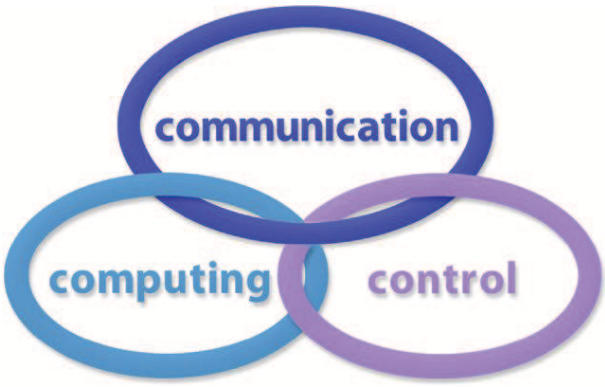
INTERNATIONAL JOURNAL

of

COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836



A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2018 Volume: 13 Issue: 5 Month: October

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

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.
ISO: Int. J. Comput. Commun. Control; **JCR Abbrev:** 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 Misu-Jan MANOLESCU.

Indexing/Coverage:

- Since 2006, Vol. 1 (S), IJCCC is covered by Clarivate Analytics and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.

  2018 Journal Citation Reports® Science Edition (Clarivate Analytics, 2017):
  Subject Category: (1) Automation & Control Systems: Q4(2009, 2011, 2012, 2013, 2014, 2015), Q3(2010, 2016, 2017); (2) Computer Science, Information Systems: Q4(2009, 2010, 2011, 2012, 2015), Q3(2013, 2014, 2016, 2017).

  Impact Factor/3 years in JCR: 0.373(2009), 0.650 (2010), 0.438(2011); 0.441(2012), 0.694(2013), 0.746(2014), 0.627(2015), 1.374(2016) **1.29 (2017)**.
  Impact Factor/5 years in JCR: 0.436(2012), 0.622(2013), 0.739(2014), 0.635(2015), 1.193(2016), **1.179(2017)**.

- Since 2008 IJCCC is indexed by Scopus: **CiteScore 2017 = 1.04**.
  Subject Category:
  (1) Computational Theory and Mathematics: Q4(2009, 2010, 2012, 2015), Q3(2011, 2013, 2014, 2016, 2017);
  (2) Computer Networks and Communications: Q4(2009), Q3(2010, 2012, 2013, 2015), Q2(2011, 2014, 2016, 2017);
  (3) Computer Science Applications: Q4(2009), Q3(2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017).

  SJR: 0.178(2009), 0.339(2010), 0.369(2011), 0.292(2012), 0.378(2013), 0.420(2014), 0.263(2015), 0.319(2016), 0.326 (2017).

- Since 2007, 2(1), IJCCC is indexed in EBSCO.

Focus & **Scope:** International Journal of Computers Communications & Control is directed to the international communities of scientific researchers in computers, communications 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, Communications, Control).

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

(1) Integrated solutions in computer-based control and communications;

(2) Computational intelligence methods & Soft computing (with particular emphasis on fuzzy logic-based methods, computing with words, ANN, evolutionary computing, collective/swarm intelligence, membrane computing, quantum computing);

(3) Advanced decision support systems (with particular emphasis on the usage of combined solvers and/or web technologies).

FAMOUS FORMER MEMBER IN THE EDITORIAL BOARD OF IJCCC

Lotfi A. Zadeh (Feb. 4, 1921 - Sept. 6, 2017)
The inventor of Fuzzy Sets, Fuzzy Logic, and Soft Computing
Former member in the Editorial Board of IJCCC between 2008-2017

FAMOUS EXCELLENCE AWARD PROPOSED FOR IJCCC

Journals Excellence Awards proposed by Scopus (2015)

EDITORIAL STAFF OF IJCCC (2018)

EDITORIAL ADDRESS:
Agora University, Cercetare Dezvoltare Agora, Tineretului 8, 410526 Oradea, Bihor, Romania,
Tel./ Fax: +40 359101032, E-mail: ijccc@univagora.ro, rd.agora@univagora.ro
URL: http://univagora.ro/jour/index.php/ijccc/

EDITORIAL BOARD OF IJCCC (MEMBERS, 2018):

Constantin GAINDRIC
IMMAS, Republic of MOLDOVA
Kishinev, 277028, Academiei 5
gaindric@math.md

Xiao-Shan GAO
Academia Sinica, CHINA
Beijing 100080, China
xgao@mmrc.iss.ac.cn

Enrique HERRERA-VIEDMA
University of Granada, SPAIN
Av. del Hospicio, s/n, 18010 Granada
viedma@decsai.ugr.es

Kaoru HIROTA
Tokyo Institute of Tech., JAPAN
G3-49,4259 Nagatsuta
hirota@hrt.dis.titech.ac.jp

Arturas KAKLAUSKAS
VGTU, LITHUANIA
Sauletekio al. 11, LT-10223 Vilnius
arturas.kaklauskas@vgtu.lt

Gang KOU
SWUFE, CHINA
Chengdu, 611130
kougang@swufe.edu.cn

Heeseok LEE
KAIST, SOUTH KOREA
85 Hoegiro, Seoul 02455
hsl@business.kaist.ac.kr

George METAKIDES
University of Patras, GREECE
Patra 265 04, Greece
george@metakides.net

Shimon Y. NOF
Purdue University, USA
610 Purdue Mall, West Lafayette
nof@purdue.edu

Stephan OLARIU
Old Dominion University, USA
Norfolk, VA 23529-0162
olariu@cs.odu.edu

Gheorghe PĂUN
Romanian Academy, ROMANIA
IMAR, Bucharest, PO Box 1-764
gpaun@us.es

Mario de J. PEREZ JIMENEZ
University of Seville, SPAIN
Avda. Reina Mercedes s/n, 41012
marper@us.es

Radu-Emil PRECUP
Pol. Univ. of Timisoara, ROMANIA
Bd. V. Parvan 2, 300223
radu.precup@aut.upt.ro

Radu POPESCU-ZELETIN
Technical University Berlin, GERMANY
Fraunhofer Institute for Open CS
rpz@cs.tu-berlin.de

Imre J. RUDAS
Obuda University, HUNGARY
Budapest, Becsi ut 96b, 1034
rudas@bmf.hu

Yong SHI
Chinese Academy of Sciences, CHINA
Beijing 100190
yshi@gucas.ac.cn, yshi@unomaha.edu

Bogdana STANOJEVIC
Serbian Academy of SA, SERBIA
Kneza Mihaila 36, Beograd 11001
bgdnpop@mi.sanu.ac.rs

Athanasios D. STYLIADIS
University of Kavala, GREECE
65404 Kavala
styliadis@teikav.edu.gr

Gheorghe TECUCI
George Mason University, USA
University Drive 4440, Fairfax VA
tecuci@gmu.edu

Horia-Nicolai TEODORESCU
Romanian Academy, ROMANIA
Iasi Branch, Bd. Carol I 11, 700506
hteodor@etc.tuiasi.ro

Dan TUFIŞ
Romanian Academy, ROMANIA
13 Septembrie, 13, 050711 Bucharest
tufis@racai.ro

Edmundas K. ZAVADSKAS
VGTU, LITHUANIA
Sauletekio ave. 11, LT-10223 Vilnius
edmundas.zavadskas@vgtu.lt

Prof. Misu-Jan Manolescu at 60 Years

Prof. Misu-Jan Manolescu, Ph.D.,
Founder of Agora University of Oradea, Romania (2000);
Rector (Provost) of Agora University of Oradea (2000-2012);
Co-Founder and Chair of Organizing Committee of the
International Conference of Computers Communications and Control (2006);
Co-Founder and Managing Editor of the
International Journal of Computers Communications & Control (2006).

Misu-Jan Manolescu was born on September 30, 1958, Craiova, Dolj, Romania.

He graduated from University of Craiova, Romania, where he received his B.Sci. (eq. M.Sci.) in Engineering at Faculty of Electrical Sciences (1984).

In 1994 he received his Ph.D. in Electro-energetic Science (Microwave field) at University of Oradea, Romania. In 2001 he received his Ph.D. in Economic Sciences (Human Resources Management field) at University of Craiova.

He is a Professor at Agora University of Oradea. His current research interests include different aspects of risk management and soft computing applications in technology and economy.

Main scientific and professional contributions of Prof. M.-J. Manolescu are: (co)-author of 9 books; co-editor of 2 proceedings indexed in ISI Web of Science; 65 papers in journals and proceedings; 10 papers indexed in ISI Web of Science; 2 innovations - co-author; 2 inventions - co-author; Founder of the Agora Foundation; Founder of the Agora International School of Oradea; Founder of the Agora University (2000, founder rector); Founder and managing editor of the "International Journal of Computers, Communication and Control"; Founder of the "Agora International Journal of Economical Sciences"; Founder of the "Agora International Journal of Juridical Sciences"; Member of the Editorial Board of "Nike" Scientific Journal, Italy; Member of the Administration Council of the "S. Pio V" Institute of Political Sciences from Rome, Italy, during 2001-2012; Member of the Administration Council of "ODIMED" (Observatory for Human Rights in the Mediterranean Area) from Rome, Italy.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, Volume 13, Issue 5, October, 2018.

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 759-771, October 2018.

A Spectral Clustering Algorithm Improved by P Systems

G. Chen, J. Hu, H. Peng, J. Wang, X. Huang

Guangchun Chen, Juan Hu, Hong Peng*,
Xiangnian Huang
School of Computer and Software Engineering
Xihua University, Chengdu, 610039, Sichuan, China
gchen@mail.xhu.edu.cn, jhu@mail.xhu.edu.cn
*Corresponding author: ph.xhu@hotmail.com
xhuang@mail.xhu.edu.cn

Jun Wang
School of Electrical and Information Engineering
Xihua University, Chengdu, 610039, Sichuan, China
wj.xhu@hotmail.com

Abstract: Using spectral clustering algorithm is difficult to find the clusters in the cases that dataset has a large difference in density and its clustering effect depends on the selection of initial centers. To overcome the shortcomings, we propose a novel spectral clustering algorithm based on membrane computing framework, called MSC algorithm, whose idea is to use membrane clustering algorithm to realize the clustering component in spectral clustering. A tissue-like P system is used as its computing framework, where each object in cells denotes a set of cluster centers and velocity-location model is used as the evolution rules. Under the control of evolution-communication mechanism, the tissue-like P system can obtain a good clustering partition for each dataset. The proposed spectral clustering algorithm is evaluated on three artificial datasets and ten UCI datasets, and it is further compared with classical spectral clustering algorithms. The comparison results demonstrate the advantage of the proposed spectral clustering algorithm.
Keywords: machine learning, spectral clustering, membrane computing, tissue-like P systems.

1 Introduction

Membrane computing initiated by Gheorghe Păun [17], was inspired from the structure and functioning of living cell as well as from the cooperation of cells in tissues, organs, and biological neural networks. Membrane computing is a class of distributed parallel computing models, known as P systems or membrane systems. In the past years, many variants of P systems have been proposed [7, 8, 11, 15, 16, 18, 19, 32, 41, 46], and they have been applied to different real-world problems [47], for example, combinatorial optimization [42, 44, 48], robots [1], image processing [4, 5, 22, 25, 40, 43], signal processing [23, 36, 45], knowledge representation [26, 34, 37], fault diagnosis [21, 28, 33, 35, 38, 39], ecology and system biology [3, 9, 10]. Most of membrane systems have been proved to be powerful (equivalent with Turing machine) and effective (able to solve the NP hard problems in a feasible time).

In the recent years, P systems were used to deal with data clustering problems. Zhao et al [49] presented an improved clustering algorithm, in which the rules in cell-like P systems were used to realize classical k-medoids algorithm. In Peng [24], evolution-communication P systems are used to deal with fuzzy clustering problems. In [27] and [29], two different mechanisms of P systems were considered to investigate automatic clustering problems. Liu et al [12] used a cell-like P systems with promoters and inhibitors to develop a k-medoids clustering algorithm.

In [20], fuzzy clustering problems were viewed as a multiobjective optimization problem and a tissue-like P system was designed to solve the optimization problem.

Spectral clustering is a popular method for solving clustering problems in a wide range of non-Euclidian spaces, linearly non-separable clusters and detecting non-convex patterns [13]. The key idea in spectral clustering is to achieve graph partitioning by performing eigen-decomposition of a graph Laplacian matrix. The obtained eigenvectors are used as the low dimensional representation of the data, and then the k-means algorithm is applied to generate the clusters. Spectral clustering approaches differ in how they define and construct the Laplacian matrix and thus which eigenvectors are selected to represent the partitioning. Moreover, different objective functions are used to derive the best cut. Chan et al. [2] proposed the ratio cut to minimize the total cost of the edges crossing the cluster boundaries, normalized by the size of the k clusters, to encourage balanced cluster sizes. Shi and Malik [31] established the normalized cut (NCut), which can measure the dissimilarities among groups and within clusters. In [6], Ding et al. proposed min-max cut criterion, which can avoid to segment the smaller subgraphs that contains only a few vertices. According to different partitioning criteria and spectral mapping methods, many different methods have been developed to realize spectral clustering algorithms. Perona and Freeman [30] proposed PF algorithm based on iterative spectrum, which is the simplest spectral clustering algorithm. Ng et al. [14] proposed the NJW algorithm, which is based on the K channel segmentation. However, there are a number of shortcomings in spectral clustering algorithms, for example, it is difficult to find the clusters with a large difference in density and their clustering effect depends on the selection of initial centers.

This paper focuses on application of membrane computing model in spectral clustering to overcome the shortcomings and presents a novel spectral clustering algorithm based on a membrane computing model, called MSC algorithm. A tissue-like P system is considered as a computing framework, and a membrane clustering algorithm is developed based on the computing framework and is embedded in a classical spectral clustering algorithm. To the best of our knowledge, this is the first attempt to use membrane computing model for improving spectral clustering algorithm.

The remainder of this paper is organized as follows. Section 2 reviews classical spectral clustering algorithm. Section 3 describes in detail the proposed membrane spectral clustering (MSC) algorithm. Experimental results and analysis are provided in Section 4. Conclusions is given in Section 5.

