Let now $\{T_i\}_{i=1}^n \subseteq \mathcal{T}_M$ and $T = \bigcap\limits_{i=1}^n T_i$. We will show that $T \in \mathcal{T}_M$. Let $x \in T$. We obtain that $x \in T_i, (\forall)i = \overline{1, n}$. Thus

$$(\exists)t_i > 0, r_i \in (0, 1) \ : \ B(x, r_i, t_i) \subseteq T_i, (\forall)i = \overline{1, n} \ .$$

Let

$$r = \min\{r_i, i = \overline{1, n}\}, t = \min\left\{\frac{t_i}{k}, i = \overline{1, n}\right\} \ .$$

We have that $B(x, r, t) \subseteq B(x, r_i, t_i), (\forall)i = \overline{1, n}$. Indeed, for $y \in B(x, r, t)$, we have $M(x, y, t) > 1 - r \geq 1 - r_i, (\forall)i = \overline{1, n}$. As $t \leq \frac{t_i}{k}, (\forall)i = \overline{1, n}$, we obtain that $M(x, y, t) \leq M(x, y, t_i)$. Thus $M(x, y, t_i) > 1 - r_i, (\forall)i = \overline{1, n}$. Hence $y \in B(x, r_i, t_i), (\forall)i = \overline{1, n}$. Therefore $B(x, r, t) \subseteq T_i, (\forall)i = \overline{1, n}$. Thus $B(x, r, t) \subseteq \bigcap\limits_{i=1}^n T_i = T$. □

Remark 11. Previous theorem extends to fuzzy b-metric space a similar result obtained by A. George and P. Veeramani [12] in the context of fuzzy metric space. The definitions for convergent sequence and Cauchy sequence given by A. George and P. Veeramani [12] in the context of fuzzy metric space can be translated in the context of fuzzy b-metric space, as follows.

Definition 12. Let $(X, M, *, k)$ be a fuzzy b-metric space and (x_n) be a sequence in X. The sequence (x_n) is said to be convergent if there exists $x \in X$ such that $M(x_n, x, t) = 1, (\forall)t > 0$. In this case, x is called the limit of the sequence (x_n) and we note $\lim\limits_{n \to \infty} x_n = x$, or $x_n \to x$.

Remark 13. Let $(X, M, *, k)$ be a fuzzy b-metric space. A sequence (x_n) is convergent to x if and only if (x_n) is convergent to x in topology $\mathcal{T}_M$.

Indeed,

$$x_n \to x \text{ in topology } \mathcal{T}_M \Leftrightarrow$$

$$\Leftrightarrow (\forall)r \in (0,1), (\forall)t > 0, (\exists)n_0 \in \mathbb{N} \ : \ x_n \in B(x,r,t), (\forall)n \geq n_0 \Leftrightarrow$$

$$\Leftrightarrow (\forall)r \in (0,1), (\forall)t > 0, (\exists)n_0 \in \mathbb{N} \ : \ M(x_n,x,t) > 1-r, (\forall)n \geq n_0 \Leftrightarrow$$

$$\Leftrightarrow \lim_{n\to\infty} M(x_n,x,t) = 1, (\forall)t > 0 \ .$$

Definition 14. Let $(X, M, *, k)$ be a fuzzy b-metric space and (x_n) be a sequence in X. The sequence (x_n) is said to be a Cauchy sequence if

$$(\forall)r \in (0,1), (\forall)t > 0, (\exists)n_0 \in \mathbb{N} \ : \ M(x_n,x_m,t) > 1-r, (\forall)n,m \geq n_0 \ .$$

A fuzzy b-metric space in which every Cauchy sequence is convergent is called complete fuzzy b-metric space.

3 Fuzzy quasi-b-metric spaces

Definition 15. A fuzzy quasi-b-metric space is a quadruple $(X, M, *, k)$, where X is a nonempty set, $*$ is a continuous t-norm, $k \geq 1$ is a given real number and M is a fuzzy set in $X \times X \times [0,\infty)$ such that for all $x, y, z \in X$ we have:

(qbM1) $M(x,y,0) = 0$;

(qbM2) $[M(x,y,t) = M(y,x,t) = 1, (\forall)t > 0]$ if and only if $x = y$;

(qbM3) $M(x,z,k(t+s)) \geq M(x,y,t) * M(y,z,s), (\forall)t,s \geq 0$;

(qbM4) $M(x,y,\cdot) : [0,\infty) \to [0,1]$ is left continuous and $\lim_{t\to\infty} M(x,y,t) = 1$.

Remark 16. V. Gregori and S. Romaguera [13] also gave this definition in the particular case $k = 1$ and the triple $(X, M, *)$ is called fuzzy quasi-metric space.

Proposition 17. If Q is a fuzzy quasi-b-metric, then Q^{-1} defined by $Q^{-1}(x,y,t) = Q(y,x,t)$ is also a fuzzy quasi-b-metric (called the conjugate of Q).

Proof: We have to check only $(qbM3)$.

$$Q^{-1}(x,z,k(t+s)) = Q(z,x,k(s+t)) \geq Q(z,y,s) * Q(y,x,t) = Q^{-1}(x,y,t) * Q^{-1}(y,z,s) \ .$$

□

Definition 18. [18]. Let $*, \circ$ be two t-norms. We say that $\circ$ dominates $*$ and we denote $\circ \gg *$ if

$$(x_1 \circ x_2) * (y_1 \circ y_2) \leq (x_1 * y_1) \circ (x_2 * y_2), (\forall)x_1, x_2, y_1, y_2 \in [0,1].$$

Remark 19. [18]. For any t-norm $*$ we have $\wedge \gg *$.

Proposition 20. Let $(X, Q, *, k)$ be a fuzzy quasi-b-metric space and $\circ$ be a continuous t-norm such that $\circ \gg *$. Let M be a fuzzy set in $X \times X \times [0,\infty)$ defined by

$$M(x,y,t) = Q(x,y,t) \circ Q^{-1}(x,y,t) \ .$$

Then $(X, M, *, k)$ is a fuzzy b-metric space.

Proof: It is easy to check $(bM1)-(bM3)$ and $(bM5)$. We prove $(bM4)$.

$$M(x,z,k(t+s)) = Q(x,z,k(t+s)) \circ Q^{-1}(x,z,k(t+s)) \geq$$

$$\geq [Q(x,y,t) * Q(y,z,s)] \circ [Q^{-1}(x,y,t) * Q^{-1}(y,z,s)] \geq$$

$$\geq [Q(x,y,t) \circ Q^{-1}(x,y,t)] * [Q(y,z,s) \circ Q^{-1}(y,z,s)] = M(x,y,t) * M(y,z,s) \ .$$

□

Corollary 21. Let $(X, Q, *, k)$ be a fuzzy quasi-b-metric space and

$$M(x,y,t) = \min\{Q(x,y,t), Q(y,x,t)\} \ .$$

Then $(X, M, *, k)$ is a fuzzy b-metric space.

Proof: We apply previous proposition for $\circ = \wedge \gg *$. □

Example 22. Let (X, d, k) be a quasi-b-metric space. Let

$$M_d : X \times X \times [0,\infty) \to [0,1], M_d(x,y,t) = \begin{cases} \frac{t}{t+d(x,y)} & \text{if} \quad t > 0 \\ 0 & \text{if} \quad t = 0 \end{cases} \ .$$

Then $(X, M_d, \wedge, k)$ is a fuzzy quasi-b-metric space. M_d will be called standard fuzzy quasi-b-metric.

Proof: The proof is standard. □

Proposition 23. If $(X, M, *, k)$ is a fuzzy quasi-b-metric space, then the relation $\leq_M$ on X defined by

$$x \leq_M y \text{ if and only if } M(x,y,t) = 1, (\forall)t > 0$$

is a partial ordering.

Proof. It is easy to check.

4 Fuzzy quasi-pseudo-b-metric spaces

Definition 24. Let X be a nonempty set. A function $d : X \times X \to [0,\infty)$ is called quasi-pseudo-b-metric with constant $k \geq 1$ if the following conditions are satisfied:

(qpb1) $d(x,x) = 0$;

(qpb3) $d(x,z) \leq k[d(x,y) + d(y,z)], (\forall)x,y,z \in X$.

The triple (X, d, k) will be called quasi-pseudo-b-metric space.

Definition 25. A fuzzy quasi-pseudo-b-metric space is a quadruple $(X, M, *, k)$, where X is a nonempty set, $*$ is a continuous t-norm, $k \geq 1$ is a given real number and M is a fuzzy set in $X \times X \times [0,\infty)$ such that for all $x,y,z \in X$ we have:

(qpbM1) $M(x,y,0) = 0$;

(qpbM2) $[M(x,x,t) = 1, (\forall)t > 0]$;

(qpbM3) $M(x,z,k(t+s)) \geq M(x,y,t) * M(y,z,s), (\forall)t,s \geq 0$;

(qpbM4) $M(x,y,\cdot):[0,\infty)\to[0,1]$ is left continuous and $\lim\limits_{t\to\infty}M(x,y,t)=1$.

Theorem 4.1. Let $(X,M,\wedge,k)$ be a fuzzy quasi-pseudo-b-metric space and

$$d_\alpha(x,y):=\inf\{t>0\ :\ M(x,y,t)>\alpha\},\alpha\in(0,1)\ .$$

Then $\mathcal{D}=\{d_\alpha\}_{\alpha\in(0,1)}$ is an ascending family of quasi-pseudo-b-metrics on X.

Proof: $(qp1)$ $d_\alpha(x,x)=\inf\{t>0\ :\ M(x,x,t)>\alpha\}=0$.
$(qp2)$

$$k[d_\alpha(x,y)+d_\alpha(y,z)]=k[\inf\{t>0\ :\ M(x,y,t)>\alpha\}+\inf\{s>0\ :\ M(y,z,s)>\alpha\}]=$$

$$=k[\inf\{t+s>0\ :\ M(x,y,t)>\alpha,M(y,z,s)>\alpha\}]=$$

$$=\inf\{k(t+s)>0\ :\ M(x,y,t)\wedge M(y,z,s)>\alpha\}\geq$$

$$\geq\inf\{k(t+s)>0\ :\ M(x,z,k(t+s))>\alpha\}=d_\alpha(x,z)\ .$$

It remains to prove that $\mathcal{D}=\{d_\alpha\}_{\alpha\in(0,1)}$ is an ascending family. Let $\alpha_1\leq\alpha_2$. Then

$$\{t>0\ :\ M(x,y,t)>\alpha_2\}\subseteq\{t>0\ :\ M(x,y,t)>\alpha_1\}\ .$$

Thus

$$\inf\{t>0\ :\ M(x,y,t)>\alpha_2\}\geq\inf\{t>0\ :\ M(x,y,t)>\alpha_1\}\ ,$$

namely $d_{\alpha_2}(x,y)\geq d_{\alpha_1}(x,y),(\forall)(x,y)\in X\times X$. □

5 Conclusions and further works

In this paper we introduce the notions of fuzzy b-metric space and fuzzy quasi-b-metric space. Thus, we have built a fertile ground to study, in further papers, some fixed point theorems in these spaces. The first problem is to established fuzzy versions of Banach contraction mapping principle in fuzzy b-metric spaces. From here we will obtain a lot of applications both in Mathematics as well as in Engineering and Computer Science. The second issue is to study set-valued contractions in fuzzy b-metric spaces and their applications in control theory and convex optimization. A real challenge is to extend the results of C. Chifu and G. Petruşel [9] in fuzzy b-metric spaces. We intend to obtain some fixed point theorems for multivalued operators in fuzzy b-metric spaces endowed with a graph. This paper may be of interest for researchers working in the following fields belonging to Computer Science and Information Technology:

(i) Integrated solution in computer-based control and communications

(ii) Computational intelligence methods

(iii) Advanced decision support systems

where fuzzy metric spaces will be applied in dealing with the problems such as: fixed point theorems and their applications in the semantics of programs; distance measurement between programs with important results to measure the complexity of programs and algorithms; color image processing and image denoising; the use of some types of fuzzy metrics in cognitive information, in time series and in bioinformatics; the appplications in neural networks; data mining and web mining applications.