2 Spectral clustering and the NJW method

Spectral clustering method is a widely used graph-based approach for data clustering. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ in $R^{n\times d}$ with k clusters. We expect the dataset X will be transformed into a weighted undirected graph $G = (V, E)$, in which $V = \{x_i\}_{i=1}^n$ is the vertex set composed of n data points, and $E = \{w_{ij}\}_{i,j=1}^n$ is the set of weighted edges, where w_{ij} indicates the pairwise similarity between the x_i and x_j. V and E contain all vertices and edges, respectively, Let $W = (w_{i,j})_{1\le i,j\le n}$ be the affinity matrix. Usually, w_{ij} in affinity matrix can be measured by a Gaussian function:

$$w_{ij} = \begin{cases} e^{-\frac{d(x_i,x_j)^2}{\sigma^2}}, & i \neq j \\ 0, & i = j \end{cases} \tag{1}$$

The degree matrix D is a diagonal matrix, whose element D_{ii} is the degree of data point x_i, i.e., $D_{ii} = \sum_{j=1}^n w_{ij}$. Based on the two matrices, we can obtain the Laplacian matrix, L. There are three forms of Laplacian matrices: (i) unnormalized Laplacian matrix ($L = D - W$), and two

normalized Laplacian matrices, (ii) symmetric Laplacian matrix ($L_{sym} = D^{-1/2}WD^{-1/2}$) and (iii) random-walk Laplacian matrix ($L_{rw} = I - D^{-1}W$).

As a spectral approach for graph partitioning problem, NJW method is one of the most widely used spectral clustering algorithms. Its idea is to find a new representation of patterns on the first k eigenvectors of the Laplacian matrix. Algorithm 1 gives the details of NJW method.

Algorithm 1 NJW method

Input: $X \in R^{n\times d},\ k \in N$
Output: $V = \{v_i | i = 1, 2, \ldots, k\}$

1: Construct the affinity matrix $W \in R^{n\times n}$ according to Eq. (16);
2: Compute the degree matrix D;
3: Compute the normalized Laplacian matrix $L_{sym} = D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$;
4: Let $0 \le \lambda_1 \le \lambda_2 \le \cdots \le \lambda_k$ be the k least eigenvalues of L_{sym} and $p_1, p_2, \ldots, p_k$ be the corresponding eigenvectors; Construct the matrix $P = [p_1, p_2, \ldots, p_k] \in R^{n\times k}$, where p_i is ith column vector;
5: Construct the matrix Y from P by renormalizing each rows of P, i.e., $Y_{ij} = P_{ij}/(\sum_j P_{ij}^2)^{1/2}$;
6: Treat each row of Y as a point in R^k, and cluster them into k clusters $c_1, c_2, \ldots, c_k$ via k-means algorithm;
7: Output the clusters that corresponds to the original data set, $v_1, v_2, \ldots, v_k$, where $v_i = \{x_j | y_j \in c_i\}$.

3 Spectral clustering algorithm based on membrane computing framework

In this paper, we will try to use membrane computing algorithm (MCA) to replace the k-means component in classical spectral clustering algorithm to realize the optimal data partitioning. The spectral clustering algorithm optimized by membrane computing model is called MSC algorithm in this paper. By contrast, classical spectral clustering algorithm is called CSC algorithm. Figure 5 shows the structural comparison of CSC and MSC algorithms. From Figure 5, we can find that first component of MSC algorithm is the same to that of CSC algorithm, but MSC algorithm uses MCA algorithm rather than k-means algorithm in component 2. Therefore, in the following, we only describe the MCA algorithm. Since the core of MCA algorithm is a tissue-like P system, we first describe the tissue-like P system, and then illustrate the proposed MCA algorithm.

3.1 A tissue-like P system

We design a tissue-like P system (consisting of q cells) as the computing framework of MCA algorithm:

$$\Pi = (O_1, O_2, \ldots, O_q, R_1, R_2, \ldots, R_q, R', i_o)$$

where O_i is the set of objects in ith cell, R_i is the set of evolution rules in ith cell, R' is the set of communication rules between the cells, and the $i_o = 0$ indicates that the environment is the output region of the system.

Figure 2 shows the tissue-like P system, which consists of q cells labeled by $1, 2, \ldots, q$ respectively. Each cell has m objects, and the environment is labeled by 0. Denote by Z_{ij} the jth

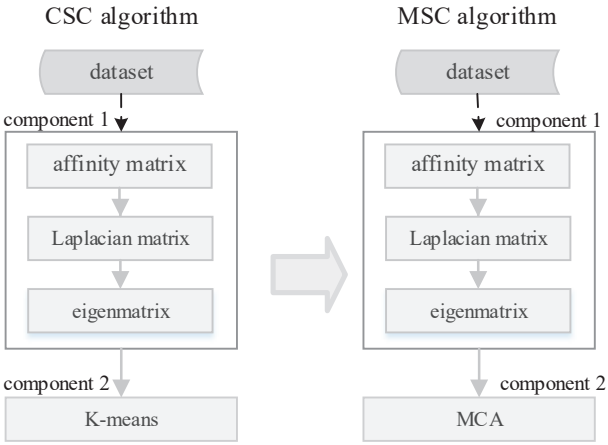


Figure 1: Structural comparison of CSC and MSC algorithms.

object in ith cell, $i = 1, 2, \ldots, q,\ j = 1, 2, \ldots, m$. The arrows in the figure indicate the communication of objects. The communication of objects is between these cells and the environment. The environment is also the output region of the system. When the system halts, the object in the environment is the optimal solution (a set of optimal cluster centers).

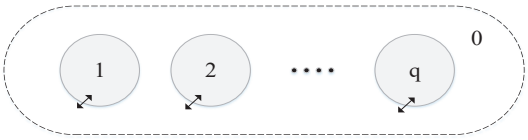


Figure 2: The designed tissue-like P system.

Each object in the cells, Z, is used to represent a set of candidate k cluster centers, $z_1, z_2, \ldots, z_k \in R^d$. Thus, object Z can be represented as a $k \times d$ dimensional vector:

$$Z = (z_{11}, z_{12}, \ldots, z_{1d}, \ldots, z_{i1}, z_{i2}, \ldots, z_{id}, \ldots, z_{k1}, z_{k2}, \ldots, z_{kd}) \tag{2}$$

where $(z_{i1}, z_{i2}, \ldots, z_{id})$ corresponds to ith candidate cluster center, $i = 1, 2, \ldots, k$.

These objects in cells will be evolved during the computation. Initially, a set of objects is generated randomly. Based on data points in a data set, we can determine a lower bound and an upper bound for each dimension, $A_j = \min\{x_{1j}, x_{2j}, \ldots, x_{nj}\}$, $B_j = \max\{x_{1j}, x_{2j}, \ldots, x_{nj}\}$, $j = 1, 2, \ldots, d$. Thus, $z_{ij} = rand([A_j, B_j])$, where $rand()$ denotes a random function that can generate the random number in $[A_j, B_j]$.

The tissue-like P system uses the communication rule of the form $< i, a; \lambda, 0 >$ to update the object in the environment, which means that object a in cell i is transported to environment 0, where λ denotes the empty object. The object in the environment is called the global optimal object, denoted by Z_{best}. For each cell, communication rule is used to communicate its best object to the environment and update the optimal object. The updating formula can be given as follows:

$$Z_{best} = \begin{cases} Z_{i,best}, & \text{if } J(Z_{i,best}) < J(Z_{best}) \\ Z_{best}, & \text{otherwise} \end{cases} \tag{3}$$

where $Z_{i,best}$ is the best object in ith cell. The object judgement is based on the following fitness function:

$$J(z_1, z_2, \ldots, z_k) = \sum_{i=1}^{k} \sum_{j=1}^{n} (u_{ij})^m \parallel x_j - z_i \parallel^2 \tag{4}$$

where u_{ij} denotes membership degree of x_j belonging to ith class, and m is a power exponent.

During the computation, tissue-like P system uses evolution rules to evolve the objects in cells. In this work, the velocity-location model of PSO is used as the evolution rules. The velocity-location model can be described as follows:

$$\begin{cases} V_j^i = wZ_j^i + c_1r_1(P_j^i - Z_j^i) + c_2r_2(Z_{best} - Z_j^i) \\ Z_j'^i = Z_j^i + V_j^i \end{cases} \tag{5}$$

where V_j^i corresponds to the speed of Z_j^i, $Z_j'^i$ is the new value of Z_j^i after evolving, and P_j^i is the best position so far for jth object in ith cell; w is the inertia weight constant, c_1 and c_2 are learning rate constants, and r_1 and r_2 are two random real numbers in $[0, 1]$. In the implementation of MCA algorithm, a linear decreasing strategy is used, i.e., $w = (0.9 - \frac{t}{2T})$, where t is the current iteration number and T is the maximum number of iterations.

In this paper, the maximum iteration number is used as halting condition. After the system halts, the best object Z_{best} in the environment is regarded as the solution. Finally, according to the optimal cluster centers, $c_1, c_2, \ldots, c_k$, N data points are classifies into k clusters.

3.2 Membrane clustering algorithm

As stated above, MCA algorithm is used as second component of the MSC algorithm. The MCA algorithm uses the designed tissue-like P system to automatically search for the optimal cluster centers for a data set to be clustered. Under the control of the evolution and communication rules, the P system continuously evolves the objects in cells and updates the global optimal object in the environment until the system halts. Figure 3 shows the flow chart of MCA algorithm.

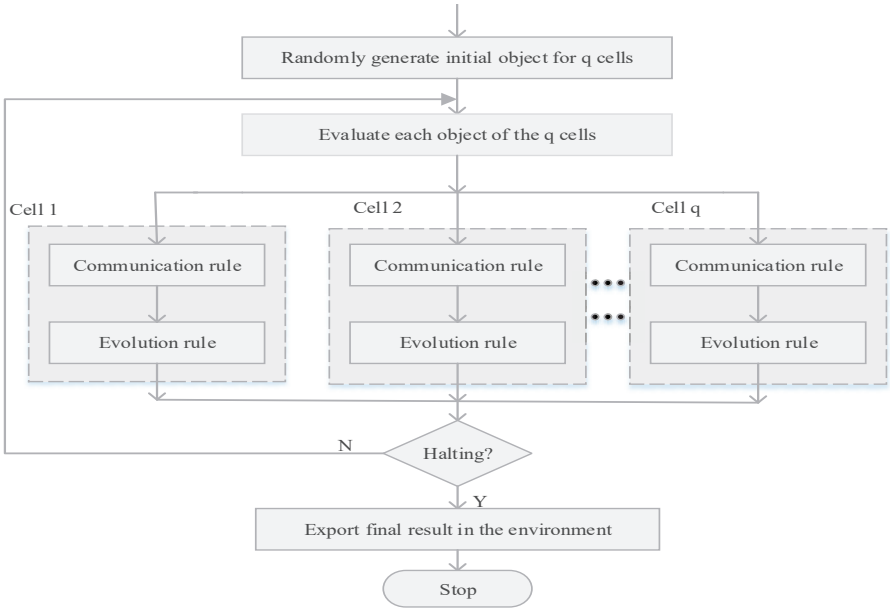
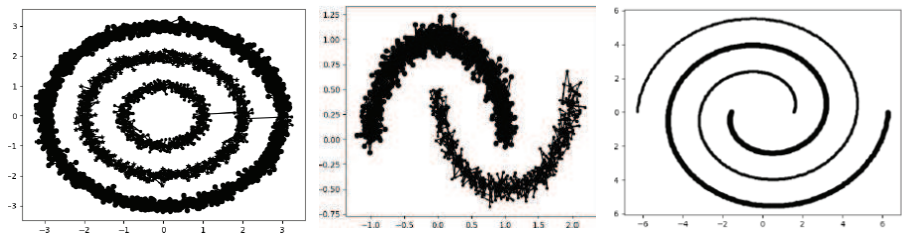


Figure 3: The flow chart of membrane clustering algorithm (MCA) used in MSC algorithm

4 Experimental results and analysis

4.1 Dataset

In order to evaluate the performance of MSC algorithm, three benchmark synthetic dataset and ten UCI dataset were used in experiments. The three synthetic dataset are *Threecircles*, *Twommoons* and *Spirl* respectively, shown in Figure 4(a)-(c). Table 1 gives the basic information for all data sets.



(a) (b) (c)

Figure 4: Three synthetic datasets. (a) Threecircle; (b) Twommoons; (c) Spirl.

Table 1: Data sets used in experiments

Datasets	No.of data	Dimension	No.of clusters
ThreeCircle	3603	2	3
Twommoons	1502	2	2
Spirl	944	2	2
Iris	150	4	3
wine	178	13	3
sonar	208	60	2
diabetes	1151	19	2
glass	214	9	6
ecoli	336	7	8
Heart	297	13	2
liver	345	6	2
ionosphere	351	34	2
sym	350	2	3

4.2 Experimental results

Three commonly used indexes of quality were used to measure the clustering performance.

(1) Adjusted Rand Index (ARI): $\rho_{ARI} \in [-1, 1]$.
This index measures the agreement between two compared partitions, namely, the ground truth (denoted as U) and the estimated by the tested clustering approach (denoted as V), and it is expressed by

$$\rho_{ARI} = \frac{a_{11} - (a_{11} + a_{01})(a_{11} + a_{10})/a_{00}}{(a_{11} + a_{01}) + (a_{11} + a_{10})/2 - (a_{11} + a_{01})(a_{11} + a_{10})/a_{00}} \tag{6}$$

where $a_{11} \in N$ is the number of sample pairs belonging to the same subset in U and in V, a_{10} is the number of sample pairs belonging to the same subset in U and to different subsets in V, $a_{01} \in N$ is the number of sample pairs belonging to different subset in U and

to the same one in V, and $a_{00} \in N$ is the number of sample pairs belonging to different subsets in U and in V.