Bibliography

[1] Alghamdi, M.A., Hussain, N., Salimi, P. (2013); Fixed point and coupled fixed point theorems on b-metric-like spaces, *Journal of Inequalities and Applications*, 2013:402.

[2] Amini-Harandi, A. (2012). Metric-like spaces, partial metric spaces and fixed points, *Fixed Point Theory and Applications*, 2012:204.

[3] Bag, T. (2013); Fuzzy cone metric spaces and fixed point theorems of contractive mappings, *Annals of Fuzzy Mathematics and Informatics*, 6(3): 657–668.

[4] Bag, T. (2014); Some fixed point theorems in fuzzy cone b-metric spaces, *International Journal of Fuzzy Mathematics and Systems*, 4(2): 255–267.

[5] Bakhtin, I.A. (1989); The contraction mapping principle in quasi-metric spaces, *Funct. Anal. Unianowsk Gos. Ped. Inst.*, 30: 26–37.

[6] Boriceanu, M., Bota, M., Petruşel, A. (2010); Multivalued fractals in b-metric spaces, *Central European Journal of Mathematics*, 8(2): 367–377.

[7] Boriceanu, M., Petruşel, A., Rus, I.A. (2010); Fixed point theorems for some multivalued generalized contraction in b-metric spaces, *International J. Math. Statistics*, 6: 65–76.

[8] Boriceanu, M. (2009); Strict fixed point theorems for multivalued operators in b-metric spaces, *Intern. J. Modern Math.*, 4: 285–301.

[9] Chifu, C., Petruşel, G. (2014); Fixed point for multivalued contraction in b-metric spaces with applications to fractals, *Taiwanese Journal of Mathematics*, 18(5): 1365–1375.

[10] Czerwik, S. (1993); Contraction mappings in b-metric space, *Acta Math. Inf. Univ. Ostraviensis*, 1: 5–11.

[11] Czerwik, S. (1998); Non-linear set-valued contraction mappings in b-metric spaces, *Atti. Sem. Math. Fig. Univ. Modena*, 46(2): 263–276.

[12] George, A., Veeramani, P. (1994); On some results in fuzzy metric spaces, *Fuzzy Sets and Systems*, 64: 395–399.

[13] Gregori, V., Romaguera, S. (2004); Fuzzy quasi-metric spaces, *Applied General Topology*, 5(1): 128–136.

[14] Hussain, N., Shah, M.H. (2011); KKM mappings in cone b-metric spaces, *Comput. Math. Appl.*, 61(4): 1677–1684.

[15] Kaleva, O., Seikkala, S. (1984); On fuzzy metric spaces, *Fuzzy Sets and Systems*, 12: 215–229.

[16] Kramosil, I., Michálek, J. (1975); Fuzzy metric and statistical metric spaces, *Kybernetica*, 11: 326–334.

[17] Matthews, S.G. (1994); Partial metric topology, in: *Proc. 8th Summer Conference on General Topology and Applications*, Ann. New York Acad. Sci., Vol. 728, The New York Academy of Sciences, 183–197.

[18] Nădăban, S. (2015); Fuzzy euclidean normed spaces for data mining applications, *International Journal of Computers Communications & Control*, 10(1): 70–77.

[19] Schweizer, B., Sklar, A. (1960); Statistical metric spaces, *Pacific J. Math.*, 10: 314–334.

[20] Shah, M.H., Hussain, N. (2012); Nonlinear contraction in partially ordered quasi b-metric spaces, *Commun. Korean Math. Soc.*, 27(1): 117–128.

[21] Shatanawi, W., Pitea, A., Lazović, R. (2014); Contraction conditions using comparison function on b-metric spaces, *Fixed Point Theory and Applications*, 2014:135.

[22] Singh, S.L., Prasad, B. (2008); Some coincidence theorems and stability of iterative procedures, *Computers and Mathematics with Applications*, 55: 2512–2520.

[23] Zadeh, L.A. (1965); Fuzzy Sets, *Informations and Control*, 8: 338–353.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):282-291, April 2016.

The Maximum Flows in Planar Dynamic Networks

C. Schiopu, E. Ciurea

Camelia Schiopu*
1. Transilvania University of Brasov
Romania, 500091 Braşov, Iuliu Maniu, 50
*Corresponding author: camelia.s@unitbv.ro

Eleonor Ciurea
Transilvania University of Brasov
Romania, 500091 Braşov, Iuliu Maniu, 50
e.ciurea@unitbv.ro

Abstract: An nontrivial extension of the maximal static flow problem is the maximal dynamic flow model, where the transit time to traverse an arc is taken into consideration. If the network parameters as capacities, arc traversal times, and so on, are constant over time, then a dynamic flow problem is said to be stationary. Research on flow in planar static network is motivated by the fact that more efficient algorithms can be developed by exploiting the planar structure of the graph. This article states and solves the maximum flow in directed $(1, n)$ planar dynamic networks in the stationary case.
Keywords: network flow, planar network, dynamic network, maximum flow.

1 Introduction

The static network flow models bridges several diverse and seemingly unrelated areas of combinatorial optimization. More often in scientific writing, flow in a network refers to the flow of electricity, phone calls, email messages, commodities being transported across truck routes, or other such kinds of flow. Many efficient algorithms have been developed to solve the maximum flows problem in static network [1].

The planar static network also arise in practical contexts such as VLSI design and communication networks, and hence it is of interest to find fast flow algorithms for this class of graphs. The computation of a maximum flow in a planar static network has been investigated by many researchers starting from the work of Ford and Fulkerson [5] who developed an $O(n^2)$ time algorithm for $(1, n)$ networks when the source node 1 and sink node n are on the same face. This algorithm was later improved to $O(n \log n)$ time by Itai and Shiloach [8]. By introducing the concept of potentials, Hassin [6] gave an algorithm that run in $O(n \log^{0.5} n)$ time using Frederickson′s shortest path algorithm [4]. Itai and Shiloach [8] also developed an algorithm to find a maximum flow in an undirected planar networks when the source node and sink node are not on the same face. For faster maximum flow algorithms in planar (but not necessarily $(1, n)$ planar) undirected and directed static networks see Hassin and Johnson [7] and Johnson and Venkatesan [9]. Khuller and Naor [10] present the flow in planar static networks with nodes capacities.

However, in some other applications, time is an essential ingredient [1]. In this instance, to account properly for the evolution of the underlying system over time, we need to use dynamic network flow models. For dynamic network flow problem see [1], [2], [3].

In this paper, we present the maximum flow problem in directed $(1, n)$ planar dynamic networks. We present the case when the planar dynamic network is stationary. Further on, in Section 2 the maximum flow in directed $(1, n)$ planar static network is exposed. In Section 3 some

basic dynamic network notations and results are presented, while in Section 4 is presented the method for solving the maximum flow in directed $(1, n)$ planar dynamic network. The conclusions are presented in Section 5 and an example is given in Section 6.

2 The maximum flow in directed $(1, n)$ planar static network

Research on flow in planar static network is motivated by the fact that more efficient algorithms can be developed by exploiting the planar structure of the digraph.

Definition 1. A digraph $G = (N, A)$ is said to be planar if we can draw it in a two-dimensional plane so that no two arcs intersect each other.

Researchers have developed very efficient algorithms (in fact, linear time algorithms) for testing the planarity of a digraph.

Definition 2. Let $G = (N, A)$ be a planar digraph. A face of G is a region of the plane bounded by arcs that satisfies the condition that any two points in the region can be connected by a continuous curve that meets no nodes and arcs. The boundary of a face x is the set of all arcs that enclose it. Faces x and y said to be adjacent if their boundaries contain a common arc.

The planar digraph G has an unbounded face.

Recall two well-known properties of planar digraphs:

– If a connected planar digraph has n nodes, m arcs and q faces, then $q = m - n + 2$;

– If a planar digraph has n nodes and m arcs, then $m < 3n$.

Our discussion in this paper applies to a special class of planar digraphs known as $(1, n)$ planar digraphs (the node source 1 and node sink n lie on the boundary of unbounded face).

Let $G = (N, A, u)$ be a static network with the set of nodes $N = \{1, \ldots, i, \ldots, j, \ldots, n\}$, the set of arcs $A = \{a_1, \ldots, a_k, \ldots, a_m\}$, $a_k = (i, j)$ and the upper bound (capacity) function $u : A \to \mathbb{R}_+$, where $\mathbb{R}$ is real number set. To define maximal static flow problem, we distinguish two special nodes in the static network $G = (N, A, u)$: a source node 1 and a sink node n.

A static flow is a function $f : A \to \mathbb{R}_+$, satisfying the following conditions:

$$\sum_j f(i,j) - \sum_k f(k,i) = \begin{cases} v, & \text{if } i = 1 \quad (1a) \\ 0, & \text{if } i \neq 1, n \quad (1b) \\ -v, & \text{if } i = n \quad (1c) \end{cases}$$

$$0 \leq f(i,j) \leq u(i,j), \quad (i,j) \in A \tag{2}$$

for some $v \geq 0$.

We refer to v as the value of the static flow f. The maximum flow problem is to determine a flow f which maximizes v. A cut is a partition of the node set N into two subsets S and $\bar{S} = N - S$. We represent this cut using notation $\left[S, \bar{S}\right]$. We refer to an arc (i, j) with $i \in S$ and $j \in \bar{S}$ as a forward arc of the cut and an arc (j, i) with $j \in \bar{S}$ and $i \in S$ as a backward arc of the cut. Let $\left(S, \bar{S}\right)$ denote the set of forward arcs in the cut and let $\left(\bar{S}, S\right)$ denote the set of backward arcs. We have that the arc set of cut is $\left[S, \bar{S}\right] = \left(S, \bar{S}\right) \cup \left(\bar{S}, S\right)$. We refer to a cut $\left[S, \bar{S}\right]$ as an $1 - n$ cut if $1 \in S$ and $n \in \bar{S}$.

For the maximum flow problem, we define the capacity of the $1 - n$ cut $\left[S, \bar{S}\right]$ as:

$$c[S,\bar{S}] = \sum_{(S,\bar{S})} u(i,j) \tag{3}$$

We refer to an $1-n$ cut whose capacity is the minimum among all $1-n$ cuts as a minimal cut.

Recall the maximum flow minimum cut theorem.

Theorem 3. *The maximum value of the flow from a source node* 1 *to a sink node* n *in network* G *equals the capacity of minimum* $1-n$ *cut.*

Many efficient algorithms have been developed to solve the maximum flows problem in some static network [1].