(2) Purity Index (PUR): $\rho_{PUR} \in [0, 1]$.
This index matches the clustering partition V with the ground truth U as a weighted sum of the maximal precision values for each subset.

$$\rho_{PUR} = \frac{1}{N} \sum_{i=1}^{k} \max_{j} |v_i \cap u_j| \tag{7}$$

(3) Jaccard Index (JAC): $\rho_{JAC} \in [0, 1]$.
This index matches the similarity among two sets, U and V, as follows:

$$\rho_{JAC} = \frac{a_{11}}{a_{11} + a_{10} + a_{01}} \tag{8}$$

In the experiment, two classical spectral clustering algorithms, K-SC and ϵ-SC, were introduced to implement two MSC algorithms, where membrane clustering algorithm is used to replace k-means component in the original spectral clustering algorithms. Thus, two MSC algorithms and the corresponding classical spectral clustering algorithms were compared in experiment. Table 2 and Table 3 show the comparison results of these algorithms on synthetic and UCI datasets, respectively. For each dataset, these tables provide the experimental results of four algorithms in terms of three indexes. Note that these experimental results are average value of 10 times independently running for each algorithm on a dataset. Moreover, we also provide the averages of these algorithms for each clustering index, respectively.

From table 2, we can see that the average value of MSC algorithm is the largest and can reach 1. The results show that the spectral clustering algorithm based on membrane computing framework has an obvious advantage in improving the average performance of spectral clustering algorithm. Comparison results of MSC and TSC on the UCI datasets show that K-SC and ϵ-SC algorithms based on membrane computing framework can significantly improve the Jaccard index, indicating that the proposed MSC algorithm is more robust and has a certain ability to deal with noise data. For the ARI and the Purity indexes, MSC algorithm achieves a comparable result in comparison to the classical algorithm.

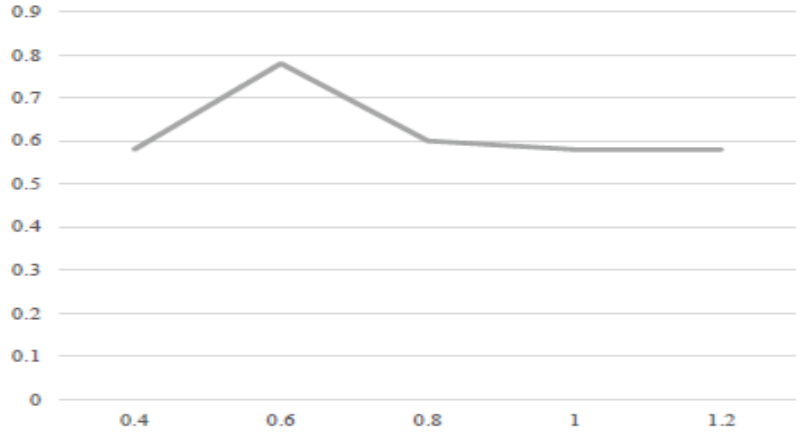
Iris dataset is used as an example to analyze the influences of parameters in MSC algorithm. Figure 5 (a)-(c) shows the influences of three parameters, including bandwidth ϵ of the Gaussian kernel function, the number of cells m and the maximum number of iterations $Maxstep$. As can be seen from the figures, MSC algorithm is more sensitive to m and ϵ, and the curve of parameter $Maxstep$ rises slowly and finally tends to straight line, which indicates that the performance of the algorithm is not improved when the maximum number of iterations is reached.

Table 2: Clustering quality assessment results (synthetic datasets).

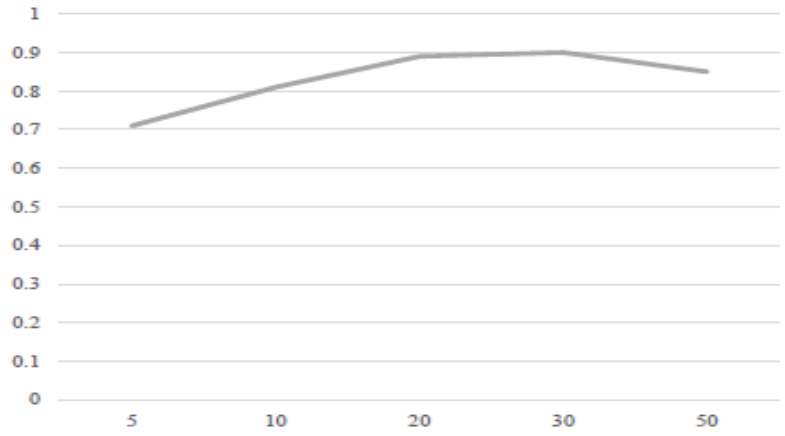
Dataset	Quality measure	K-SC		ϵ-SC	
		k-means	MCA	k-means	MCA
Threecircle	ARI	1.0	1.0	0.51	1.0
	Purity	1.0	1.0	0.84	1.0
	Jaccard	1.0	1.0	0.56	1.0
Twommoons	ARI	0.37	1.0	0.59	1.0
	Purity	0.81	1.0	0.89	1.0
	Jaccard	0.56	1.0	0.70	1.0
Spirl	ARI	1.0	1.0	0.89	1.0
	Purity	1.0	1.0	0.73	1.0
	Jaccard	1.0	1.0	0.95	1.0
Average	ARI	0.79	1.0	0.66	1.0
	Purity	0.93	1.0	0.82	1.0
	Jaccard	0.85	1.0	0.73	1.0

Table 3: Clustering quality assessment results (UCI repository datasets).

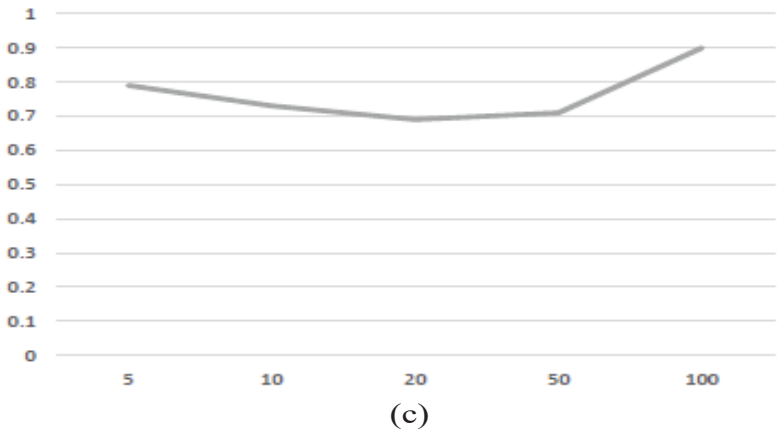
Dataset	Quality measure	K-SC		ϵ-SC	
		k-means	MCA	k-means	MCA
Iris	ARI	0.44	0.66	0.63	0.75
	Purity	0.87	0.86	0.84	0.90
	Jaccard	0.50	0.64	0.60	0.71
wine	ARI	0.03	0.30	0.29	0.0
	Purity	0.57	0.67	0.53	0.40
	Jaccard	0.24	0.37	0.38	0.34
Sonar	ARI	0.02	-0.01	0.00	-0.0058
	Purity	0.57	0.51	0.55	0.50
	Jaccard	0.34	0.48	0.34	0.48
diabetes	ARI	0.16	0.01	0.16	0.0
	Purity	0.70	0.53	0.70	0.53
	Jaccard	0.43	0.51	0.43	0.50
glass	ARI	0.17	0	0.16	0.01
	Purity	0.51	0.36	0.52	0.36
	Jaccard	0.25	0.26	0.33	0.26
ecoli	ARI	0.37	0.20	0.48	0.42
	Purity	0.56	0.45	0.66	0.70
	Jaccard	0.32	0.29	0.42	0.65
Heart	ARI	0.31	0.04	0.37	0.40
	Purity	0.78	0.55	0.80	0.75
	Jaccard	0.49	0.51	0.52	0.63
liver	ARI	0	-0.02	-0.01	0.12
	Purity	0.65	0.58	0.98	0.67
	Jaccard	0.36	0.51	0.50	0.56
ionosphere	ARI	0.15	0.01	0.13	0.25
	Purity	0.70	0.64	0.68	0.73
	Jaccard	0.44	0.54	0.41	0.55
sym	ARI	0.56	0.75	0.49	0.38
	Purity	0.68	0.87	0.75	0.54
	Jaccard	0.79	0.72	0.43	0.43
Average	ARI	0.28	0.20	0.28	0.23
	Purity	0.66	0.61	0.70	0.61
	Jaccard	0.41	0.49	0.43	0.52



(a)



(b)



(c)

Figure 5: Free parameter analysis over UCI datasets based on the Purity: (a) bandwidth ϵ of the Gaussian kernel function; (b) the number of cells m; (c) the maximum number of iterations $Maxstep$.

5 Conclusion

In this paper, we used membrane computing framework to develop a novel spectral clustering algorithm, called MSC algorithm. The core component of MSC algorithm is a tissue-like P system which is composed of several cells and uses the improved PSO algorithm as evolution mechanism. We evaluated the performance of the proposed algorithm on three artificial data sets and ten UCI datasets. The results show that compared with the classical spectral clustering algorithm, the proposed algorithm can improve the clustering performance. This study also demonstrates the effectiveness of using the membrane computing framework to solve data clustering problems.

MSC algorithm used membrane clustering algorithm (MCA) instead of k-means component in classical spectral clustering algorithm, which searches for the optimal solution by both the evolution of objects in multiple cells and the communication of objects between the cells. It is well known that membrane computing is a distributed computing model. However, MSC algorithm is not implemented in parallel due to limitation of the computer's serial architecture. Therefore, our further work is to discuss the parallel implementation of MSC algorithm on GPGPU and/or FPGA.

Funding

This work was partially supported by the National Natural Science Foundation of China (No. 61472328), Chunhui Project Foundation of the Education Department of China (Nos. Z2016143 and Z2016148), Research Foundation of the Education Department of Sichuan province (No. 17TD0034), China, and the Innovation Fund of Postgraduate, Xihua University(no. ycjj2017073).

Bibliography

[1] Buiu, C.; Vasile, C.; Arsene, O. (2012); Development of membrane controllers for mobile robots, *Information Sciences*, 187, 33-51, 2012.

[2] Chan, P.K.; Schlag, M.D.F.; Zien, J.Y. (1993); Spectral k-way ratio-cut partitioning and clustering, *DAC*, 749-754, 1993.

[3] Colomer, A.M.; Margalida, A.; Pérez-Jiménez, M.J. (2013); Population dynamics P system (PDP) models: a standarized protocol for describing and applying novel bio-inspired computing tools, *Plos One*, 4, 1-13, 2013.

[4] Díaz-Pernil, D.; Berciano, A.; Peña-Cantillana, F.; Gutiérrez-Naranjo, M.A. (2013); Segmenting images with gradient-based edge detection using membrane computing, *Pattern Recognition Letters*, 34(8), 846-855, 2013.

[5] Díaz-Pernil, D.; Peña-Cantillana, F.; Gutiérrez-Naranjo, M.A. (2013); A parallel algorithm for skeletonizing images by using spiking neural P systems, *Neurocomputing*, 115, 81-91, 2013.

[6] Ding, C.; He, X.; Zha, H.; Gu, M.; Simon, H. (2001); Spectral min-max cut for graph partitioning and data clustering, *Technical Report TR-2001-XX*, Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, 2001.

[7] Dzitac, I. (2015); Impact of membrane computing and P systems in ISI WoS. celebrating the 65th birthday of Gheorghe Păun, *International Journal of Computers Communications & Control*, 10(5), 617-626, 2015.

[8] Freund, R.; Păun, G.; Pérez-Jiménez, M.J. (2005); Tissue-like P systems with channel-states, *Theoretical Computer Science*, 330, 101-116, 2005.

[9] Garcia-Quismondo, M.; Levin, M.; Lobo-Fernández, D. (2017); Modeling regenerative processes with Membrane Computing, *Information Sciences*, 381, 229-249, 2017.

[10] Gheorghe, M.; Manca, V.; Romero-Campero, F.J. (2010); Deterministic and stochastic P systems for modelling cellular processes, *Natural Computing*, 9(2), 457-473, 2010.

[11] Ionescu, M.; Păun G.; Yokomori, T. (2006); Spiking neural P systems, *Fundamenta Informaticae*, 71, 279-308, 2006.

[12] Liu, X.; Zhao, Y.; Sun, W. (2016); K-medoids-based consensus clustering based on cell-like P systems with promoters and inhibitors, *Bio-inspired Computing - Theories and Applications*, 95-108, 2016.

[13] Luxburg, U.V. (2007); A tutorial on spectral clustering, *Statistics and Computing*, 17(4), 395-416, 2007.

[14] Ng, A.Y., Jordan, M., Weiss, Y. (2001); On spectral clustering: analysis and an algorithm, *Proc Nips*, 849-856, 2001.

[15] Pan, L.; Wang, J.; Hoogeboom, H.J. (2012); Spiking neural P systems with astrocytes, *Neural Computation*, 24(3), 805-825, 2012.

[16] Pan, L.; Păun, G. (2009); Spiking neural p systems with anti-spikes, *International Journal of Computers Communications & Control*, 4(3), 273-282, 2009.

[17] Păun, G. (2000); Computing with membranes, *Journal of Computer System Sciences*, 61(1), 108-143, 2000.

[18] Păun, G.; Rozenberg, G.; Salomaa, A. (2010); *The Oxford Handbook of Membrance Computing*, Oxford Unversity Press, New York, 2010.

[19] Păun, G. (2016); Membrane computing and economics: a general view, *International Journal of Computers Communications & Control*, 11(1), 105-112, 2016.

[20] Peng, H.; Shi, P.; Wang, J.; Riscos-Núñez, A.; Pérez-Jiménez, M.J. (2017); Multiobjective fuzzy clustering approach based on tissue-like membrane systems, *Knowledge-Based Systems*, 125, 74-82, 2017.

[21] Peng, H.; Wang, J.; Ming, J.; Shi, P.; Pérez-Jiménez, M.J.; Yu, W.; Tao, C. (2018); Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems, *IEEE Transaction on Smart Grid*, 2018. Available at http://dx.doi.org/10.1109/TSG.2017.2670602.

[22] Peng, H.; Wang, J.; Pérez-Jiménez, M.J. (2015); Optimal multi-level thresholding with membrane computing, *Digital Signal Processing*, 37, 53-64, 2015.

[23] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2014); The framework of P systems applied to solve optimal watermarking problem, *Signal Processing*, 101, 256-265, 2014.

[24] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2015); An unsupervised learning algorithm for membrane computing, *Information Sciences*, 304(20), 80-91, 2015.

[25] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Shi, P. (2013); A novel image thresholding method based on membrane computing and fuzzy entropy, *Journal of Intelligent and Fuzzy Systems*, 24(2), 229-237, 2013.

[26] Peng, H.; Wang, J.; Pérez-Jiménez, M.J.; Wang, H.; Shao, J.; Wang, T. (2013); Fuzzy reasoning spiking neural P system for fault diagnosis, *Information Sciences*, 235(20), 106-116, 2013.

[27] Peng, H.; Wang, J.; Shi, P.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2016); An extended membrane system with active membrane to solve automatic fuzzy clustering problems, *International Journal of Neural Systems*, 26, 1-17, 2016.

[28] Peng, H.; Wang, J.; Shi, P.; Pérez-Jiménez, M.J.; Riscos-Núñez, A. (2017); Fault diagnosis of power systems using fuzzy tissue-like P systems, *Integrated Computer-Aided Engineering*, 24, 401-411, 2017.

[29] Peng, H.; Wang, J.; Shi, P.; Riscos-Núñez, A.; Pérez-Jiménez, M.J. (2015); An automatic clustering algorithm inspired by membrane computing, *Pattern Recognition Letters*, 68(15), 34-40, 2015.

[30] Perona, P.; Freeman, W. (1998); A factorization approach to grouping, *Computer Vision ECCV'98*, Springer, 655-670, 1998.

[31] Shi, J.; Malik, J. (2000); Normalized cuts and image segmentation, IEEE Transactions on pattern analysis and machine intelligence, 22(8), 888-905, 2000.

[32] Song, T.; Pan, L., Păun, G. (2014), Spiking neural P systems with rules on synapses, *Theoretical Computer Science*, 529, 82-95, 2014.

[33] Tu, M.; Wang, J.; Peng, H.; Shi, P. (2014); Application of adaptive fuzzy spiking neural P systems in fault diagnosis of power systems, *Chin. Jour. Elect.*, 23(1), 87-92, 2014.

[34] Wang, J.; Peng, H. (2013); Adaptive fuzzy spiking neural P systems for fuzzy inference and learning, *International Journal of Computer Mathematics*, 90(4), 857-868, 2013.

[35] Wang, J.; Peng, H.; Tu, M.; Pérez-Jiménez, M.J. (2016); A fault diagnosis method of power systems based on an improved adaptive fuzzy spiking neural P systems and PSO algorithms, *Chin. Jour. Elect.*, 25(2), 320-327, 2016.

[36] Wang, J.; Shi, P.; Peng, H. (2016); Membrane computing model for IIR filter design, *Information Sciences*, 329, 164-176, 2016.

[37] Wang, J.; Shi, P.; Peng, H.; Pérez-Jiménez, M.J.; Wang, T. (2013); Weighted fuzzy spiking neural P system, *IEEE Trans. Fuzzy Syst.*, 21(2), 209-220, 2013.

[38] Wang, T.; Zhang, G.X.; Zhao, J.B.; He, Z.Y.; Wang, J., Pérez-Jiménez, M.J. (2015); Fault diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems, *IEEE Trans. Power Syst.*, 30(3), 1182-1194, 2015.

[39] Xiong, G.; Shi, D.; Zhu, L.; Duan, X. (2013); A new approach to fault diagnosis of power systems using fuzzy reasoning spiking neural P systems, *Mathematical Problems in Engineering*, 2013(1), 211-244, 2013.

[40] Yahya, R.I.; Hasan, S.; George, L.E.; Alsalibi, B. (2015); Membrane computing for 2D image segmentation, *International Journal of Advances in Soft Computing and its Application*, 7(1), 35-50, 2015.

[41] Zeng, X.; Zhang, X.; Song, T.; Pan, L. (2014); Spiking neural P systems with thresholds, *Neural Computation*, 26(7), 1340-1361, 2014.

[42] Zhang, G.; Cheng, J.; Gheorghe, M.; Meng, Q. (2013); A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems, *Applied Soft Computing*, 13(3), 1528-1542, 2013.

[43] Zhang, G.; Gheorghe, M.; Li, Y. (2012); A membrane algorithm with quantum-inspired subalgorithms and its application to image processing, *Natural Computing*, 11(4), 701-717, 2012.

[44] Zhang, G.; Gheorghe, M.; Pan, L.; Pérez-Jiménez, M.J. (2014); Evolutionary membrane computing: a comprehensive survey and new results, *Information Sciences*, 279, 528-551, 2014.

[45] Zhang, G.; Liu, C.; Rong, H. (2010); Analyzing radar emitter signals with membrane algorithms, *Mathematical and Computer Modelling*, 52, 1997-2010, 2010.

[46] Zhang, X.; Pan, L.; Păun, A. (2015); On the universality of axon P systems, *IEEE Transactions on Neural Networks and Learning Systems*, 26(11), 2816-2829, 2015.

[47] Zhang, G.; Pérez-Jiménez, M.J.; Gheorghe, M. (2017); *Real-life Applications With Membrane Computing*, Springer, 2017.

[48] Zhang, G.; Rong, H.; Neri, F.; Pérez-Jiménez, M.J. (2014); An optimization spiking neural P system for approximately solving combinatorial optimization problems, *International Journal of Neural Systems*, 24, 1-16, 2014.

[49] Zhao, Y.; Liu, X.; Qu, J. (2012); The k-medoids clustering algorithm by a class of P system, *Journal of Information & Computational Science*, 9(18), 5777-5790, 2012.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 772-791, October 2018.

How Reliable are Compositions of Series and Parallel Networks Compared with Hammocks?

V. Drăgoi, S.R. Cowell, V. Beiu, S. Hoară, P. Gaşpar

Vlad Drăgoi*
1. Faculty of Exact Sciences
Aurel Vlaicu University of Arad
Elena Dragoi St., 310330, Arad, Romania
2. Laboratoire LITIS - EA 4108
Université de ROUEN - UFR Sciences et Techniques
Avenue de l'université, 76800 Saint Etienne du Rouvray, France
*Corresponding author: vlad.dragoi@uav.ro

Simon R. Cowell, Valeriu Beiu, Sorin Hoară, Păstorel Gaşpar
Faculty of Exact Sciences
Aurel Vlaicu University of Arad
Elena Dragoi St., 310330, Arad, Romania
{simon.cowell, valeriu.beiu, sorin.hoara, pastorel.gaspar}@uav.ro

Abstract: A classical problem in computer/network reliability is that of identifying simple, regular and repetitive building blocks (motifs) which yield reliability enhancements at the system-level. Over time, this apparently simple problem has been addressed by various increasingly complex methods. The earliest and simplest solutions are series and parallel structures. These were followed by majority voting and related schemes. For the most recent solutions, which are also the most involved (e.g., those based on Harary and circulant graphs), optimal reliability has been proven under particular conditions.

Here, we propose an alternate approach for designing reliable systems as repetitive compositions of the simplest possible structures. More precisely, our two motifs (basic building blocks) are: two devices in series, and two devices in parallel. Therefore, for a given number of devices (which is a power of two) we build all the possible compositions of series and parallel networks of two devices. For all of the resulting two-terminal networks, we compute exactly the reliability polynomials, and then compare them with those of size-equivalent hammock networks.

The results show that compositions of the two simplest motifs are not able to surpass size-equivalent hammock networks in terms of reliability. Still, the algorithm for computing the reliability polynomials of such compositions is linear (extremely efficient), as opposed to the one for the size-equivalent hammock networks, which is exponential. Interestingly, a few of the compositions come extremely close to size-equivalent hammock networks with respect to reliability, while having fewer wires.[a]

Keywords: two-terminal network, series and parallel network, composition, reliability polynomial.

[a]This paper is partially reprinted and extended, with permission based on Licence Number 4395350722700 © IEEE, from "2018 7th International Conference on Computers Communications and Control (ICCCC)."

1 Introduction

One well-known problem in information processing is that of identifying schemes (for a particular given technology) that maximize *reliability*. Reliability is an attribute of a system, a *reliable system* being one which works error-free for extended periods of time—originally an issue only

for safety-critical applications. The most common interpretation of network reliability [3], [4], [5] is connectivity-based. In [5], the author emphasizes a variety of interpretations for the reliability of networks, such as all-terminal, k-terminal or two-terminal networks:

> "*It comes as no surprise that hundreds of seemingly natural definitions arise by examining the plethora of different types of networks, causes and types of failures, and levels and types of operation. One should not expect to find a single definition for reliability that accommodates the many real situations of importance.*"
>
> (Charles J. Colbourn)

Obviously, the design-for-reliability problem becomes more challenging as the system grows larger (more complex) and is required to function without interruptions for longer times. Another aspect of interest is that enhancing/maximizing reliability should be done with a limited number of additional (redundant) components. The number of components is the simplest and most obvious *cost function*, but other cost functions (also known as *figures-of-merit*, or FoM) have been proposed and used, such as, e.g., area, power, or energy. It follows that *design-for-reliability* is a constraint optimization problem: maximize system reliability given limited resources (keeping costs as small as possible). This problem permeates way beyond computers into most man-made systems. Nature also seems to rely on reliability principles/schemes at different levels (the most well-known example here being the human brain, having 10^{11} neurons interconnected by 10^{15} axons and dendrites, working over many years).

In the following we shall first of all restrict the scope of our discussions to computers. In this context, reliability was established through five lectures given at Caltech by John von Neumann in January 1952, which were published four years later [22]. The focus was on how to design reliable circuits/computers using unreliable logic gates. The answer was to replicate gates and combine their effects by voting and/or multiplexing. Another take on this topic was advanced four years later by Edward F. Moore and Claude E. Shannon [20], [21]. The major difference was that instead of starting from gates, Moore and Shannon decided to pursue their analysis starting from relays (switching devices). Their results were much more encouraging than [22]. In particular, their device-level scheme:

- can be used with arbitrarily poor devices (i.e., absolutely random switching devices);
- requires redundancy factors which are $10^{(2...3)}$ (2 to 3 orders of magnitude) less than those needed by gate-level schemes.

Clearly, logic gates are made out of switching devices (transistors), hence device-level approaches, such as [20] should be used to enhance the reliability of the gates, before applying gate-level schemes such as those suggested in [22]. Still, this approach was not taken, as over the last few decades, the CMOS transistors have always been reliable enough. With novel nanoscale switching devices and nanoarchitectures under investigation [23], [26] the story is starting to look different [14], [13]. This prospect has triggered our interest in revisiting the work of Moore and Shannon [7]. Their scheme for improving on an unreliable switching device was to replace the device itself by a two-terminal network of identical unreliable switching devices.

That is why we further narrow down the focus of this paper to two-terminal networks. In particular, Moore and Shannon have introduced in [20] and argued in [21] for a particular type of two-terminal networks: *hammock networks*. A thorough comparison of hammock networks with other highly effective specialized networks is called for. For instance, a comparison of hammocks with circulant and Harary graphs (for which optimal reliability has been proven under particular conditions [19], [9], [25]). In any case, regular networks bode well with novel

array-based designs including vertical FET, FinFETs [12], gate-all-around FET, and arrays of beyond CMOS devices [6]. Our fresh analyses of small hammock networks [7] (as well as possible extensions [8], [2]) are exact. They have confirmed once again how challenging is to compute the associated reliability polynomials. These suggest that the design-for-reliability process using hammock networks will turn out to be quite involved.

An alternate design option (also mentioned in [20]) advocates for growing larger networks by combining two smaller networks. These can be connected in series, in parallel, or by "composing" them, i.e., replacing each and every element of a network with the other network (translates into composing their associated reliability polynomials). Compositions of hammock networks are mentioned in [20] and [21]. Still, series and parallel networks are easier to evaluate (as their reliability polynomials are simpler [17]), while compositions of series and parallel networks inherit this benefit. It means that one clear advantage of an approach that relies on *composing series and parallel networks* is a simpler design procedure. The fundamental question is how such composed networks perform versus other two-terminal networks of the same size. In particular, in this paper we compare compositions of small series and parallel networks versus hammock networks.

Related work. First of all we mention that this article is an extended version of a conference paper [10]. We bring here new results concerning combinatorial properties of compositions and a more detailed analysis of the various *FoM*s as functions of some specific design requirement, such as number-of-devices and wires. The same technique of composing networks was also used in [1]. There the authors compared hammock networks with compositions of smaller hammock networks. Their results emphasize the merit of composing smaller networks, as they come really close to hammocks while having more efficient algorithms for computing their reliability polynomials.

Our contribution. The main results that we prove in this article can be summarized as follows:

- Compositions of series and parallel networks are planar matchstick minimal networks (see Proposition 4);
- Their *width* w and *length* l, as well as *number-of-devices* n and *wires* ω are related to the Hamming weight of the corresponding binary vector (see Proposition 5 and Theorem 11);
- There is an algorithm that determines whether a matchstick minimal network is a composition of series and parallel. Our solution is a symmetric binary tree decomposition of depth $m = \log_2 n$;
- The reliability polynomial of compositions of series and parallel networks can be computed very efficiently (see Theorem 16).

The article also provides essential simulations, by detailing the reliability polynomials for all compositions of series and parallel networks with $n = 64$ as well as for the two 8-by-8 hammocks. By means of several *FoMs* we give arguments which prove that, in this particular case, hammocks are more reliable than compositions of series and parallel networks. However, the advantage of compositions is undeniable since they come close to hammocks, they have fewer wires for the same number of devices, and more significantly in our opinion, their reliability can be computed efficiently.

Organisation of the paper. The paper is structured as follows. We introduce two-terminal and hammock networks in Section 2. Compositions of series and parallel networks are discussed in Section 3. Section 4 starts by introducing reliability polynomials and several *FoMs* that we are going to use. Afterwards, we determine (exactly) the reliability polynomials for the hammock

and composition networks under investigation, and use these for comparative analyses. The paper ends with some conclusions and further directions for research.

2 Two-terminal networks

2.1 Definitions and properties

Definition 1. Let n be a strictly positive integer. We say that $\boldsymbol{N}$ is a two-terminal network of size n, or an n-network if $\boldsymbol{N}$ is a circuit, made of n identical devices, that has two distinguished contacts/terminals: an input or source S, and an output or terminus T.