Next we present the maximum flow problem in directed $(1,n)$ planar static network. First, we define the dual directed static network denoted by $G^{'} = (N^{'}, A^{'}, c^{'})$. We add the arc $(n,1)$ with $u(n,1) = 0$, which divides the unbounded face into two faces: a new bounded face and a new unbounded face. In this case we have $n^{'} = q+1$ faces, with $q = m-n+2$. Then we place a node $x^{'}$ inside each face x of the network G. We have $N^{'} = \{1^{'}, \ldots, x^{'}, \ldots, y^{'}, \ldots, n^{'}\}$. Let $1^{'}$ and $n^{'}$, respectively, denote the nodes in the dual directed static network $G^{'}$ corresponding to the new bounded face and the new unbounded face. Each arc $(i,j) \in A$ lies on the boundary of the two faces x and y. Corresponding to this arc, the network $G^{'}$ contains two oppositely arcs $(x^{'}, y^{'})$ and $(y^{'}, x^{'})$. If arc (i,j) is a clockwise arc in the face x, we define the cost $c^{'}(x^{'}, y^{'}) = u(i,j)$ and the cost $c^{'}(y^{'}, x^{'}) = 0$. We define arc costs in the opposite manner if arc (i,j) is a counterclockwise arc in the face x. The network $G^{'}$ contains the arcs $(1^{'}, n^{'})$ and $(n^{'}, 1^{'})$ which we delete from the network. We have $A^{'} = \{(x^{'}, y^{'}), (y^{'}, x^{'}) | x^{'}, y^{'} \in N^{'}, (x^{'}, y^{'})$ and $(y^{'}, x^{'})$ correspond to $(i,j) \in A\}$. There is an one-to-one correspondence between $1-n$ cuts in the network G and paths from node $1^{'}$ to node $n^{'}$ in the network $G^{'}$. Moreover, the capacity of the cut equals the cost of the corresponding path. Consequently, we can obtain a minimum $1-n$ cut $[S,\bar{S}]$ and $c[S,\bar{S}]$ in network G by determining a shortest path $P^{'}$ and $c^{'}(P^{'})$ from node $1^{'}$ to node $n^{'}$ in the network $G^{'}$. We can solve the shortest path problem in the network $G^{'}$ using Dijkstra′s algorithm [1].

Now, we present an algorithm for finding a maximum flow in a directed $(1,n)$ planar static network $G = (N, A, u)$. Let $d^{'}(x^{'})$ denote the shortest path distance from node $1^{'}$ to node $x^{'}$ in the dual directed static network $G^{'} = (N^{'}, A^{'}, c^{'})$. The algorithm for maximum flow in directed $(1,n)$ planar static network (AMFDPSN) is presented in Figure 1 [1].

```
1: AMFDPSN;
2: begin
3: compute the network G';
4: DIJKSTRA (G', d');
5: for (i,j) ∈ A do
6:     f(i,j) := d'(y') − d'(x');
7: end for
8: end.
```

Figure 1: Algorithm for maximum flow in directed $(1,n)$ planar static network

Theorem 4. *The AMFDPSN determines a maximum flow in network* G.

Theorem 5. *The AMFDPSN determines a maximum flow in* $O(n^2)$ *time.*

Using Frederickson′s algorithm (see [4]), the AMFDPSN determines a maximum flow in $O(n^{1.5})$ time.

3 The maximum flows in dynamic network

Let $G = (N, A, u)$ be a static network with the node set N, the arc set A, the upper bound function $u, 1$ the source node and n the sink node.

Let $\mathbb{N}$ be the natural number set and let $H = \{0, 1, \ldots, T\}$ be the set of periods, where T is a finite time horizon, $T \in \mathbb{N}$. Let us state the transit time function $h : A \times H \to \mathbb{N}$ and the time capacity function $u_h : A \times H \to \mathbb{R}_+$, where $h(i, j; t)$ represents the transit time of arc (i, j) at time t, $t \in H$ and $u_h(i, j; t)$ represents the capacity (upper bound) of arc (i, j) at time $t, t \in H$.

The maximal dynamic flow problem for T time periods is to determine a flow function $f_h : A \times H \to \mathbb{N}$, which should satisfy the following conditions in dynamic network $G_h = (N, A, h, u_h)$:

$$\sum_{t=0}^{T}\Big(\sum_{j} f_h(i, j; t) - \sum_{k}\sum_{\tau} f_h(k, i; \tau)\Big) = v_H, \text{if } i = 1 \tag{4a}$$

$$\sum_{j} f_h(i, j; t) - \sum_{k}\sum_{\tau} f_h(k, i; \tau) = 0, \text{if } i \neq 1, n, \quad t \in H \tag{4b}$$

$$\sum_{t=0}^{T}\Big(\sum_{j} f_h(i, j; t) - \sum_{k}\sum_{\tau} f_h(k, i; \tau)\Big) = -v_H, \text{if } i = n \tag{4c}$$

$$0 \leq f_h(i, j; t) \leq u_h(i, j; t), \text{for all } \ (i, j) \in A \ \text{ and for all } \ t \in H \tag{5}$$

$$\max v_H, \tag{6}$$

where $\tau = t - h(k, i; \tau)$, $v_H = \sum_{t=0}^{T} v(t)$, $v(t)$ is the flow value at time t and $f_h(i, j; t) = 0, \ (i, j) \in A, \ t \in \{T - h(i, j; t) + 1, \ldots, T\}$.

In other words, a dynamic flow f_h from 1 to n is any flow f_h from 1 to n in which not more than $u_h(i, j; t)$ flow units starting from node i at time t and arriving at node j at time $t + h(i, j; t)$ for all arcs (i, j) and all t. Note that in a dynamic flow, units may be departing from the source at time $0, 1, \ldots, T'$, $T' < T$. A maximum dynamic flow for T time periods from 1 to n is any dynamic flow from 1 to n in which the maximum possible number of flow units arrive at the sink node n during the first T time periods. We will show how to transform the maximum dynamic flow problem in the dynamic network $G_h = (N, A, h, u_h)$ into a static flow problem on a static network $G'_H = (N'_H, A'_H, u'_H)$, called the reduced time-expanded network.

For a given dynamic network $G_h = (N, A, h, u_h)$, we form the time expanded network $G_H = (N_H, A_H, u_H)$ as follows. We make $T + 1$ copies i_t, $t = 0, 1, \ldots, T$ of each node i in G_h. Node i_t in G_H represents node i in G_h at time t. For each (i, j) in G_h, there are arcs (i_t, j_θ), $\theta = t + h(i, j; t), \quad t = 0, 1, \ldots, T - h(i, j; t)$ with capacity $u_H(i_t, j_\theta) = u_h(i, j; t)$ in G_H. The arc (i_t, j_θ) in G_H represents the potential movement of a commodity from node i to node j in time $h(i, j; t)$. The number of nodes in G_H is $n(T + 1)$, and number of arcs is limited by $m(T + 1) - \sum_{A} \bar{h}(i, j)$, where $\bar{h}(i, j) = \min\{h(i, j; 0), \ldots, h(i, j; T)\}$. It is easy to see that any dynamic flow in dynamic network G_h is equivalent to a static flow in static network G_H from the source nodes $1_0, 1_1, \ldots, 1_T$ to the sink nodes $n_0, n_1, \ldots, n_T$ and vice versa. We can further reduce the multiple source, multiple sink problem in network G_H to the single source, single sink problem by introducing a supersource node 1^* and a supersink node n^* constructing time superexpanded network $G^*_H = (N^*_H, A^*_H, u^*_H)$, where $N^*_H = N_H \cup \{1^*, n^*\}$, $A^*_H = A_H \cup \{(1^*, 1_t) | t = 0, 1, \ldots, T\} \cup \{(n_t, n^*) | t = 0, 1, \ldots, T\}$, $u^*_H(i_t, j_\theta) = u_H(i_t, j_\theta)$ for all $(i_t, j_\theta) \in A_H$, $u^*_H(1^*, 1_t) = u^*_H(n_t, n^*) = \infty, \ t = 0, 1, \ldots, T$. Now, we construct the time reduced

expanded network $G_H^{'} = (N_H^{'}, A_H^{'}, u_H^{'})$ as follows. We define the function h^*, $h^* : A_H^* \to \mathbb{N}$, $h^*(1^*, 1_t) = h^*(n_t, n^*) = 0$, $t = 0, 1, \ldots, T$, $h^*(i_t, j_\theta) = h(i, j; t)$, $t = 0, 1, \ldots, T - h(i, j; t)$. Let $d^*(1^*, i_t)$ be the length of the shortest path from the source node 1^* to the node i_t in network G_H^* and $d^*(i_t, n^*)$the length of the shortest path from node i_t to the sink node n^*, with respect to h^*. The computation of $d^*(1^*, i_t)$ and $d^*(i_t, n^*)$ for all $i_t \in N_H^*$ is performed by means of the usual shortest path algorithms. We have $N_H^{'} = \{1^*, n^*\} \cup \{i_t | i_t \in N_H, d^*(1^*, i_t) + d^*(i_t, n^*) \leq T\}$, $A_H^{'} = \{(1^*, 1_t) | d^*(1_t, n^*) \leq T\} \cup \{(n_t, n^*) | d^*(1^*, n_t) \leq T\} \cup \{(i_t, j_\theta) | (i_t, j_\theta) \in A_H, d^*(1^*, i_t) + h^*(i_t, j_\theta) + d^*(j_\theta, n^*) \leq T\}$ and $u_H^{'}$ is restriction of u_H^* at $A_H^{'}$. In network $G_H^{'}$ we rewrite the nodes 1^*, n^* by $1^{'}$, respectively $n^{'}$. It is easy to see that the network $G_H^{'}$ is always a partial subnetwork of G_H^*. Since an item released from a node at a specific time does not return to that location at the same or an earlier time, the networks G_H, G_H^*, $G_H^{'}$ cannot contain any circuit, and are therefore acyclic always.

In the most general dynamic model, the parameter $h(i) = 1$ is waiting time at node i, and the parameter $u_h(i; t)$ is defined as the capacity of the node i, which represents the maximum amount of flow that can wait at node i from time t to $t + 1$. This most general dynamic model is not discussed in this paper.

The maximum dynamic flow problem for T time periods in dynamic network G_h formulated in conditions (4), (5), (6) is equivalent with the maximum static flow problem in static network $G_H^{'}$ as follows:

$$\sum_{j_\theta} f_H^{'}(i_t, j_\theta) - \sum_{k_{\tau'}} f_H^{'}(k_{\tau'}, i_t)) = \begin{cases} v_H^{'}, & \text{if } i_t = 1^{'}, \quad (7a) \\ 0, & \text{for all } i_t \neq 1^{'}, n^{'}, \quad (7b) \\ -v_H^{'}, & \text{if } i_t = n^{'}, \quad (7c) \end{cases}$$

$$0 \leq f_H^{'}(i_t, j_\theta) \leq u_H^{'}(i_t, j_\theta), \text{for all } (i_t, j_\theta) \in A_H^{'} \quad (8)$$

$$max \quad v_H^{'}, \quad (9)$$

where by convention $i_t = 1^{'}$ for $t = -1$ and $i_t = n^{'}$ for $t = T + 1$.

It is easy to see that network $G_H^{'}$ is no planar, in general.

A dynamic flow problem is said to be stationary if the network parameters as capacities, arc traversal times, and so on, are constant over time ($c : A \to \mathbb{R}_+, h : A \to \mathbb{N}$, and so on). In the stationary case it does not require the construction of the reduce time expanded static network $G_H^{'} = (N_H^{'}, A_H^{'}, u_H^{'})$ for solving the maximum dynamic flow problem for any T. A maximum dynamic flow in the stationary case can be generated from a maximum value and minimum time flow f in static network $G = (N, A, c, u)$, where $c(i, j) = h(i, j)$ is the cost for any arc $(i, j) \in A$. The algorithm for stationary maximum dynamic flow (ASMDF) problem is presented in Figure 2 [5].

1: ASMDF;
2: BEGIN
3: AMVMCF (G,f);
4: ADFEF $(f, r(P_1), \ldots, r(P_k))$;
5: ARF $(r(P_1), \ldots, r(P_k))$;
6: END.

Figure 2: Algorithm for stationary maximum dynamic flow.

The procedure AMVMCF performs the algorithm for maximum value and minimum cost flow f in network G. For statements we suppose that use the algorithm of Klein variant (minimum

mean cycle canceling algorithm, see [1]). This algorithm have the complexity $O(n^2m^3 \log n)$. The procedure ADFEF performs the algorithm for decomposition of flow f in elementary flows with $r(P_1), ..., r(P_k)$ path flows. Is necessary that $c(P_i) \leq T$. This algorithm have complexity $O(m^2)$. The procedure ARF performs the algorithm for send $r(P_i)$ flow, $i = 1 \ldots, k$, starting out from source node 1 at time periods 0 and repeat it after each time period as long as there is enough time left in the horizon for the flow along the path to arrive at the sink node n. This algorithm have complexity $O(kT)$. Hence, the algorithm for stationary maximum dynamic flow have complexity $O(n^2m^3 \log n)$ (we consider that k$T \leq n^2m^3 \log n$). The flow obtained with ASMDF is called a temporally repeated flow for the obvious reason that is consists of repeated shipments along the same flow paths from 1 to n. The maximum value of a temporally repeated flow obtained with ASMDF is:

$$v_H = (T + 1)v - \sum_A h(i, j)f(i, j) \tag{10}$$

where v is the maximum value of the flow f obtained with AMVMCF.

4 The maximum flows in planar dynamic networks

In this section we consider the maximum flows in planar dynamic networks in the stationary case. Hence, we use the ASMDF which has presented in Section 3. The network $G = (N, A, c, u)$ is planar.

The minimum mean cycle canceling algorithm is a special case of the Klein′s algorithm (cycle canceling algorithm, see [1]). Recall that the mean cost of a directed cycle (circuit) $\mathring{P}$ is $(\sum(c(i, j)|(i, j) \in \mathring{P}))/\left|\mathring{P}\right|$, and that the minimum mean cycle is a cycle with the smallest mean cost in the network G. Is known that use dynamic programing algorithm to find the minimum mean cycle in $O(nm)$ time, see [1].

In this case, the minimum mean cycle canceling algorithm starts with a maximum flow f in the network G. This flow is computed with algorithm presented in Section 2. At every iteration, the minimum mean cycle canceling algorithm identifies a minimum mean cycle $\mathring{P}$ in residual network $\tilde{G}$. If the mean cost of the cycle $\mathring{P}$ is negative, the algorithm augments the maximum possible flow along $\mathring{P}$, updates $\tilde{G}$, and repeats this process. If the mean cost of $\mathring{P}$ is nonnegative, $\tilde{G}$ contains no negative cycle and f is a maximum value and minimum cost flow, so the algorithm terminates. This algorithm is surprisingly simple to state.

Theorem 6. *The ASMDS correctly computes the maximum flow in planar stationary dynamic network.*

Proof: The ASMDS correctly computes the maximum flow in general stationary dynamic network. Obviously that algorithm is correctly and for planar network. □

Theorem 7. *The ASDMS applied in planar network has the complexity $O(n^5 logn)$.*

Proof: The ASMDF applied in general network has the complexity $O(n^2m^3 logn)$. In planar network we have $m = O(n)$. Hence, the ASMDF applied in planar network has the complexity $O(n^5 \log n)$. □

5 Conclusions

The computation of a maximum flow in a general network has been an important and well studied problem, both in the fields of computer science and operations research. Many efficient

algorithms have been developed to solve this problem, see, e.g., [1]. Research on maximum flow in planar network is motivated by the fact that more efficient algorithms can be developed by exploiting the planar structure of the graph. The planar flow algorithms are not only extremely efficient, but they are also very elegant. Planar networks also arise in practical contexts such as VLSI design and communication networks, and hence it is of interest to find fast flow algorithms for this class of networks.

In this paper, we have studied a generalization of the maximum flow in directed $(1, n)$ planar networks, to include transit time features encountered in many practical situations. The our model, assumes that all attributes in the problem, including arc capacities and transit times, do not change over time. In this case we have used an efficient procedure to find the maximum value and minimum cost flow in directed $(1, n)$ planar static networks $G = (N, A, c = h, u)$, and then develops a set of temporally repeated flows, with the optimal flow decomposed into a set of path flows. We remark that the problem of maximum flow in $(1, n)$ planar dynamic networks was not studied up to the present. Also, we introduce the notion of reduced time expanded network $G_H^{'} = (N_H^{'}, A_H^{'}, u_H^{'})$ and show how make this network.

Future research directions include problems:

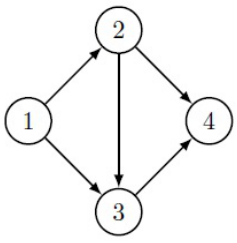
(1) the maximum flow in directed $(1, n)$ planar dynamic networks, where the transit times, the capacities of arcs are all time-varying;

(2) the maximum flow in directed $(1, n)$ planar dynamic networks with lower bounds in stationary case and in nonstationary case.

These are more practical features in many real-world problems where we desire to control the speed of flows at different arcs.

6 Example

The planar dynamic network is presented in Figure 3(a) and time horizon being set to $T = 4$, therefore $H = \{0, 1, 2, 3, 4\}$. The transit times $h(i, j)$ and the upper bounds (capacities) $u(i, j)$ for all arcs are indicate in Figure 3(b).



(a)

(i, j)	$(1, 2)$	$(1, 3)$	$(2, 3)$	$(2, 4)$	$(3, 4)$
$h(i, j)$	1	3	1	2	1
$u(i, j)$	3	2	1	2	2

(b)

Figure 3: The planar dynamic network

Figure 4 shows the $(1^{'}, 4^{'})$ dual network $G^{'} = (N^{'}, A^{'}, c^{'})$ corresponding to network G_h.

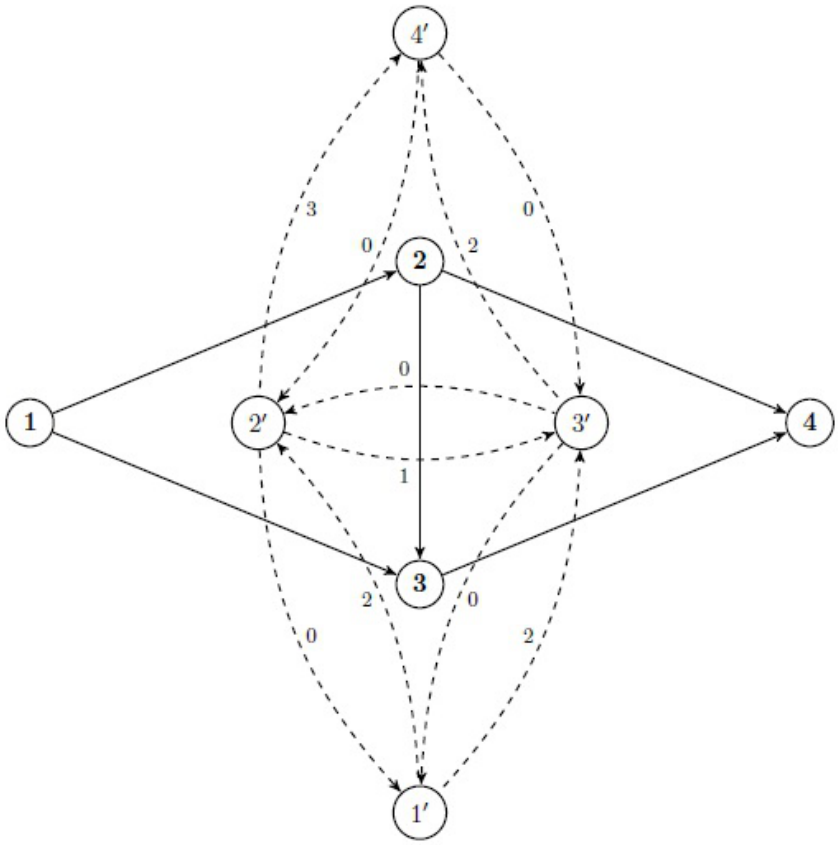


Figure 4: Dual network $G^{'}$ corresponding to the network G

The flow obtained with AMFDPSN is presented in Figure 5(a).

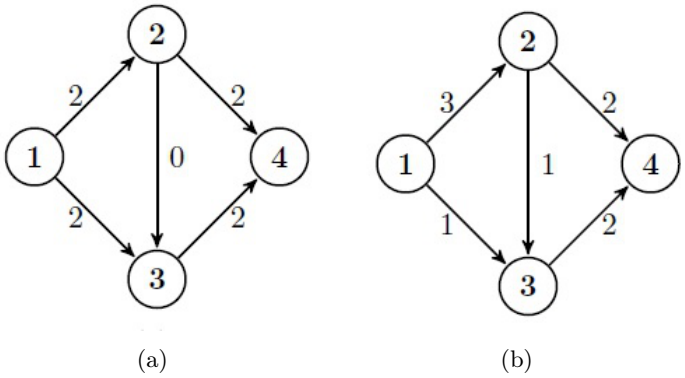


Figure 5: (a) maximum flow; (b) maximum flow of minimum cost

With the minimum mean cycle canceling algorithm we obtain the maximum flow of minimum cost and is presented in Figure 5(b). Applying the procedure ADFEF we have the following path: $P_1 = (1, 2, 4), h(P_1) = 3, r(P_1) = 2$; $P_2 = (1, 2, 3, 4), h(P_2) = 3, r(P_2) = 1$; $P_3 = (1, 3, 4), h(P_3) = 4, r(P_3) = 1$. With the procedure ARF we obtain the maximum dynamic flow which is shown in network $G_H^{'} = (N_H^{'}, A_H^{'}, u_H^{'})$ in Figure 6.