With any two-terminal network we associate three parameters: *width* w, *length* l, and *size* n. The width w of $\boldsymbol{N}$ is the size of a "minimal cut" separating S from T. The length l of $\boldsymbol{N}$ is the size of a "minimal path" from S to T. The size of a two-terminal network $\boldsymbol{N}$ is related to l and w by:

$$n \geq wl \tag{1}$$

(see Theorem 3 in [20]).

When the equality in eq. (1) holds, we say that $\boldsymbol{N}$ is a minimal network. Even though there are several types of minimal networks, we will study here only matchstick minimal networks $\boldsymbol{N}$. We will denote a matchstick minimal network of width w and length l by $\boldsymbol{N}_{w,l}$. The set of all matchstick minimal networks of size $n = wl$ will be denoted $\mathcal{N}_n$, and we have:

$$\mathcal{N} = \bigcup_n \mathcal{N}_n \quad \text{and} \quad \mathcal{N}_n = \bigcup_{w|n} \mathcal{N}_{w,n/w}. \tag{2}$$

Notice that the set $\mathcal{N}_{1,1}$ has cardinality 1, since there is only one two-terminal network with $w = l = 1$, that is the single device network $\boldsymbol{N}_{1,1}$. In the sequel, we will distinguish two subsets of $\mathcal{N}_n$, namely the set of hammocks and the set of compositions of series and parallel networks.

2.2 Hammock networks

Matchstick minimal networks with the well-known "brick-wall" pattern are known as hammocks [20] (see Fig. 1). They can be generated starting from a parallel-of-series PoS network(see Fig. 1) by connecting vertically adjacent pairs of wires by short vertical "matchsticks" (red vertical lines in Fig. 1). If w and l are both even there are two solutions $\boldsymbol{H}_{w,l}$ and $\boldsymbol{H}^{+}_{w,l}$ (see Fig. 1), while otherwise we are left only with $\boldsymbol{H}_{w,l}$ (see [7] for more details).

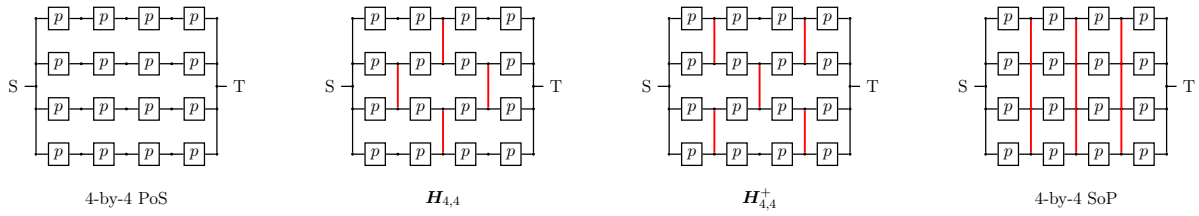


Figure 1: Square 4-by-4 parallel-of-series, hammocks and series-of-parallel.

3 Compositions of series and parallel

3.1 Definitions and properties

Definition 2. Let $\boldsymbol{C}$ represent a composition of networks. If we start from the device itself the simplest possible compositions are: two devices in series $\boldsymbol{C}^{(0)}$, and two devices in parallel $\boldsymbol{C}^{(1)}$. At the to the next level, a composition of $\boldsymbol{C}^{(0)}$ with $\boldsymbol{C}^{(1)}$ is $\boldsymbol{C} = \boldsymbol{C}^{(0)} \bullet \boldsymbol{C}^{(1)}$ which is obtained by replacing each device in $\boldsymbol{C}^{(0)}$ by $\boldsymbol{C}^{(1)}$, with the convention that the nodes S and T in $\boldsymbol{C}^{(1)}$ where unlabeled. This composition will be abbreviated as $\boldsymbol{C}^{\boldsymbol{u}}$, where $\boldsymbol{u} = (0, 1)$.

Notation 3. *More generally, given* $\boldsymbol{u} = (u_0, \ldots, u_{m-1}) \in \{0,1\}^m$*, we will denote by* $\boldsymbol{C}^{\boldsymbol{u}}$ *the composition* $\boldsymbol{C}^{(u_0)} \bullet \cdots \bullet \boldsymbol{C}^{(u_{m-1})}$*, and the set of all such compositions by* $\mathcal{C}_{2^m}$ *(as an example see* $\mathcal{C}_{2^3}$ *in Fig. 2).*

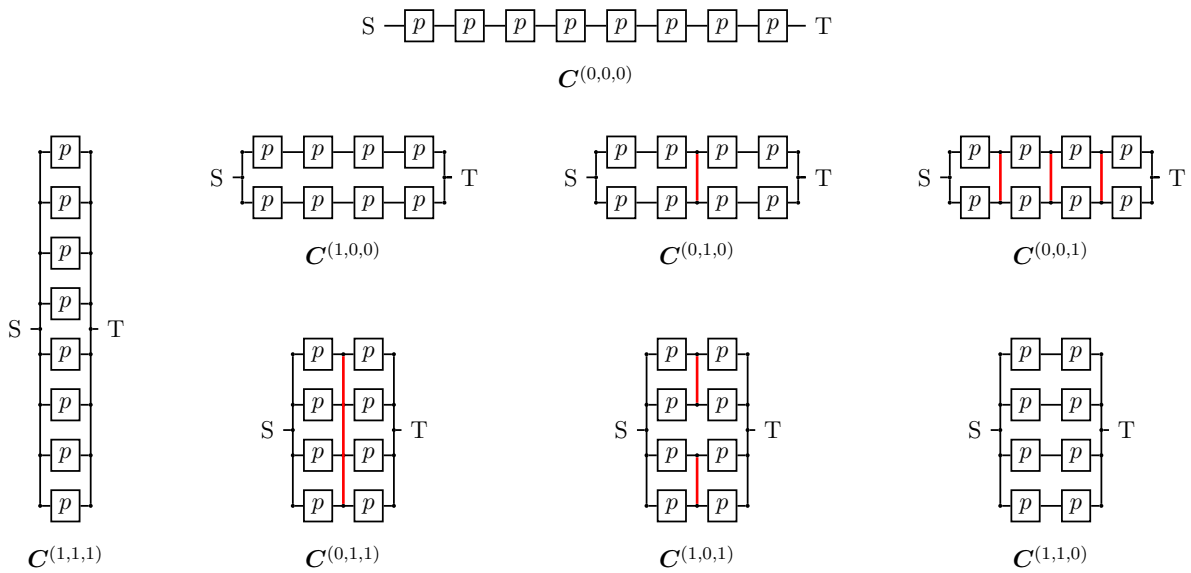


Figure 2: All the elements of the set $\mathcal{C}_{2^3}$.

Theorem 4. *Let* m *be a strictly positive integer. Then any element in* $\mathcal{C}_{2^m}$ *is a matchstick minimal network of size* 2^m*. Moreover, we have* $\#\mathcal{C}_{2^m} = 2^m$*.*

Proof: The fact that compositions of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ are matchstick minimal networks follows from Theorem 3 in [20]. The set of compositions of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ is:

$$\mathcal{C}_{2^m} = \{\boldsymbol{C}^{\boldsymbol{u}} | \boldsymbol{u} \in \{0,1\}^m\}, \tag{3}$$

from which it follows immediately that $\#\mathcal{C}_{2^m} = 2^m$. □

Notice that Proposition 4 provides an efficient method for generating matchstick minimal networks of size $n = 2^m$. Further we will detail how to compute w and l for any network $\boldsymbol{C} \in \mathcal{C}_{2^m}$. To achieve this goal, we introduce the well-known concept of Hamming weight from coding theory. For any binary vector $\boldsymbol{u} \in \{0,1\}^m$, its Hamming weight $|\boldsymbol{u}|$ is the number of non-zero components of $\boldsymbol{u}$.

Theorem 5. *Let* m *be a strictly positive integer and* $\boldsymbol{C}^{\boldsymbol{u}} \in \mathcal{C}_{2^m}$*. Then* $\boldsymbol{C}^{\boldsymbol{u}}$ *is a matchstick minimal network of size* 2^m*, length* $l = 2^{m-|\boldsymbol{u}|}$ *and width* $w = 2^{|\boldsymbol{u}|}$*.*

Proof: For a binary m-tuple $\boldsymbol{u} = (u_0, \ldots, u_{m-1}) \in \{0,1\}^m$ the weight $|\boldsymbol{u}|$ gives the number of times $\boldsymbol{C}^{(1)}$ is present in the composition, i.e., the number of times we compose in parallel. By induction it can be deduced that $w = 2^{|\boldsymbol{u}|}$. Since $\boldsymbol{C}^{\boldsymbol{u}}$ has $n = 2^m$ devices, it follows that $l = 2^{m-|\boldsymbol{u}|}$. □

In conclusion, writing

$$\mathcal{C}_{2^i,2^{m-i}} = \left\{ \mathcal{C}_{2^{|\boldsymbol{u}|},2^{m-|\boldsymbol{u}|}} | \boldsymbol{u} \in \{0,1\}^m, |\boldsymbol{u}| = i \right\} \tag{4}$$

we have

$$\mathcal{C}_{2^m} = \bigcup_{i=0}^{m} \mathcal{C}_{2^i,2^{m-i}} \tag{5}$$

3.2 Combinatorial properties

Since compositions of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ offer an efficient way of creating matchstick minimal networks, the first question that we raise here is to determine the proportion of compositions. In other words, if one randomly picks a matchstick minimal network $\boldsymbol{N} \in \mathcal{N}_{2^m}$,with respect to the uniform distribution over the set of all matchstick minimal networks, then "What is the probability that $\boldsymbol{N} \in \mathcal{C}_{2^m}$?"

Theorem 6. *Let $\boldsymbol{N}$ be a matchstick minimal network of size 2^m. Then we have*

$$Pr(\boldsymbol{N} \in \mathcal{C}_{2^m}) \sim 2^{-(2^{\lfloor m/2 \rfloor}-1)(2^{\lceil m/2 \rceil}-1)+m}.$$

Proof: From [7] we have that for a fixed l and w such that $n = wl$ the number of matchstick minimal networks of length l and width w equals $2^{(l-1)(w-1)}$. Hence the total number of matchstick minimal networks of size 2^m is equal to

$$\sum_{i=0}^{m} 2^{(2^i-1)(2^{m-i}-1)} \sim 2^{(2^{\lfloor m/2 \rfloor}-1)(2^{\lceil m/2 \rceil}-1)}, \tag{6}$$

which ends our proof. □

So, if we randomly pick a matchstick minimal network $\boldsymbol{N}$, the probability that $\boldsymbol{N}$ is a composition of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ is rapidly decreasing while m is increasing. However, the question now is how to determine whether $\boldsymbol{N}$ is an element of $\mathcal{C}_{2^m}$. To answer this question we define the following two operations

Definition 7 (Vertical/horizontal cut)**.** Let n, w, l be strictly positive integers and $\boldsymbol{N}$ be a matchstick minimal network of width w and length l. We say that $\boldsymbol{N}$ admits a *vertical cut* if there exists a vertical complete matchstick, and we write $\boldsymbol{N} = (\boldsymbol{N}_l | \boldsymbol{N}_r)$. We say that $\boldsymbol{N}$ admits a *horizontal cut* if there is a horizontal free band, i.e. with no matchsticks,and we write $\boldsymbol{N} = \begin{pmatrix} \boldsymbol{N}_u \\ \hline \boldsymbol{N}_d \end{pmatrix}$.

Lemma 8. *Let w and l be strictly positive integers and $\boldsymbol{N}$ be a wl matchstick minimal network. Then $\boldsymbol{N}$ can not admit both a horizontal and a vertical cut.*

Proof: The result is straightforward from the definition of a horizontal and vertical cuts (Definition 7). □

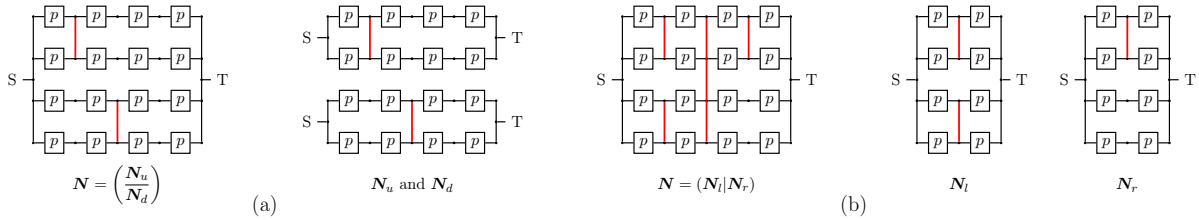


Figure 3: Matchstick minimal networks that admit: (a) horizontal cut; (b) vertical cut.

In this article we only consider those vertical and horizontal cuts that bisect the network into two identical halves. In other words a wl -network $\boldsymbol{N}$ can be cut vertically or horizontally and we write $\boldsymbol{N} = (\boldsymbol{N}_l|\boldsymbol{N}_r)$ with $\boldsymbol{N}_r = \boldsymbol{N}_l$, or $\boldsymbol{N} = \left(\frac{\boldsymbol{N}_u}{\boldsymbol{N}_d}\right)$ with $\boldsymbol{N}_u = \boldsymbol{N}_d$, where $\boldsymbol{N}_l$ is a $w \times l/2$ network and $\boldsymbol{N}_u$ is a $w/2 \times l$ network.

Theorem 9 (Decomposable networks)**.** *Let m be a strictly positive integer and $\boldsymbol{N}$ be a matchstick minimal network of size $n = 2^m$.*

- *if $\boldsymbol{N} = (\boldsymbol{N}_l|\boldsymbol{N}_l)$ (i.e., admits a vertical cut in half), then $\boldsymbol{N} = \boldsymbol{C}^{(0)} \bullet \boldsymbol{N}_l$;*
- *if $\boldsymbol{N} = \left(\frac{\boldsymbol{N}_u}{\boldsymbol{N}_u}\right)$ (i.e., admits a horizontal cut in half), then $\boldsymbol{N} = \boldsymbol{C}^{(1)} \bullet \boldsymbol{N}_u$.*

Moreover, $\boldsymbol{N} \in \mathcal{C}_{2^m}$ if and only if $\boldsymbol{N}$ admits a binary tree decomposition, with respect to "$|$" and "$-$", of depth m, where the leaves of the tree are $\boldsymbol{N}_{1,1}$.