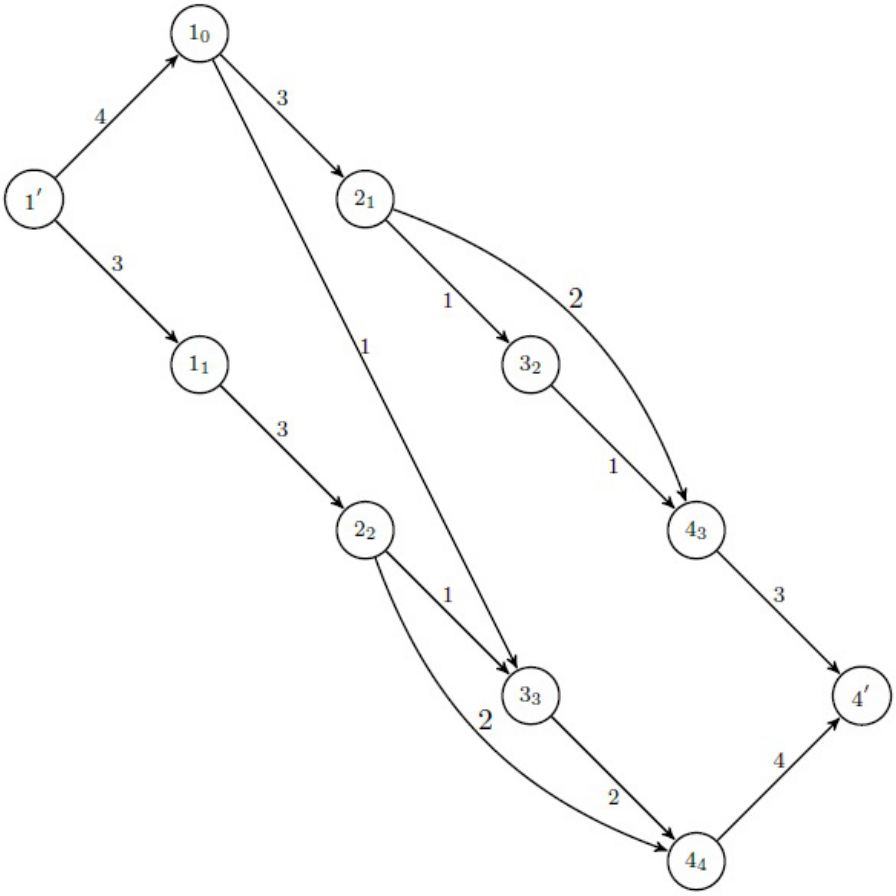


Figure 6: The maximum dynamic flow

Applying the formula (10) we have $v_H = (4+1)4 - (3+3+1+4+2) = 7$.

For $S_H^{'} = \left\{1^{'}, 1_0, 1_1, 2_2, 3_3\right\}$, $\bar{S}_H^{'} = \left\{2_1, 3_2, 4_3, 4_4, 4^{'}\right\}$ we have $\left[S_H^{'}, \bar{S}_H^{'}\right] = (S_H^{'}, \bar{S}_H^{'}) = \{(1_0, 2_1), (2_2, 4_4), (3_3, 4_4)\}$ and $v_H^{'} = v_H = f_H^{'}(S_H^{'}, \bar{S}_H^{'}) = u_H^{'}(S_H^{'}, \bar{S}_H^{'}) = 3 + 2 + 2 = 7$.

Bibliography

[1] Ahuja,R.; Magnanti,T.; Orlin,J. (1993); *Network Flows. Theory, algorithms and applications*, Editing Prentice hall, Inc.,Englewood Clifss, New Jersey.

[2] Cai,X.; Sha,D.; Wong,C. (2007); *Time-varying Network Optimization*, Editing Springer.

[3] Ciurea,E. (1984); Les problemes des flots dynamiques, *Cahiers du CERO*, 26(1-2): 3-9.

[4] Frederickson,G. (1987); Fast algorithms for shortest path in planar graphs, with applications, *SIAM Journal on Computing*, 16: 1004-1022.

[5] Ford,L.; Fulkerson,D. (1962); *Flows in Networks*, Princeton University Press, Princeton, N.J.

[6] Hassin,R. (1981); Maximum flows in (s,t) planar networks, *Information Processing letters*, 13: 107.

[7] Hassin,R.; Johnson,D. (1985); An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks, *SIAM Journal on Computing*, 14: 612-624.

[8] Itai, A.; Shiloach, Y. (1979); *Maximum flow in planar networks*, SIAM Journal on Computing, 8: 135-150.

[9] Johnson, D.; Venkatesan,S. (1982); Using divide and conquer to find flows in directed planar networks in $O(n^{1.5} \log n)$ time, *Proceedings of the 20th Annual Allerton Conference on Communication , Control and Computing*, University of Illinois, Urbana-Champaign, IL., 898-905.

[10] Khuller,S.; Naor,J. (1994); Flows in planar graphs with vertex capacities, *Algorithmica*, 11: 200-225.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(2):292-304, April 2016.

Numerical P Systems with Thresholds

Z. Zhang, L. Pan

Zhiqiang Zhang
Key Laboratory of Image Information Processing and Intelligent Control,
School of Automation, Huazhong University of Science and Technology,
Wuhan 430074, Hubei, China
zhiqiangzhang@hust.edu.cn

Linqiang Pan*
Key Laboratory of Image Information Processing and Intelligent Control,
School of Automation, Huazhong University of Science and Technology,
Wuhan 430074, Hubei, China
*Corresponding author: lqpan@mail.hust.edu.cn

Abstract: Numerical P systems are a class of P systems inspired both from the structure of living cells and from economics. In this work, a control of using evolution programs is introduced into numerical P systems: a threshold is considered and a program can be applied only when the values of the variables involved in the production function of the program are greater than/equal to (lower-threshold) or smaller than/equal to (upper-threshold) the threshold. The computational power of numerical P systems with lower-threshold or upper-threshold is investigated. It is proved that numerical P systems with a lower-threshold, with one membrane and linear production functions, working both in the all-parallel mode and in the one-parallel mode are universal. The result is also extended to numerical P systems with an upper-threshold, by proving the equivalence of the numerical P systems with lower- and upper-thresholds.
Keywords: membrane computing, numerical P system, computation power, universality, register machine.

1 Introduction

Membrane computing is a branch of natural computing, which is inspired from the structure and functioning of living cells. The computing devices considered in membrane computing are called *P systems*. They are parallel, distributed and non-deterministic computational models. According to their membrane structure, there are two main classes of P systems: *cell-like P systems*, with a hierarchical arrangement of membranes [7], and *tissue-like P systems* or *neural-like P systems*, with a net of processor units placed in the nodes of a directed graph [3, 5]. The present work deals with a class of cell-like P systems, called *numerical P systems* [10].

Numerical P systems are motivated by the cell structure and the economic reality. Numerical variables are placed in the regions of a membrane structure. These variables can evolve by means of programs, which are composed of two components, a *production function* and a *repartition protocol*. A production value of the region at a given step is computed by means of the production function. This value is distributed to variables from the region where the program resides, and to variables in its upper and lower neighbors according to the repartition protocol. By a synchronized use of production functions, followed by the repartition of the obtained values, a transition is defined between system configurations. The values assumed by a distinguished variable during a computation form the set of numbers computed by the system.

Many computational properties of numerical P systems have been investigated at both the theoretical level and at the application level [4,9,10,12–17]. Several strategies of using production-repartition programs were considered: sequential (at each step, in each region, only one program

can be applied), all-parallel (all programs in a region of the membrane structure are used simultaneously, with each variable participating in all programs where it appears), one-parallel (the programs are chosen to be used in parallel in such a way that each variable participates in only one of the chosen programs).

Using a threshold is an interesting strategy of controlling the use of production-repartition programs. The idea was introduced in numerical P systems in [12], under the name of *enzymatic control*: a distinguished variable, called enzyme, is associated with each program and the program is applied only if the current value of the enzyme is not smaller than the smallest value of the variables involved in the production function of the program. The "enzymatic control" is useful in designing robot controllers based on numerical P systems [13–15].

We here introduce a related but different strategy, similar to the threshold control used in [18] for spiking neural P systems: rules can be used according to the result of the comparison of the number of spikes in the neuron with the constant, which corresponds to the fact that a neuron can fire when its potential is greater than or equal to its threshold. In our case, a constant is associated with the numerical P system and it is used as a control threshold in two natural ways: a program can be applied only when the values of the variables involved in the production function are not smaller than (the lower-threshold case), respectively not greater than (the upper-threshold case) the constant. The computational power of such P systems is investigated. Specifically, it is proved that universality results can be obtained for such P systems with one membrane and linear production functions, working both in the all-parallel mode and in the one-parallel mode. The proof is done (by simulating register machines) only for lower-thresholds, then the result is extended to the case of upper-threshold by proving that numerical P systems with upper-thresholds can simulate systems with lower-thresholds.

The possible usefulness of the threshold control remains to be examined for applications (in robot control). For the sake of applications, it could be useful to consider its stronger versions, such as taking different thresholds for different membranes or even for different programs in the system (maybe also mixing the way to use the thresholds, in the lower or upper ways).

2 Preliminaries

Readers are assumed to be familiar with basic elements of membrane computing, e.g., from [7, 8, 11]. Here we only mention some notions and notations which are used in this paper.

We denote by $\mathbb{N}$ the set of natural numbers, and the set of real numbers is denoted by $\mathbb{R}$. The family of all recursively enumerable sets of k-dimensional vectors of non-negative integers is denoted by $Ps(k)RE$. Since numbers can be seen as one-dimensional vectors, we can replace $Ps(1)$ by N in the notation, thus obtaining NRE.

An *n-register machine* is a construct $M = (n, P, m)$, where $n > 0$ is the number of registers, P is a finite sequence of instructions bijectively labeled with the elements of the set $\{0, 1, \ldots, m\}$, 0 is the label of the first instruction to be executed, and m is the label of the halt instruction of P. Registers contain non-negative integer values. The instructions of P have the following forms:

- $j : (INC(r), k, l)$, with $0 \leq j < m, 0 \leq k, l \leq m$, and $1 \leq r \leq n$.
  This instruction, labeled with j, increments the value contained in register r, then non-deterministically jumps either to instruction k or to instruction l.

- $j : (DEC(r), k, l)$, with $0 \leq j < m, 0 \leq k, l \leq m$, and $1 \leq r \leq n$.
  If the value contained in register r is positive, then decrement it by 1 and jump to instruction k. If the value of r is zero, then jump to instruction l (without altering the content of the register).

- m: *Halt.*

A *deterministic* register machine is a register machine in which all INC instructions have the form $j : (INC(r), k, k)$; we write these instructions simply as $j : (INC(r), k)$.

A register machine M generates a set $N(M)$ of numbers in the following way: the machine starts with all registers being empty (i.e., storing the number zero); the machine applies the instruction with label 0 and continues to apply instructions as indicated by the labels (and made possible by the contents of registers); if it reaches the halt instruction, then the number present in register 1 at that time is said to be generated by M. If the computation does not halt, then no number is generated. It is known that register machines generate all sets of numbers which are Turing computable, hence they characterize NRE [6].

A register machine can also be used to compute functions. A function $f : \mathbb{N}^\alpha \to \mathbb{N}^\beta$ is computed by a register machine M if, when starting with n_1 to n_α in registers 1 to α, if $f(n_1, \ldots, n_\alpha) = (r_1, \ldots, r_\beta)$, then M halts in the final label m with registers 1 to β containing r_1 to r_β, and all other registers being empty; if $f(n_1, \ldots, n_\alpha)$ is undefined, then the final label of M is never reached.

A register machine can also be used as an accepting device. A set N of numbers is accepted by a deterministic register machine M if, when starting with $x \in N$ in register 1, M halts in the final label m with all registers being empty.

The following propositions concerning the computational power of register machines are essential for the main results established in this work [1, 2, 6].

Proposition 1. *For any partial recursive function f: $\mathbb{N}^\alpha \to \mathbb{N}^\beta (\alpha, \beta > 0)$, there exists a deterministic register machine M with $(max\{\alpha, \beta\} + 2)$ registers computing f.*

Proposition 2. *For any recursively enumerable set $N \subseteq Ps(\alpha)RE$ of vectors of non-negative integers there exists a deterministic register machine M with $(\alpha + 2)$ registers accepting N.*

Proposition 3. *For any recursively enumerable set $N \subseteq Ps(\beta)RE$ of vectors of non-negative integers there exists a non-deterministic register machine M with $(\beta + 2)$ registers generating N.*

3 Numerical P Systems with Thresholds

We introduce the class of numerical P systems to be investigated in this work. The definition is general, for the computing case.

A numerical P system with a threshold is a construct

$$\Pi = (m, H, \mu, T, (Var_1, Pr_1, Var_1(0)), \ldots, (Var_m, Pr_m, Var_m(0)), Var_{in}, Var_{out}),$$

where

- $m \geq 1$ is the number of membranes;
- H is an alphabet of labels for membranes in μ;
- μ is a rooted tree with q nodes labeled with the elements of H;
- T is a constant, called threshold;
- Var_i, $1 \leq i \leq m$, is the set of variables in region i;
- $Var_i(0)$, $1 \leq i \leq m$, is the set of initial values of the variables in region i;

- Pr_i, $1 \leq i \leq m$, is the set of programs in region i; each program has the form

$$F_{l,i}(x_{1,i}, \ldots, x_{k_i,i})|_T \rightarrow c_{l,i,1}|v_{l,i,1} + \cdots + c_{l,i,l_i}|v_{l,i,l_i},$$

  where $F_{l,i}(x_{1,i}, \ldots, x_{k_i,i})$ is the production function, and $c_{l,i,1}|v_{l,i,1} + \cdots + c_{l,i,l_i}|v_{l,i,l_i}$ is the repartition protocol of the program;

- Var_{in} and Var_{out} are the sets of input and of output variables, respectively.

The programs allow the system to evolve the values of variables during computations. Each program is composed of two parts: a production function and a repartition protocol. The former can be any function using variables from the region that contains the program. Only polynomial functions are considered here. By using the production functions in each region, the system computes a production value from the values of its variables at that time. This value is distributed to variables from the region where the program resides, and to variables in its upper (parent) and lower (children) compartments, as specified by the repartition protocol.

The programs are applied under the control of the threshold T, according to two strategies: bounding the values of variables from below (lower-threshold) and bounding them from above (upper-threshold).

More precisely, in the first case a program can be applied only when the current value of each variable from its production function is greater than or equal to the threshold T. Dually, in the upper-threshold case, a program can be applied only when the current value of each variable from its production function is smaller than or equal to the threshold T.

The repartition of the "production" takes place as follows. For a repartition protocol $RP_{l,i}$, variables $v_{l,i,1}, \ldots, v_{l,i,l_i}$ come from the membrane i where the program resides, the parent membrane and the children membrane. Formally, $\{v_{l,i,1}, \ldots, v_{l,i,l_i}\} \subseteq Var_i \cup Var_{par(i)} \cup (\bigcup_{ch \in Ch(i)} Var_{ch})$, where $par(i)$ is the parent of membrane i and $Ch(i)$ is the set of children of membrane i. The coefficients $c_{l,i,1}, \ldots, c_{l,i,l_i}$ are natural numbers (they may be also 0, in which case the terms "$+0|x$" are omitted), which specify the proportion of the current production value distributed to each variable $v_{l,i,1}, \ldots, v_{l,i,l_i}$. At time t, if we denote with $C_{l,i} = \sum_{s=1}^{l_i} c_{l,i,s}$ the sum of all coefficients of the repartition protocol, and denote with

$$q_{l,i}(t) = \frac{F_{l,i}(x_{1,i}(t), \ldots, x_{k_i,i}(t))}{C_{l,i}} \tag{1}$$

the "unitary portion", then the value $ad_{l,i,r}(t) = q_{l,i}(t) \cdot c_{l,i,r}$ represents the value added to variable $v_{l,i,r}$. If variable $v_{l,i,r}$ appears in several repartition protocols, for example, $RP_{l_1,i_1}, \ldots, RP_{l_k,i_k}$, all these values $ad_{l_1,i_1,r}, \ldots, ad_{l_k,i_k,r}$ are added to variable $v_{l,i,r}$. After computing the production function value, the variables involved in the production function are reset to zero. So, if at time t variable $v_{l,i,r}$ is involved in at least one production function, its value at time $t+1$ is $v_{l,i,r}(t+1) = \sum_{s=1}^{k} ad_{l_s,i_s,r}(t)$; otherwise, $v_{l,i,r}(t+1) = v_{l,i,r}(t) + \sum_{s=1}^{k} ad_{l_s,i_s,r}(t)$.

Such a system evolves in the all-parallel mode (at each step, in each membrane, all programs which can be applied are applied, allowing that more than one program share the same variable) or in the one-parallel mode (apply programs in the all-parallel mode with the restriction that one variable can appear in only one of the applied programs). A configuration represents the values of all system's variables at a given computation step. Initially, the variables have the values specified by $Var_i(0), 1 \leq i \leq m$. Using the programs in the way mentioned above, a transition of the system from a configuration to the next one is defined. A sequence of such transitions forms a computation. If no program in each region can be applied, we say that the system reaches a *halting configuration.*

In this way, a numerical P system can compute a function f: $\mathbb{N}^{\alpha} \to \mathbb{N}^{\beta}(\alpha, \beta \geq 0)$: the α values of the arguments are introduced in the system as the initial values of variables in Var_{in} and the β-vector of the function value is obtained in the variables from Var_{out} in the halting configuration of the system. If the system never reaches a halting configuration, then no result is obtained.

By ignoring the input variables, (non-deterministic) numerical P systems with thresholds can also be used in the *generating mode*, whereas by ignoring the output variables we can use (deterministic or non-deterministic) numerical P systems with thresholds in the *accepting mode*.

Note that $q_{j,i}(t)$ are integers only if the value of the production functions $F_{j,i}(x_{1,i}(t), \ldots, x_{k_i,i}(t))$ is divisible by the respective sums $C_{j,i}(t)$. If at any step, all the values of the production functions are divisible by the respective sums, we associate this kind of systems with the notation *div*. If a current production is not divisible by the associated coefficients total, then we can take the following decisions [10]: (i) the remainder is lost (the production which is not immediately distributed is lost), (ii) the remainder is added to the production obtained in the next step (the non-distributed production is carried over to the next step), (iii) the system simply stops and aborts, no result is associated with that computation. We denote these three cases with *lost, carry, stop*, respectively. In this paper, the numerical P systems with thresholds that we construct are of the *div* type.

The set of natural numbers generated or accepted in the way mentioned above by a system Π is denoted by $N_{\alpha}(\Pi), \alpha \in \{gen, acc\}$. We use $N_{\alpha}T_{\gamma}P_m^D(poly^n(r), \beta)$ to denote the family of all sets $N_{\alpha}(\Pi)$ of numbers computed by systems Π working in α mode, with at most m membranes, production functions which are polynomials of degree at most n, with integer coefficients, with at most r variables in each polynomial, using the rules in the mode $\beta \in \{all, one\}$, where *all* stands for all-parallel, and *one* stands for one-parallel, and with the threshold used in the $\gamma \in \{l, u\}$ way, with l indicating the lower-threshold case and u indicating the upper-threshold case; the letter D indicates the use of deterministic systems (we remove D when the systems may also be non-deterministic). If one of the parameters m, n, r is not bounded, then we replace it with $*$.

4 The Universality of Numerical P Systems with Lower-thresholds

In this section, we investigate the computational power of numerical P systems with lower-thresholds working in the all-parallel mode and in the one-parallel mode.

Theorem 4. *Each partial recursive function $f : \mathbb{N}^{\alpha} \to \mathbb{N}^{\beta}$ $(\alpha > 0, \beta > 0)$ can be computed by a deterministic numerical P system with a lower-threshold, with only one membrane, using linear production functions that use each at most three variables, and working in the all-parallel mode.*

Proof: Let $M = (n, P, m)$ be a deterministic register machine with n registers, computing function f. The initial instruction of M has the label 0 and the machine halts only if the instruction with label m is reached. According to Proposition 1, $n = max\{\alpha, \beta\} + 2$ is enough. Before the computation starts, let us assume that the values of the first α registers are equal to $r_1, \ldots, r_{\alpha}$. When the computation halts, the values stored in the registers $1, \ldots, \beta$ are the values computed by $f(r_1, \ldots, r_{\alpha})$.

We construct the following numerical P system with a lower-threshold:

$$\Pi_M = (1, \{0\}, [_0\ ]_0, 1, (Var_0, Pr_0, Var_0(0)), Var_{in}, Var_{out}),$$

where

- $Var_0 = \{x_{i,1}, x_{i,2}, p_j \mid 1 \leq i \leq n, 0 \leq j \leq m\}$;

- $Var_0(0)$ is the vector of initial values of the variables, with:
  - $x_{i,1} = x_{i,2} = r_i$, for all $1 \leq i \leq \alpha$;
  - $x_{i,1} = x_{i,2} = 0$, for all $1 + \alpha \leq i \leq n$;
  - $p_j = 0$, for all $0 \leq j \leq m$ with the exception of $p_0 = 1$;
- $Pr_0 = \{3p_j|_1 \to 1|x_{i,1} + 1|x_{i,2} + 1|p_k$, for all instructions $j : (INC(i), k) \in P\}$
  $\cup \{p_j|_1 \to 1|p_l,$
  $x_{i,1} - x_{i,2} - p_j|_1 \to 1|p_l,$
  $x_{i,1} - x_{i,2} + p_j|_1 \to 1|p_k,$
  $2(x_{i,1} - p_j)|_1 \to 1|x_{i,1} + 1|x_{i,2}$, for all instructions $j : (DEC(i), k, l) \in P\}$;
- $Var_{in} = \{x_{1,i}, \ldots, x_{\alpha,i} \mid 1 \leq i \leq 2\}$;
- $Var_{out} = \{x_{1,1}, \ldots, x_{\beta,1}\}$.

Note that the threshold is equal to 1. The value of register i $(1 \leq i \leq n)$ is encoded by variables $x_{i,1}$ and $x_{i,2}$. The values of $x_{i,1}$ and $x_{i,2}$ are always equal. The input values $r_1, \ldots, r_\alpha$ are set as the initial values of variables $x_{1,i}, \ldots, x_{\alpha,i}$ $1 \leq i \leq 2$, respectively. Variables $p_0, \ldots, p_m$ are used to indicate the instruction to be simulated. During the computation, the values of $p_0, \ldots, p_m$ are equal to 1 or 0 (at most one of them is equal to 1 in each step, and this indicates that the system starts to simulate the corresponding instruction of M).

The increment instruction $j : (INC(i), k)$ is simulated by the program $3p_j|_1 \to 1|x_{i,1} + 1|x_{i,2} + 1|p_k$ in one step. When $p_j = 0$, the program cannot be applied because the value of p_j is smaller than the threshold 1. When $p_j = 1$, the program can be applied since the value of p_j is equal to the threshold. After the application of this program, each of the variables $x_{i,1}, x_{i,2}, p_k$ obtains a portion 1, and variable p_j is reset to zero. Variable $p_k = 1$ indicates that the instruction labeled k will be simulated at the next step, the increment of variables $x_{i,1}, x_{i,2}$ corresponds to the increase of the number stored in register i by 1. So, the increment instruction $j : (INC(i), k)$ has been correctly simulated.