The proof of this proposition is based on the previous Lemma.

Remark 10. Notice that $\boldsymbol{H}_{w,l}$ is not decomposable as a compositions of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ unless $w = 1$, $l = 1$, or $w = l = 2$, as it does not admit either a vertical or horizontal cut.

Algorithm 1 Decomposition of matchstick minimal networks into composition of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$

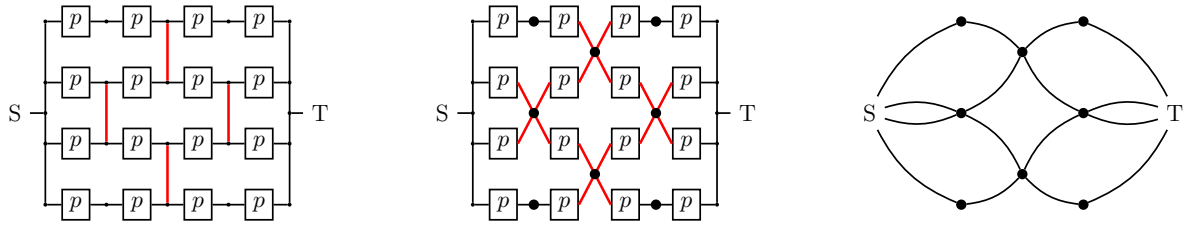
Input: A matchstick minimal network $\boldsymbol{N}$ of size $w \times l = 2^m$
Output: The corresponding binary vector $\boldsymbol{u}$ if $\boldsymbol{N}$ is decomposable

$\boldsymbol{u} = [\quad]$
while $\boldsymbol{N} \neq \boldsymbol{N}_{1,1}$ **do**
  if $\boldsymbol{N} = (\boldsymbol{N}_l|\boldsymbol{N}_l)$ **then**
    $\boldsymbol{N} = \boldsymbol{N}_l$
    Append 0 to $\boldsymbol{u}$
  else if $\boldsymbol{N} = \left(\frac{\boldsymbol{N}_u}{\boldsymbol{N}_u}\right)$ **then**
    $\boldsymbol{N} = \boldsymbol{N}_u$
    Append 1 to $\boldsymbol{u}$
  else
    Break;
  end if
end while

3.3 Representations

In order to compare compositions with Hammocks we consider the parameters of the circuits, that is the number of devices n, as well as the number of wires, ω in the circuits . The first representation of the brick-wall pattern, which is from Moore and Shannon [20], uses vertical matchsticks as in Fig. 1. The second possibility also suggested by Moore and Shannon [20] is to use the graph representation. Here, we will adopt the third representation from [21], which gave the name to these networks: *hammocks*. These three representations can all be seen in Fig. 4.

When counting the number of wires ω we will consider that there are w wires that connect S to the circuit, and w wires that connect T to the circuit. We also count 4 wires whenever we have an "X" shape matchstick. With this convention at hand we can now count ω for a matchstick minimal network, in particular a composition or a hammock.



Brick-wall representation ([20]) Hammock representation ([21]) Graph representation ([20])

Figure 4: Three different representations of $\boldsymbol{H}_{4,4}$.

Theorem 11. *Let m be a strictly positive integer and $\boldsymbol{C^u} \in \mathcal{C}_{2^m}$. Then the number of wires of the circuit $\boldsymbol{C^u}$ is $\omega = 2^m + 2^{i+1}$, where i is the position of the most significant bit of the corresponding binary vector $\boldsymbol{u}$ equal to 1. When $\boldsymbol{u}$ is the zero vector the number of wires is $\omega = 2^m + 1$.*

Proof: The first case $\boldsymbol{u} = (0, \ldots, 0) \in \{0,1\}^m$ can be easily deduced from the definition of the composition. Next consider $\boldsymbol{u} = (1, 0, 0, \ldots, 0) \in \{0,1\}^m$. This circuit $\boldsymbol{C^u}$ is a parallel of two identical circuits $\boldsymbol{C^v}$ where $\boldsymbol{v} = (0, \ldots, 0) \in \{0,1\}^{m-1}$. But we know that the number of wires for $\boldsymbol{C^v}$ is $\omega = 2^{m-1}+1$ and the number of devices for $\boldsymbol{C^v}$ equals 2^{m-1}. We also deduce that there are $2^m/2^{m-1} = 2$ identical blocks in the composition of $\boldsymbol{C^u}$. Notice that these two blocks do not share any wire in common. Hence, we obtain the total number of wires for $\boldsymbol{C^u}$, that equals the number of blocks times the number of wires in each block. More exactly the number of wires for $\boldsymbol{C^u}$ is $\omega = 2 \times \left(2^{m-1} + 1\right) = 2^m + 2$.

Now we can prove our theorem for any $\boldsymbol{u} \in \{0,1\}^m$. Let $\boldsymbol{u} = (u_0, \ldots, u_{m-1})$ be a binary vector such that $u_i = 1$ and $u_j = 0$ for any $j > i$. Denote $\boldsymbol{u}_{i,m-1} = (u_i, \ldots, u_{m-1})$, which equals $\boldsymbol{u}_{i,m-1} = (1, 0, \ldots, 0)$. Notice that $\boldsymbol{C^{u_{i,m-1}}}$ is composed of 2^{m-i} devices and $2^{m-i}+2$ wires. We also know that there are $2^m/2^{m-i}$ identical blocks, all equal to $\boldsymbol{C^{u_{i,m-1}}}$, that do not share any wire in common such that $\boldsymbol{C^u}$ is the composition of these blocks. Then the number of wires of $\boldsymbol{C^u}$ is $\omega = 2^i \times \left(2^{m-i} + 2\right) = 2^m + 2^{i+1}$.

□

Theorem 12. *Let w and l be two strictly positive integers. Then*

$$\omega = \begin{cases} 2wl - l & for \quad \boldsymbol{H}_{w,l} \quad with\ w\ and\ l\ even \\ 2wl - l + 1 & for \quad \boldsymbol{H}_{w,l} \quad with\ w\ or\ l\ odd \\ 2wl - l + 2 & for \quad \boldsymbol{H}^{+}_{w,l} \quad with\ w\ and\ l\ even \end{cases} \tag{7}$$

Proof: We give here only the proof for one of the cases. For the remaining two cases the arguments are exactly the same. So, let l and w be two odd strictly positive integers. This implies that we have $l-1$ columns of "X" shape matchsticks and horizontal wires. On each one of these columns we count $(w-1)/2$ matchsticks and one horizontal wire. Hence, each column has $4\times((w-1)/2)+1$ wires. And there are $l-1$ such columns, which makes the total number of "interior" wires equal to $(l-1)\times(2w-1)$. By "interior" wire we mean wires that connect only devices and not S or T with any of the devices. Finally, we have to add the number of "exterior" wires, namely those connecting to S and T, which are $2w$. Thus, we obtain $\omega = 2wl - l + 1$. □

Corollary 13. *For square hammocks we obtain*

$$\omega = \begin{cases} 8k^2-2k & for \quad \boldsymbol{H}_{2k,2k} \\ 8k^2-2k+2 & for \quad \boldsymbol{H}^{+}_{2k,2k} \\ 8k^2+6k+2 & for \quad \boldsymbol{H}_{2k+1,2k+1} \end{cases} \tag{8}$$

Remark 14. From eq. (8) taking $k = 2^{m/2}$ it follows that $\boldsymbol{H}_{2^{m/2},2^{m/2}}$ has $8\times\left(2^{m/2-1}\right)^2 - 2^{m/2} = 2^m + (2^m - 2^{m/2})$ wires. Also notice that from Theorem 11 there are $\binom{m}{m/2}/2$ elements in $\mathcal{C}_{2^{m/2},2^{m/2}}$ having at most $2^m + 2^{m-1}$ wires, which is smaller than $2^m + (2^m - 2^{m/2})$.

4 Evaluating reliability

4.1 Reliability polynomials

We will use a classical convention for the reliability polynomial $R(p)$, where $p \in [0,1]$ is the probability that a device is closed. Since the polynomial is associated with a network $\boldsymbol{N}$ (either $\boldsymbol{H}$ or $\boldsymbol{C}$), we shall use the notation $R(\boldsymbol{N};p)$. This gives $R(\boldsymbol{C};p)$ and $R(\boldsymbol{H};p)$ for compositions of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ and respectively hammocks.

Lemma 15. *$R(\boldsymbol{C}^{(0)};p) = p^2$ and $R(\boldsymbol{C}^{(1)};p) = 1-(1-p)^2$.*

This is well-known [22], [20]. We can now determine the reliability $R(\boldsymbol{C};p)$ for any $\boldsymbol{C}$.

Theorem 16. *Let m be a strictly positive integer and $\boldsymbol{u} = (u_0,\ldots,u_{m-1}) \in \{0,1\}^m$. Then:*

$$R(\boldsymbol{C}^{\boldsymbol{u}};p) = R(\boldsymbol{C}^{(u_0)}) \circ \cdots \circ R(\boldsymbol{C}^{(u_{m-1})};p), \tag{9}$$

where $R(\boldsymbol{C}^{(0)};p)$ and $R(\boldsymbol{C}^{(1)};p)$ are given by Lemma 15.

The proof of Theorem 16 follows directly from Definition 2 and Lemma 15.

Remark 17. Notice that compositions of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ are by definition series and parallel networks. Hence, they inherit all the nice properties of this big family of networks. Series and parallel networks were extensively studied [24], [18], [11] and efficient algorithms exist for computing their reliability polynomials (the complexity of these algorithms is linear in n). However, as we have shown in Theorem 16, compositions of $\boldsymbol{C}^{(0)}$ and $\boldsymbol{C}^{(1)}$ admit a closed form formula of complexity $\log_2(n)$ for computing their reliability polynomials.

Table 1: Reliability polynomials for $\boldsymbol{C^u}$.

$\boldsymbol{N}$	$R(\boldsymbol{N};p)$
$\boldsymbol{C}^{(1,1,1,0,0,0)}$	$8p^8-28p^{16}+56p^{24}-70p^{32}+56p^{40}-28p^{48}+8p^{56}-p^{64}$
$\boldsymbol{C}^{(1,1,0,1,0,0)}$	$16p^8-16p^{12}-92p^{16}+192p^{20}+112p^{24}-720p^{28}+698p^{32}+384p^{36}$ $-1552p^{40}+1744p^{44}-1116p^{48}+448p^{52}-112p^{56}+16p^{60}-p^{64}$
$\boldsymbol{C}^{(1,0,1,1,0,0)}$	$32p^8-96p^{12}-120p^{16}+1424p^{20}-4424p^{24}+8304p^{28}-10894p^{32}+10560p^{36}$ $-7744p^{40}+4320p^{44}-1816p^{48}+560p^{52}-120p^{56}+16p^{60}-p^{64}$
$\boldsymbol{C}^{(0,1,1,1,0,0)}$	$64p^8-448p^{12}+1680p^{16}-4256p^{20}+7952p^{24}-11424p^{28}+12868p^{32}-11440p^{36}$ $+8008p^{40}-4368p^{44}+1820p^{48}-560p^{52}+120p^{56}-16p^{60}+p^64$
$\boldsymbol{C}^{(1,1,0,0,1,0)}$	$64p^8-128p^{10}+96p^{12}-32p^{14}-1532p^{16}+6144p^{18}-10752p^{20}+10752p^{22}+9664p^{24}-95616p^{26}$ $+269664p^{28}-450464p^{30}+441338p^{32}+118784p^{34}-1729536p^{36}+4486144p^{38}-7423040p^{40}$ $+8938624p^{42}-8199136p^{44}+5857184p^{46}-3294716p^{48}+1464320p^{50}-$ $512512p^{52}+139776p^{54}-29120p^{56}+4480p^{58}-480p^{60}+32p^{62}-p^{64}$
$\boldsymbol{C}^{(1,0,1,0,1,0)}$	$128p^8-256p^{10}-320p^{12}+1472p^{14}-5496p^{16}+15616p^{18}+7200p^{20}-138656p^{22}+254648p^{24}+104576p^{26}-$ $1062432p^{28}+1528032p^{30}-17422p^{32}-3037184p^{34}+4820608p^{36}-3005056p^{38}-1494624p^{40}$ $+5473536p^{42}-6668992p^{44}+5345344p^{46}-3166616p^{48}+1441024p^{50}$ $-509600p^{52}+139552p^{54}-29112p^{56}+4480p^{58}-480p^{60}+32p^{62}-p^{64}$
$\boldsymbol{C}^{(0,1,1,0,1,0)}$	$256p^8-512p^{10}-2688p^{12}+9088p^{14}+5904p^{16}-61952p^{18}+61632p^{20}+165440p^{22}-454320p^{24}+141568p^{26}$ $+1016256p^{28}-1785920p^{30}+443716p^{32}+2654720p^{34}-4588384p^{36}+2904160p^{38}+1526280p^{40}$ $-5480576p^{42}+6670048p^{44}-5345440p^{46}+3166620p^{48}-1441024p^{50}$ $+509600p^{52}-139552p^{54}+29112p^{56}-4480p^{58}+480p^{60}-32p^{62}+p^{64}$
$\boldsymbol{C}^{(1,0,0,1,1,0)}$	$512p^8-3072p^{10}+8960p^{12}-16640p^{14}-43744p^{16}+765312p^{18}-4637568p^{20}+18013760p^{22}-51204560p^{24}$ $+113425312p^{26}-203255568p^{28}+301928416p^{30}-378028286p^{32}+403556352p^{34}-370208768p^{36}$ $+293307392p^{38}-201225472p^{40}+119608832p^{42}-61506048p^{44}+27263232p^{46}-10354528p^{48}$ $+3339648p^{50}-903168p^{52}+201152p^{54}-35952p^{56}+4960p^{58}-496p^{60}+32p^{62}-p^{64}$
$\boldsymbol{C}^{(0,1,0,1,1,0)}$	$1024p^8-6144p^{10}+1536p^{12}+114176p^{14}-542144p^{16}+1039104p^{18}+797952p^{20}-11825024p^{22}$ $+43312992p^{24}-105270976p^{26}+196334304p^{28}-297069632p^{30}+375202628p^{32}-402199296p^{34}$ $+369674944p^{36}-293137856p^{38}+201182992p^{40}-119600736p^{42}+61504944p^{44}-27263136p^{46}$ $+10354524p^{48}-3339648p^{50}+903168p^{52}-201152p^{54}+35952p^{56}-4960p^{58}+496p^{60}-32p^{62}+p^{64}$
$\boldsymbol{C}^{(0,0,1,1,1,0)}$	$4096p^8-57344p^{10}+415744p^{12}-2050048p^{14}+7653632p^{16}-22887424p^{18}+56715264p^{20}$ $-119066112p^{22}+214987136p^{24}-337392384p^{26}+463591296p^{28}-560492800p^{30}+598138512p^{32}$ $-564338304p^{34}+470897216p^{36}-347203584p^{38}+225750336p^{40}-129016384p^{42}$ $+64511136p^{44}-28048704p^{46}+10518296p^{48}-3365856p^{50}+906192p^{52}-201376p^{54}$ $+35960p^{56}-4960p^{58}+496p^{60}-32p^{62}+p^{64}$

For hammocks we rely on the results just published in [7] as well on the ones for $\boldsymbol{H}_{8,8}$ and $\boldsymbol{H}^{+}_{8,8}$ recently reported in [10], [1]. The associated reliability polynomials were computed using our own recursive depth-first traversal of a binary tree algorithm, and are reported in Table 2.