The decrement instruction $j : (DEC(i), k, l)$ is simulated in one step by the following four programs:

$$p_j|_1 \to 1|p_l, \tag{2}$$

$$x_{i,1} - x_{i,2} - p_j|_1 \to 1|p_l, \tag{3}$$

$$x_{i,1} - x_{i,2} + p_j|_1 \to 1|p_k, \tag{4}$$

$$2(x_{i,1} - p_j)|_1 \to 1|x_{i,1} + 1|x_{i,2}. \tag{5}$$

When a decrement instruction $j : (DEC(i), k, l)$ starts to be simulated, which means that $p_j = 1$, there are the following two cases.

- $p_j = 1$, $x_{i,1} = x_{i,2} = 0$. In this case, only program (2) satisfies the threshold condition. After applying program (2), variable p_j is set to zero, and variable p_l receives a contribution 1, which indicates the next instruction to be simulated. Programs (3)–(5) cannot be applied since variables $x_{i,1}$ and $x_{i,2}$ are zero (smaller than the threshold 1). Hence the values of $x_{i,1}$ and $x_{i,2}$ are not changed, and the computation continues with the simulation of instruction l of register machine M.
- $p_j = 1$, $x_{i,1} = x_{i,2} \geq 1$. In this case, all the four programs satisfy the threshold condition, thus all of them can be applied. Program (4) transfers the production value 1 to variable p_k, which indicates the next instruction k to be simulated. By using program (5), variables $x_{i,1}$

and $x_{i,2}$ are decreased; their values are first zeroed and each of them receives a contribution of their former value minus one. The role of program (3) is to cancel the effect of program (2). Program (2) transfers the value of p_j to p_l, thus p_l gets a contribution of 1, which is canceled by program (3) by sending a contribution of -1 to p_l. Hence the values of variables $x_{i,1}$ and $x_{i,2}$ are decremented by one and the next instruction to be simulated is the one labeled with k.

After the simulation of any instruction of M, the values of both variables $x_{i,1}$ and $x_{i,2}$ are equal to the contents of register i ($1 \leq i \leq n$), while only one of variables $p_0, \ldots, p_m$ is equal to 1, indicating the next instruction of M to be simulated. When M reaches the halt instruction, the corresponding value of variable p_m is equal to 1. Since no program contains the variable p_m in the production function, Π_M reaches a final configuration; the result of the computation is the values of variables $x_{1,1}, \ldots, x_{\beta,1}$. □

According to Proposition 2, for any recursively enumerable set $N \subseteq Ps(\alpha)RE$ of vectors of non-negative integers there exists a deterministic register machine M with $(\alpha + 2)$ registers accepting N. For this register machine M, following the proof in Theorem 4, we can construct a numerical P system with a lower-threshold that accepts N. So, the following corollary holds.

Corollary 5. $Ps(\alpha)RE = N_{acc}T_lP_1^D(poly^1(3), all)$.

For numerical P systems with lower-thresholds working in the one-parallel mode, the following similar results hold.

Theorem 6. *Each partial recursive function $f : \mathbb{N}^\alpha \to \mathbb{N}^\beta$ $(\alpha, \beta > 0)$ can be computed by a one-membrane numerical P system with a lower-threshold working in the one-parallel mode, having linear production functions that use each at most five variables.*

Proof: We proceed like in the proof of Theorem 4, with the difference that here we simulate both deterministic and non-deterministic register machines. Let $M = (n, P, m)$ be a non-deterministic register machine with $n = \max\{\alpha, \beta\}+2$ registers, computing the function f. As usual, the input values $r_1, \ldots, r_\alpha$ are stored in the first α registers before the computation starts, with all the other registers being empty. When the computation halts, the values $f(r_1, \ldots, r_\alpha)$ will be found in registers $1, \ldots, \beta$.

The numerical P system with a lower-threshold to simulate register machine M is constructed as follows.

$$\Pi_M = (1, \{0\}, [_0\,]_0, 1, (Var_0, Pr_0, Var_0(0)), Var_{in}, Var_{out}),$$

where

- $Var_0 = \{p_{j,g}, x_{i,g} \mid 1 \leq g \leq 5,\ 1 \leq i \leq m, 0 \leq j \leq n\}$;
- $Var_0(0)$ is the vector of initial values of the variables, with:
  - $x_{i,g} = r_i$, for all $1 \leq i \leq \alpha, 1 \leq g \leq 5$;
  - $x_{i,g} = 0$, for all $1 + \alpha \leq i \leq n, 1 \leq g \leq 5$;
  - $p_{j,g} = 0$, for all $0 \leq j \leq m, 1 \leq g \leq 5$;
  - $p_{0,g} = 1$, for all $1 \leq g \leq 5$;

- $Pr_0 = \{2\sum_{g=1}^{5} p_{j,g}|_1 \to \sum_{g=1}^{5} 1|x_{i,g} + \sum_{g=1}^{5} 1|p_{k,g},$
  $2\sum_{g=1}^{5} p_{j,g}|_1 \to \sum_{g=1}^{5} 1|x_{i,g} + \sum_{g=1}^{5} 1|p_{l,g};$
  for all instructions $j : (INC(i), k, l) \in P\}$
  $\cup \{5(x_{i,1} - p_{j,1})|_1 \to \sum_{g=1}^{5} 1|x_{i,g},$
  $8(x_{i,2} - x_{i,3} + p_{j,2})|_1 \to \sum_{g=1}^{5} 1|p_{k,g} + \sum_{g=1}^{3} 1|p_{j,g},$
  $-5(x_{i,4} - x_{i,5} + p_{j,3})|_1 \to \sum_{g=1}^{5} 1|p_{l,g},$
  $5p_{j,4}|_1 \to \sum_{g=1}^{5} 1|p_{l,g},$
  $-3p_{j,5}|_1 \to \sum_{g=1}^{3} 1|p_{j,g}$; for all instructions $j : (DEC(i), k, l) \in P\}$;

- $Var_{in} = \{x_{i,g} \mid 1 \le g \le 5, 1 \le i \le \alpha\}$;

- $Var_{out} = \{x_{1,1}, \ldots, x_{\beta,1}\}$.

In order to ensure that at each step only one variable can appear in the production functions of the applied programs, the value of register i $(1 \le i \le n)$ is contained in five variables $x_{i,g}1 \le g \le 5$, and the system uses five variables $p_{i,g}, 1 \le g \le 5$, to control the simulation of the instruction with label i of register machine M (in the following, for brevity, we use $x_{i,g}$ and $p_{i,g}$ to represent all the five variables for $1 \le g \le 5$, respectively). During the computation, variables $x_{i,g}$ are always equal to each other, and the same holds for variables $p_{i,g}$. The input values r_i $(1 \le i \le \alpha)$ are introduced into the system as the initial values of variables $x_{i,g}$ $(1 \le i \le \alpha)$, respectively. When the instruction i is simulated, all the five variables $p_{i,g}$ are equal to 1, while all the others are zero.

The simulation of an increment instruction $j : (INC(i), k, l)$ is done in one step by the following two programs:

$$2\sum_{g=1}^{5} p_{j,g}|_1 \to \sum_{g=1}^{5} 1|x_{i,g} + \sum_{g=1}^{5} 1|p_{k,g}, \tag{6}$$

$$2\sum_{g=1}^{5} p_{j,g}|_1 \to \sum_{g=1}^{5} 1|x_{i,g} + \sum_{g=1}^{5} 1|p_{l,g}. \tag{7}$$

If $p_{j,g} = 0$, then programs (6) and (7) cannot be executed since the values of variables $p_{j,g} = 0$ are smaller than the thresholds 1. If $p_{j,g} = 1$, then only one of programs (6) and (7) can be applied because their production functions share the same variables (the system works in the one-parallel mode). If program (6) (resp., program (7)) is applied, then variable $x_{i,g}$ is increased by one, setting $p_{k,g}$ (resp., $p_{l,g}$) to 1, thus the system starts to simulate instruction k (resp., instruction l), and resetting variables $p_{j,g}$ to zero. If M is deterministic, then the simulation of the instruction $j : (INC(i), k)$ is performed by using the program (6). In this case, no competition occurs between the programs, and so the simulation is deterministic.

The simulation of a decrement instruction $j : (DEC(i), k, l)$ is done in one step by the

following five programs:

$$5(x_{i,1} - p_{j,1})|_1 \to \sum_{g=1}^{5} 1|x_{i,g}, \tag{8}$$

$$8(x_{i,2} - x_{i,3} + p_{j,2})|_1 \to \sum_{g=1}^{5} 1|p_{k,g} + \sum_{g=1}^{3} 1|p_{j,g}, \tag{9}$$

$$-5(x_{i,4} - x_{i,5} + p_{j,3})|_1 \to \sum_{g=1}^{5} 1|p_{l,g} \tag{10}$$

$$5p_{j,4}|_1 \to \sum_{g=1}^{5} 1|p_{l,g}, \tag{11}$$

$$-3p_{j,5}|_1 \to \sum_{g=1}^{3} 1|p_{j,g}. \tag{12}$$

If $p_{j,g} = 0$, then programs (8)–(12) cannot be applied, because $p_{j,g} = 0$ are smaller than the threshold. So, when $p_{j,g} = 0$, no undesirable simulation steps can appear.

If $p_{j,g} = 1, x_{i,g} = 0$, then the values of $x_{i,g}$ should remain unchanged, and the computation should jump to the simulation of instruction l, which is realized by programs (11) and (12) in one step. (Note that in this case programs (8) – (10) cannot be applied, for the values of $x_{i,g}$ are smaller than the thresholds.) The effect of program (11) is to reset variables $p_{j,4}$ to zero and to give a contribution 1 to each of variables $p_{l,g}$, whose values are 1 after the application of this program, thus correctly simulating the passing to instruction l. At the same time, program (12) is applied, the role of which is to set all the variables $p_{j,g}$ ($g \neq 4$) to zero. Variable $p_{j,5}$ appears in the production function, so its initial value is canceled, and it receives no contribution, hence its final value is zero. For variables $p_{j,1}$, $p_{j,2}$ and $p_{j,3}$, their initial values are 1 and receive contribution -1, hence their final values are zero, which is also correct.

If $p_{j,g} = 1, x_{i,g} \geq 1$, then the values of $x_{i,g}$ should be decremented and the computation should proceed to the simulation of instruction k. In this case, all the five programs (8)–(12) can be applied. Programs (8) and (9) decrement the values of $x_{i,g}$ and increment the values of $p_{k,g}$ by 1. The other programs have auxiliary roles. Note that all the variables $p_{j,g}$ and $x_{i,g}$ appear in the production functions of programs (8)–(12), so their values are first reset to zero, and their final values will be the sum of all the contributions they receive. Variables $x_{i,g}$ only appear in the repartition protocol of program (8), which gives a contribution of their initial value minus 1, thus correctly decrementing their values by one. Variables $p_{j,1}$, $p_{j,2}$ and $p_{j,3}$ receive a contribution 1 from program (9) and a contribution -1 from program (12), thus their values will be equal to 0. Variables $p_{j,4}$ and $p_{j,5}$ do not appear in any repartition protocol of programs (8) – (12), thus their final values are zero. The role of program (10) is to cancel the effect of program (11). Program (11) sends a contribution 1, and simultaneously program (10) sends a contribution -1, to each of variables $p_{l,g}$, whose final values are hence equal to 0. Each of variables $p_{j,g}$ receives a contribution 1, thus their final values are 1, which is also correct.