Table 2: Reliability polynomials for $\boldsymbol{H}^{+}_{8,8}$ and $\boldsymbol{H}_{8,8}$.

$\boldsymbol{N}$	$R(\boldsymbol{N};p)$
$\boldsymbol{H}_{8,8}$	$650p^8 - 580p^9 + 908p^{10} - 6880p^{11} + 4628p^{12} - 12104p^{13} + 31618p^{14} + 372p^{15} + 10594p^{16} + 196688p^{17} - 404536p^{18}$ $+915388p^{19} - 5608084p^{20} + 7645892p^{21} - 12887466p^{22} + 56185408p^{23} - 61734474p^{24} + 83601572p^{25}$ $-412397124p^{26} + 272424760p^{27} + 274694424p^{28} + 1746408000p^{29} - 221980272p^{30} - 12868843904p^{31}$ $+11123958002p^{32} - 11120041788p^{33} + 156260690872p^{34} - 378857360436p^{35} + 264833158482p^{36}$ $-60539595908p^{37} + 345161573768p^{38} + 1581294699620p^{39} - 10357633700988p^{40} + 19594821559752p^{41}$ $-7205288635438p^{42} - 36413539831436p^{43} + 75842387925382p^{44} - 55098726855452p^{45} - 30641343744796p^{46}$ $+111186328020944p^{47} - 111483252211446p^{48} + 33001245825824p^{49} + 53388841078258p^{50} - 85170175686428p^{51}$ $+59759032847258p^{52} - 15870886733412p^{53} - 12944378218252p^{54} + 19685718553176p^{55} - 14268363534224p^{56}$ $+7162471625508p^{57} - 2694331712884p^{58} + 775005119032p^{59} - 169487849178p^{60} + 27440435336p^{61}$ $-3113881376p^{62} + 221751056p^{63} - 7474305p^{64}$
$\boldsymbol{H}^{+}_{8,8}$	$720p^8 - 720p^9 + 1052p^{10} - 7864p^{11} + 6482p^{12} - 16012p^{13} + 43042p^{14} - 16492p^{15} + 35378p^{16} + 202080p^{17} - 418416p^{18}$ $+840000p^{19} - 6142350p^{20} + 7346188p^{21} - 11370674p^{22} + 74129792p^{23} - 100005860p^{24} + 118520824p^{25}$ $-656753496p^{26} + 1014391664p^{27} - 1060302334p^{28} + 5318496368p^{29} - 8451329352p^{30} + 2451624096p^{31}$ $-37298482094p^{32} + 119852403404p^{33} - 23621628548p^{34} - 197506250928p^{35} - 337635320852p^{36}$ $+1438498163768p^{37} - 67797908976p^{38} - 4198255335740p^{39} + 3015682674902p^{40} + 8881103035456p^{41}$ $-15082194064786p^{42} - 3506508481748p^{43} + 18452358491432p^{44} + 39640921395644p^{45} - 181496618059380p^{46}$ $+286828005143040p^{47} - 191038611524520p^{48} - 138468526731136p^{49} + 542912067034010p^{50} - 803232038481876p^{51}$ $+814719587176720p^{52} - 634769854840740p^{53} + 396340290321940p^{54} - 202017905414696p^{55} + 84634369170678p^{56}$ $-29121695246028p^{57} + 8171460088944p^{58} - 1843848199008p^{59} + 327008804562p^{60} - 43949841128p^{61}$ $+4211763728p^{62} - 256604464p^{63} + 7474305p^{64}$

4.2 Figures-of-merit

To compare compositions with hammocks we will rely on several *FoMs*:

1. The well-known *Reliability Improvement Index* (RII) introduced by Klaschka in [15], [16];
2. The steepness of $R(\boldsymbol{N};p)$ mentioned by Moore and Shannon [20];
3. The variation of $R(\boldsymbol{N};p)$, which was mentioned in [10].

Reliability Improvement Index

The *Reliability Improvement Index* is defined [15], [16] for any network $\boldsymbol{N}$ as

$$\mathrm{RII}(\boldsymbol{N}) = \frac{\log(p)}{\log\left(R(\boldsymbol{N};p)\right)}. \tag{10}$$

The RII is a measure of the reliability increase produced by a network $\boldsymbol{N}$ and was used in [7] to estimate how much matchstick minimal networks improve on a single device.

Steepness of the reliability polynomials

The *ideal* reliability function proposed by Moore and Shannon is the staircase function:

$$\chi(p) = \begin{cases} 0 & 0 \le p \le 0.5 \\ 1 & 0.5 < p \le 1 \end{cases}$$

One of the goal of [20] was to identify networks having reliability polynomials exhibiting steep $0 \to 1$ transitions. We define FoM_1 as:

$$FoM_1 = \max_{p\in[0,1]} R'(\boldsymbol{N}; p). \tag{11}$$

Since the transition point might be important, we give a finer FoM for the steepness of the polynomials. This is an enhancement over FoM_1 which measures how steep is the reliability polynomial as well as how far from 0.5 is the threshold. So, in general FoM_1^* is equal to FoM_1 weighted by a function of the distance between p_0 and 0.5. Here, we choose a very simple function, that is

$$FoM_1^* = \max_{p\in[0,1]} R'(\boldsymbol{N}; p) \cdot \frac{1}{|0.5 - p_0|}, \tag{12}$$

where p_0 is the point where the maximum is achieved. Notice that in our case this is well-defined since no network studied in this paper has $p_0 = 0.5$. But this is no longer the case for self-dual networks where $p_0 = 0.5$, and a modified FoM_1 should be proposed.

Variation of the reliability polynomial

Another FoM is introduced in this paper. It is related to the variation achieved by a reliability polynomial in a given interval. We shall use the area under $R'(\boldsymbol{N}; p)$ in a given symmetric interval (with respect to 0.5). This is exactly the variation of $R(\boldsymbol{N}; p)$ on that interval, hence for any $\boldsymbol{N}$:

$$FoM_2(p_0) = R(\boldsymbol{N}; 1 - p_0) - R(\boldsymbol{N}; p_0) = \int_{p_0}^{1-p_0} R'(\boldsymbol{N}; p)dp. \tag{13}$$

This FoM_2 is well-defined for the staircase function since χ may be written as the integral of the delta Dirac function over the sub-domain $[0, 1]$. Therefore, FoM_2 is the area under the delta Dirac function, between two symmetric points t and $1 - t$, with $0 \leq t < 0.5$.

4.3 Numerical results

Reliability improvement index

The first set of simulations was performed over the whole set $\mathcal{C}_{2^6}$ as well as for the two hammocks under investigations, $\boldsymbol{H}_{8,8}$ and $\boldsymbol{H}^+_{8,8}$ (see Table 2). Using eq. (10) we have calculated all the RII*s*. These can be seen in Fig. 5,6,7. In Fig. 5 the scale is linear to get a clear picture of the very large RII values for p close to 1. In includes only the square networks, more exactly $\boldsymbol{N} \in \mathcal{C}_{8,8}$ in blue and $\boldsymbol{N} = \boldsymbol{H}_{8,8}, \boldsymbol{H}^+_{8,8}$ in red. A zoom in on the region of interest is shown in Fig. 6, where the yellow horizontal line at RII = 1 represents the border between networks that improve reliability and networks which do not. Finally, the complete picture (Fig. 7), in log scale, includes all networks $\boldsymbol{N} \in \mathcal{C}_{2^6}$, the non-square ones being plotted in orange.

This figure shows a wide range of variation for RII*s*. Among these, those which go below RII = 1 are not improving over a single device, which means they should not be used. This is in support of selecting square networks which tend to stick together close to RII = 1 when $p = 0.5$.

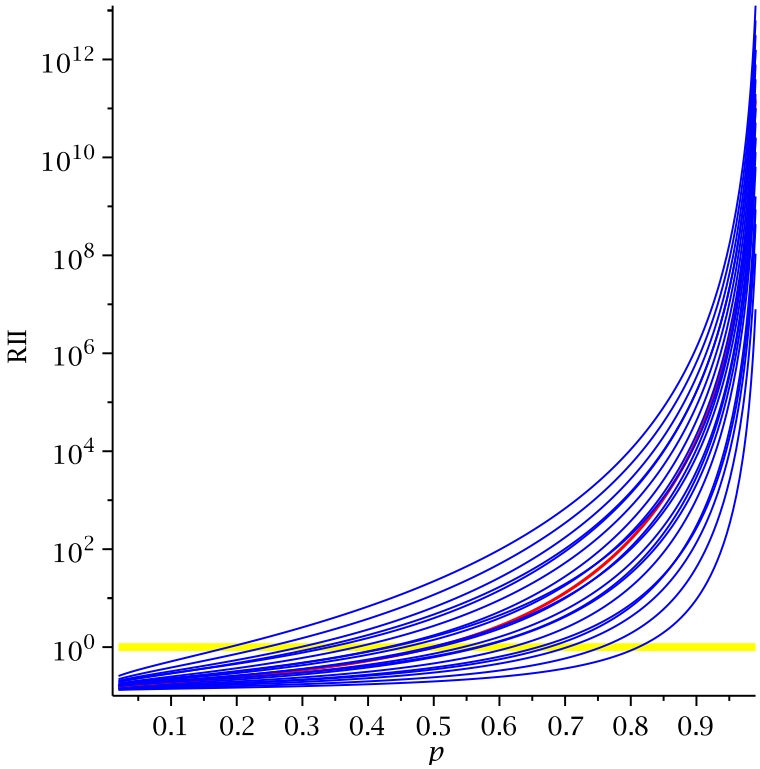


Figure 5: RII($\boldsymbol{N}$) for $\boldsymbol{N} \in \mathcal{C}_{8,8}$ (blue) and $\boldsymbol{N} = \boldsymbol{H}_{8,8}, \boldsymbol{H}^{+}_{8,8}$ (red).

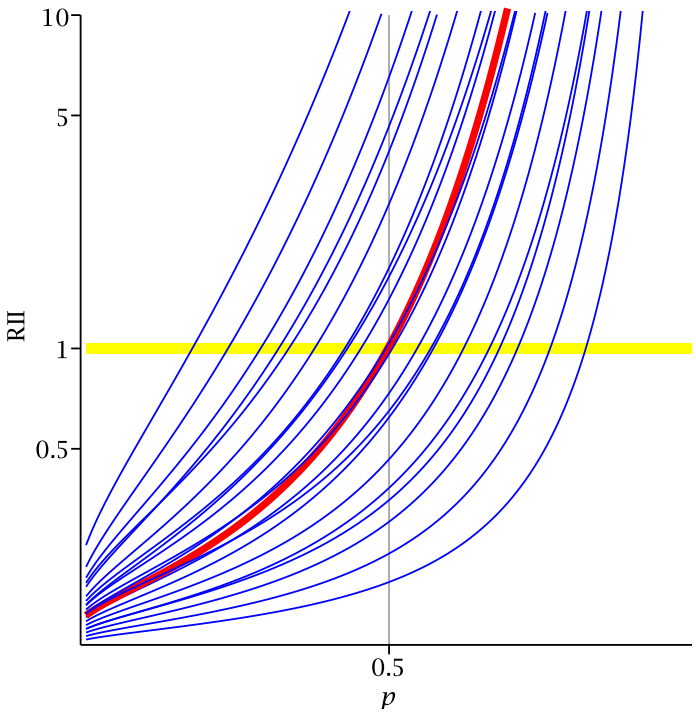


Figure 6: Zoom on RII($\boldsymbol{N}$) for $\boldsymbol{N} \in \mathcal{C}_{8,8}$ (blue) and $\boldsymbol{N} = \boldsymbol{H}_{8,8}, \boldsymbol{H}^{+}_{8,8}$ (red).