After the simulation of each instruction of M, all the variables $x_{i,g}$ are equal to the contents of register i ($1 \leq i \leq n$), while the variables $p_{j,g}$ ($0 \leq j \leq m$) correctly indicate the next instruction of M to be simulated. When the program counter of M reaches the value k, the corresponding values of variables $p_{k,g}$ is equal to 1. When the program counter of M reaches the value m, the corresponding values of variables $p_{m,g}$ are equal to 1. Since no program contains variables $p_{m,g}$ in the production function, Π_M reaches a halting configuration; the result of the computation is values of variables $x_{1,1}, \ldots, x_{\beta,1}$. □

According to Propositions 2 and 3, for any recursively enumerable set $N \subseteq Ps(\alpha)RE$ of vectors of non-negative integers there exists a deterministic (or non-deterministic) register machine M with $(\alpha+2)$ registers accepting (generating, respectively) N. For this register machine M, following the proof in Theorem 6, we can construct a deterministic (or non-deterministic) numerical P system with a lower-threshold that accepts (or generates, respectively) N.

Corollary 7. $Ps(\alpha)RE = N_{gen}T_lP_1(poly^1(5), one) = N_{acc}T_lP_1^D(poly^1(5), one)$.

In conclusion, we obtain the following characterizations of *NRE*.

Theorem 8. $NRE = N_{acc}T_lP_1^D(poly^1(3), all) = N_{gen}T_lP_1(poly^1(5), one) =$
$= N_{acc}T_lP_1^D(poly^1(5), one)$.

Proof: The first equation can be obtained according to Corollary 5, where $\alpha = 1$. Similarly, letting $\alpha = 1$ in Corollary 7, we can obtain the last two equations. □

5 The Universality of Numerical P Systems with Upper-thresholds

In this section we prove that the computational power of numerical P systems with upper-thresholds (for short, UTNP systems) is equivalent with that of numerical P systems with lower-thresholds (for short, LTNP systems).

Lemma 9. *For any numerical P system with a lower-threshold Π_l, there is a P system Π_u with an upper-threshold, with the same variables, such that the corresponding variables of Π_l and Π_u have equal values but of opposite sign.*

Proof: Let Π_l be a numerical P system with a lower-threshold of the form considered in the previous sections. We construct a numerical P system with an upper-threshold Π_u in the following way. Π_u has the same membrane structure as Π_l. In the same membrane, the two systems have the same variables. The initial values of variables in Π_u is the same as in Π_l multiplied with -1. Similarly, for the thresholds of the two systems (they are equal, but of opposite signs). For a program

$$f_l(x_1, \ldots, x_i)|_T \to c_1|v_1 + \cdots + c_l|v_l$$

in Π_l, we introduce in Π_u the program

$$f_u(x_1, \ldots, x_i)|_{-T} \to c_1|v_1 + \cdots + c_l|v_l,$$

where $f_u(x_1, \ldots, x_i)$ is constructed as follows:

- If the production function $f_l(x_1, x_2, \ldots, x_n)$ is an odd function, that is,

  $$f_l(-x_1, -x_2, \ldots, -x_n) = -f_l(x_1, x_2, \ldots, x_n),$$

  then $f_u(x_1, x_2, \ldots, x_n) = f_l(x_1, x_2, \ldots, x_n)$;

- If the production function $f_l(x_1, x_2, \ldots, x_n)$ is an even function, that is,

  $$f_l(-x_1, -x_2, \ldots, -x_n) = f_l(x_1, x_2, \ldots, x_n),$$

  then $f_u(x_1, x_2, \ldots, x_n) = -f_l(x_1, x_2, \ldots, x_n)$;

- If the production function $f_l(x_1, x_2, \ldots, x_n)$ is neither an even function nor an odd function, then it can be expressed as the addition of an even function with an odd function, that is,

$$f_l(x_1, x_2, \ldots, x_n) = \frac{f_l(x_1, x_2, \ldots, x_n) + f_l(-x_1, -x_2, \ldots, -x_n)}{2} + \frac{f_l(x_1, x_2, \ldots, x_n) - f_l(-x_1, -x_2, \ldots, -x_n)}{2},$$

and then

$$\begin{aligned} f_u(x_1, x_2, \ldots, x_n) = &- \frac{f_l(x_1, x_2, \ldots, x_n) + f_l(-x_1, -x_2, \ldots, -x_n)}{2} \\ &+ \frac{f_l(x_1, x_2, \ldots, x_n) - f_l(-x_1, -x_2, \ldots, -x_n)}{2} \\ &= -f_l(-x_1, -x_2, \ldots, -x_n). \end{aligned}$$

Based on the previous construction of the UTNP system Π_u, we can get that, if the two systems are deterministic, working in the all-parallel mode, then at any step, the program in Π_l and its corresponding program in Π_u can simultaneously be applied or cannot be applied, and the two production functions have equal values but of opposite signs. Thus at any step the variable in the two systems get equal contributions but of opposite signs; this is true also for the initial values, hence always the values of variables are equal but of opposite signs. The systems halt simultaneously.

In conclusion, no matter whether Π_l, Π_u work in computing mode or in generating mode, this lemma holds true. □

Corollary 10. $Ps(\alpha)RE = N_{acc}T_uP_1^D(poly^1(3), all)$.

Proof: According to Proposition 2, for any recursively enumerable set $N \subseteq Ps(\alpha)RE$ of vectors of non-negative integers there exists a deterministic register machine M with $(\alpha + 2)$ registers accepting N. For this register machine M, following the proof in Theorem 4, we can construct a numerical P system with a lower-threshold Π_M that accepts N.

For Π_M, according to Lemma 9, we can construct an UTNP system Π_u with "contrary" configurations (equal values of variables, but of opposite signs). Now we add the programs

$$1 + p_m - x_{i,1}|_{-1} \to 1|x_{i,1},\ 1 \leq i \leq \beta. \tag{13}$$

to Π_u thus obtaining a new UTNP system Π'_u. The initial value of p_m is 0, hence programs (13) cannot be applied. As long as $p_m = 0$, there is no difference between the functioning of Π'_u and Π_u. When p_m is equal to -1, Π_u reaches a halt configuration, while Π'_u continues executing program (13). The effect of programs (13) is transforming the variables $x_{i,1} \leq -1$ to their contrary. Thus the system Π'_u has the same output as Π_M. □

In a similar way to the proof of Corollary 7, we can prove the following corollary.

Corollary 11. $Ps(\alpha)RE = N_{gen}T_uP_1(poly^1(5), one) = N_{acc}T_uP_1^D(poly^1(5), one)$.

If we set $\alpha = 1$ in Corollary 10 and Corollary 11, then we can get the following characterizations.

Theorem 12. $NRE = N_{acc}T_uP_1^D(poly^1(3), all) = N_{gen}T_uP_1(poly^1(5), one) =$
$= N_{acc}T_uP_1^D(poly^1(5), one)$.

6 Conclusions and Discussions

In this work, we have introduced thresholds into numerical P systems, and the computational power of such P systems has been investigated. Specifically, we proved that universality can be obtained for such P systems with one membrane and linear production functions working both in the all-parallel mode and in the one-parallel mode.

The rules of numerical P systems with thresholds constructed in Section 4 are applied in the all-parallel mode and in the one-parallel mode, respectively. It remains open what the computational power of numerical P systems with thresholds working in the sequential mode is.

In this work, the polynomial functions used in numerical P systems with the two kinds of thresholds have at most 3 variables for all-parallel systems and 5 variables for one-parallel systems. It is a natural question whether the number of variables can be decremented.

The thresholds are used in the sense of lower-bounds and upper-bounds. Other ways to use the thresholds could be of interest, for example, applying a program only if the values of all (or part of) the variables are strictly greater (or smaller) than the threshold.

Numerical P systems and enzymatic numerical P systems have already been used in robot control [4, 10, 16, 17]. It remains to check whether numerical P systems with thresholds are also useful for such applications.

Acknowledgements

This work was supported by National Natural Science Foundation of China (61320106005 and 61472154), Ph.D. Programs Foundation of Ministry of Education of China (2012014213008), Natural Science Foundation of Hubei Province (2011CDA027), and the Innovation Scientists and Technicians Troop Construction Projects of Henan Province (154200510012).

Bibliography

[1] Freund, R.; Oswald, M. (2002); GP Systems with Forbidding Context. *Fundamenta Informaticae* 49(1-3), 81–102.

[2] Freund, R.; Păun, G. (2001); On the Number of Non-Terminal Symbols in Graph-Controlled, Programmed and Matrix Grammars. In: *Machines, Computations, and Universality*, 3rd Internat. Conf., MCU, Lecture Notes in Computer Science, vol. 2055, Springer, Berlin, 214–225.

[3] Ionescu, M.; Păun, G., Yokomori; T. (2006); Spiking Neural P Systems. *Fundamenta Informaticae* 71(2-3), 279–308.

[4] Leporati, A.; Porreca, A.E.; Zandron, C.; Mauri, G. (2013); Improving Universality Results on Parallel Enzymatic Numerical P Systems. *Proc. 11th Brainstorming Week on Membrane Computing*, Sevilla, 4–8.

[5] Martín-Vide, C.; Pazos, J.; Păun, Gh.; Rodriguez-Paton, A. (2003); Tissue P Systems. *Theoretical Computer Science* 296(2), 295–326.

[6] Minsky, M.L. (1967); *Computation: Finite and Infinite Machines*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

[7] Păun, G. (2000); Computing with Membranes. *Journal of Computer and System Sciences* 61(1), 108–143.

[8] Păun, G. (2002); *Membrane Computing–An Introduction*. Springer-Verlag, Berlin.

[9] Păun, G. (2013); Some Open Problems about Numerical P Systems. *Proc. 11th Brainstorming Week on Membrane Computing*, Sevilla, 245–252.

[10] Păun, G.; Păun, R. (2006); Membrane Computing and Economics: Numerical P Systems. *Fundamenta Informaticae*, 73(1), 213–227.

[11] Păun, G.;Rozenberg, G.; Salomaa A.(eds.)(2010); *The Oxford Handbook of Membrane Computing*. Oxford University Press, New York.

[12] Pavel, A.B.; Arsene, O.; Buiu, C. (2010); Enzymatic Numerical P Systems–A New Class of Membrane Computing Systems. In: *IEEE Fifth International Conference on Bio- Inspired Computing: Theories and Applications (BIC-TA)*, 1331–1336.

[13] Pavel, A.B.; Buiu, C. (2012); Using Enzymatic Numerical P Systems for Modeling Mobile Robot Controllers. *Natural Computing* 11(3), 387–393.

[14] Pavel, A.B.; Vasile, C.I.; Dumitrache, I. (2012); Robot Localization Implemented with Enzymatic Numerical P Systems. In: *Biomimetic and Biohybrid Systems*, Springer, 204–215.

[15] Pavel, A.B.; Vasile, C.I.; Dumitrache, I. (2013); Membrane Computing in Robotics. In: *Beyond Artificial Intelligence*, Springer, Berlin, 125–135.

[16] Vasile, C.I.; Pavel, A.B.; Dumitrache, I. (2013); Universality of Enzymatic Numerical P Systems. *International Journal of Computer Mathematics* 90(4), 869–879.

[17] Vasile, C.I.; Pavel, A.B.; Dumitrache, I.; Păun, G. (2012); On the Power of Enzymatic Numerical P Systems. *Acta Informatica* 49(6), 395–412.

[18] Wang, J.; Hoogeboom, H.J.; Pan, L.; Păun, G.; Pérez-Jiménez, M.J. (2010); Spiking Neural P Systems with Weights. *Neural Computation* 22(10), 2615–2646.

Author index