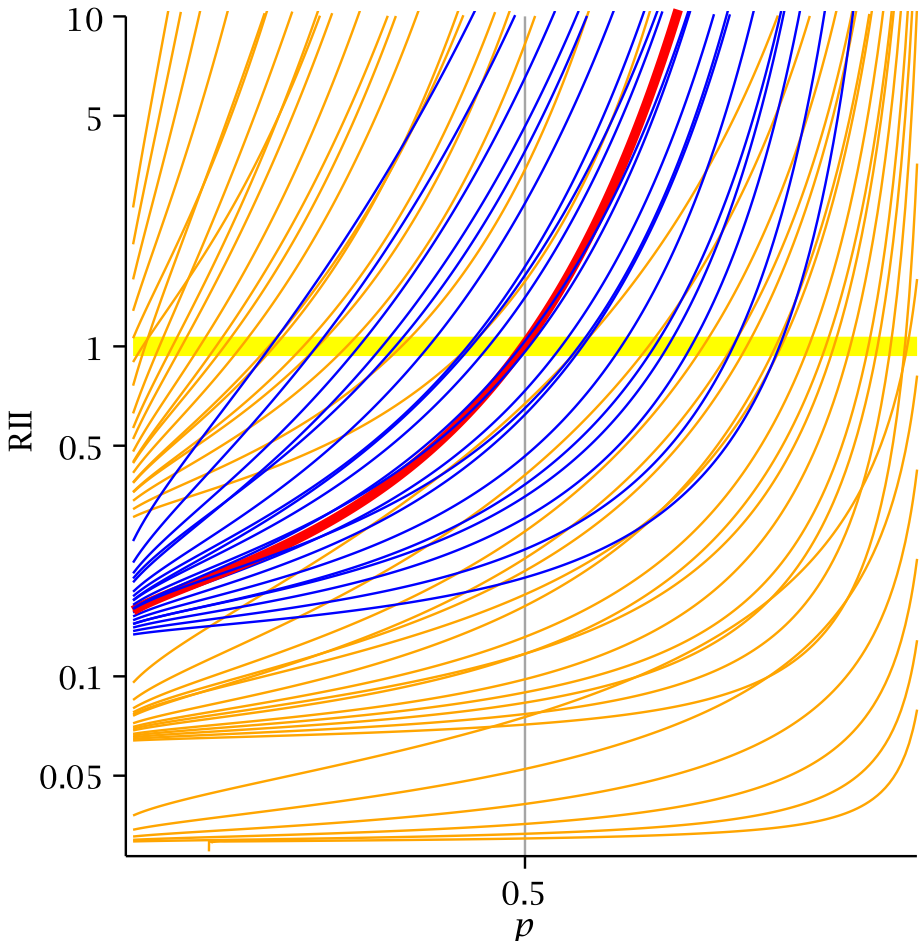


Figure 7: RII($\boldsymbol{N}$) for $\boldsymbol{N} \in \mathcal{C}_{8,8}$ (blue), $\boldsymbol{N} = \boldsymbol{H}_{8,8}, \boldsymbol{H}^{+}_{8,8}$ (red) and $\boldsymbol{N} \in \mathcal{C}_{2^6} \setminus \mathcal{C}_{8,8}$ (orange) in log scale.

Steepness of the reliability polynomials

In Fig. 8 we plot $R'(\boldsymbol{N}; p)$ for all $\boldsymbol{N} \in \mathcal{C}_{8,8}$, as well as $\boldsymbol{N} = \boldsymbol{H}_{8,8}$ and $\boldsymbol{N} = \boldsymbol{H}^{+}_{8,8}$. We notice that $\max\limits_{p\in[0,1]} R'(\boldsymbol{H}_{8,8}; p) = \max\limits_{p\in[0,1]} R'(\boldsymbol{H}^{+}_{8,8}; p) = 3.75252$ and is reached at $p_0 = 0.501745$, respectively $p_0 = 0.498255$. For compositions we have:

$$\max_{\boldsymbol{C}\in\mathcal{C}_{8,8}} \left(\max_{p\in[0,1]} R'(\boldsymbol{C}; p) \right) = 4.1035,$$

which is achieved by $\boldsymbol{u} = (1, 1, 1, 0, 0, 0)$ at $p_0 = 0.760$.

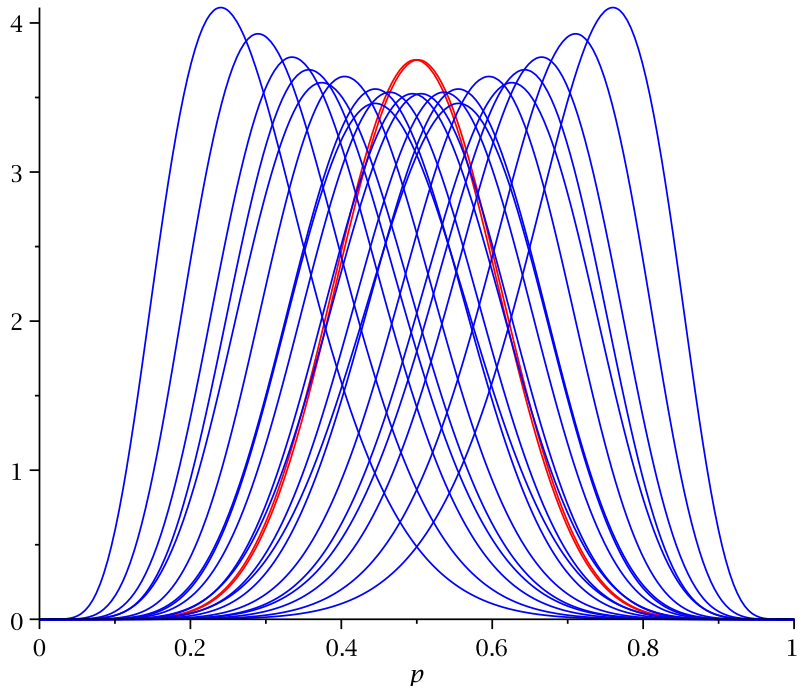


Figure 8: $R'(\boldsymbol{N};p)$

This result shows the limitation of this FoM, as it does not take into account how far the threshold point is from the desired point, that is 0.5. That is why the enhanced version of this FoM, namely FoM_1^* gives better results in comparing reliability. This fact is illustrated in Fig. 9 and 10.

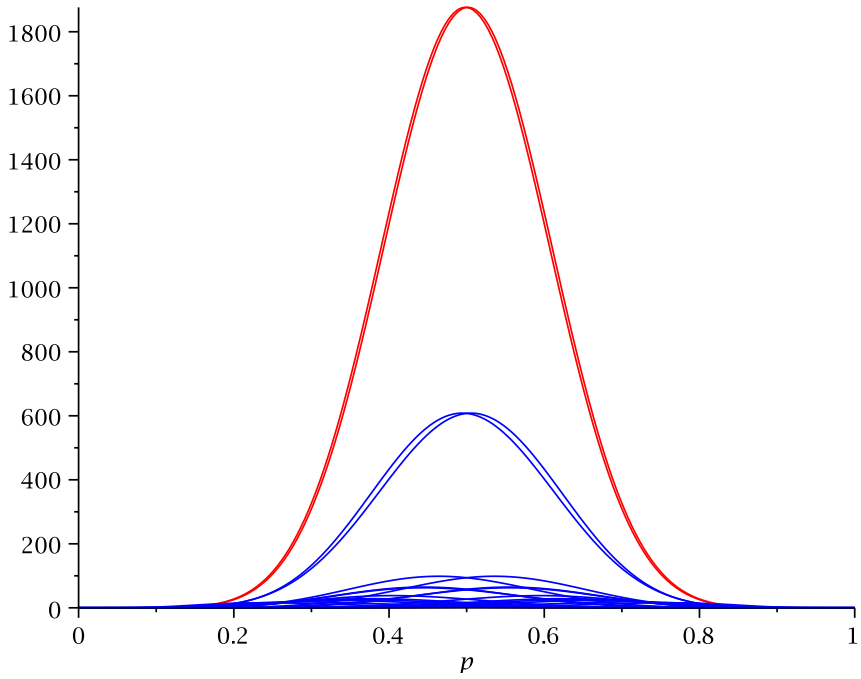


Figure 9: $\dfrac{R'(\boldsymbol{N};p)}{|p_0-0.5|}$

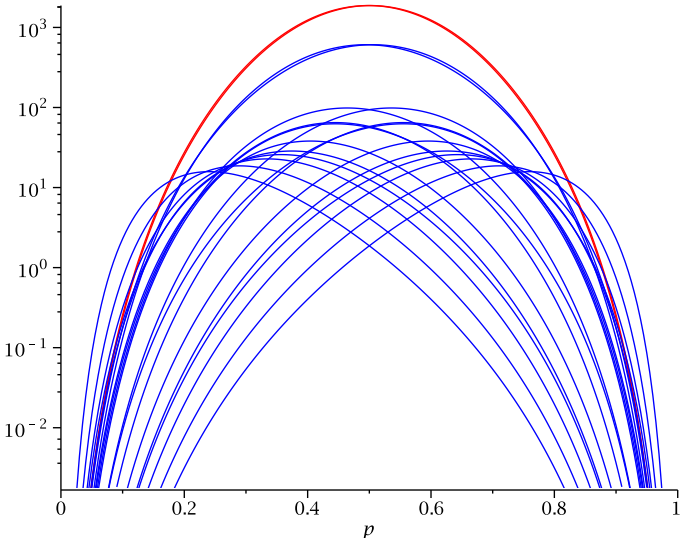


Figure 10: $\log\left(\dfrac{R'(\boldsymbol{N};p)}{|p_0-0.5|}\right)$

We observe from Fig. 9 and 10 that hammocks are "better" than compositions with respect to FoM_1^*. Indeed, for hammocks we obtain $FoM_1^*(\boldsymbol{H}_{8,8}) = 1876.25$ and for compositions we have

$$\max_{\boldsymbol{C}\in\mathcal{C}_{8,8}}\left(\frac{\max\limits_{p\in[0,1]} R'(\boldsymbol{C};p)}{|p_0-0.5|}\right) = 587.56,$$

which is achieved by $\boldsymbol{u}=(0,1,0,1,1,0)$ and $\boldsymbol{u}=(1,0,1,0,0,1)$.

Variation of the reliability polynomials

In Fig. 11 we plot $R(\boldsymbol{N};1-p_0)-R(\boldsymbol{N};p_0)$ as a function of $0\le p_0<0.5$ for $\boldsymbol{H}_{8,8}$ and $\boldsymbol{H}^+_{8,8}$, and $\boldsymbol{C}\in\mathcal{C}_{8,8}$.

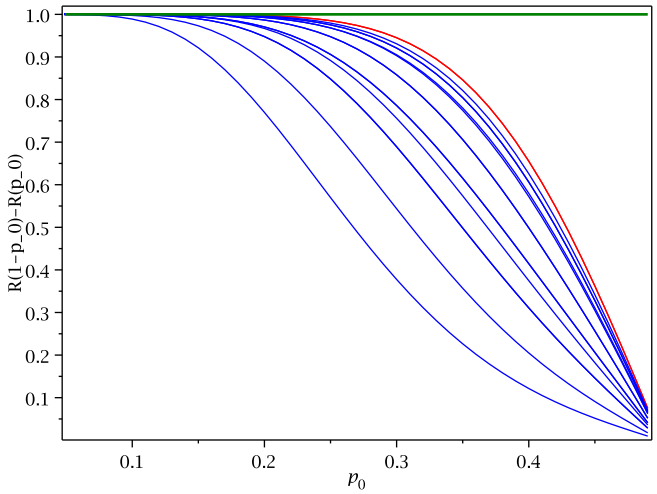


Figure 11: $R(\boldsymbol{N};1-p_0)-R(\boldsymbol{N};p_0)$ as a function of p_0 $(0\le p_0<0.5)$;$\boldsymbol{N}=\boldsymbol{H}_{8,8}$ and $\boldsymbol{H}^+_{8,8}$ (red), $\boldsymbol{N}=\boldsymbol{C}\in\mathcal{C}_{8,8}$ (blue), and $\chi(1-p_0)-\chi(p_0)$ (green).

We notice a difference between the curves starting to develop from $p_0 = 0.25$ onwards. In Table 3 we report the exact values for $p_0 = 0.25$, which correspond to $R(0.75) - R(0.25)$. The two hammocks we have considered here achieve the same value $R(\boldsymbol{H}_{8,8}; 0.75) - R(\boldsymbol{H}_{8,8}; 0.25) = 0.985173$. For compositions the best value

$$\max_{\boldsymbol{u}\in\{0,1\}^6} R(\boldsymbol{C}^{\boldsymbol{u}}; 0.75) - R(\boldsymbol{C}^{\boldsymbol{u}}; 0.25) = 0.979507,$$

is achieved for $\boldsymbol{u} = (0, 1, 0, 1, 1, 0)$ and $\boldsymbol{u} = (1, 0, 1, 0, 0, 1)$. It should be mentioned that the same two compositions achieve the best values for FoM_1^*. In fact, $FoM_2(0.25)$ correlates perfectly with FoM_1^*. Indeed, if we totally order compositions and hammocks with respect to FoM_1^*, then the same order holds for $FoM_2(0.25)$. And as expected, the two FoMs point out to the same network as being the most reliable, namely the hammock.

Recall that one of the leading arguments when comparing networks was the restriction of the number of devices, which in our case study is always $n = 64$. Now, if we plot $FoM_2(0.25)$ as a function of the number of wires ω, we see that one of the best compositions, namely $\boldsymbol{u} = (0, 1, 0, 1, 1, 0)$ has fewer wires than the hammocks (see Fig. 12).

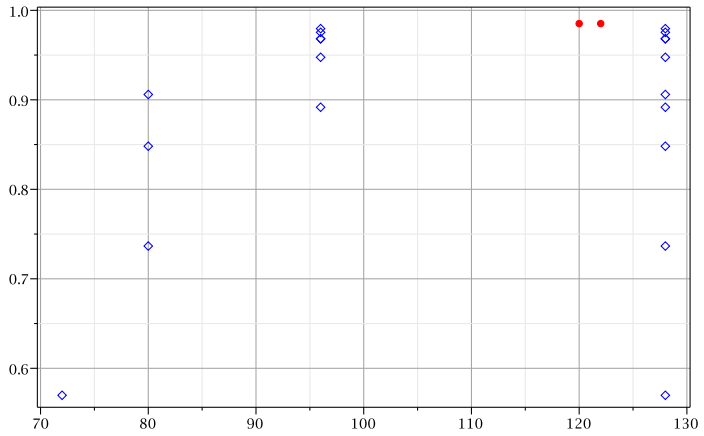


Figure 12: $R(\boldsymbol{N}; 0.75) - R(\boldsymbol{N}; 0.25)$ as a function of the number of wires ω. $\boldsymbol{N} = \boldsymbol{H_{8,8}}$ and $\boldsymbol{N} = \boldsymbol{H^{+}_{8,8}}$ (red), and $\boldsymbol{N} = \boldsymbol{C}$ with $\boldsymbol{C} \in \mathcal{C}_{8,8}$ (blue)

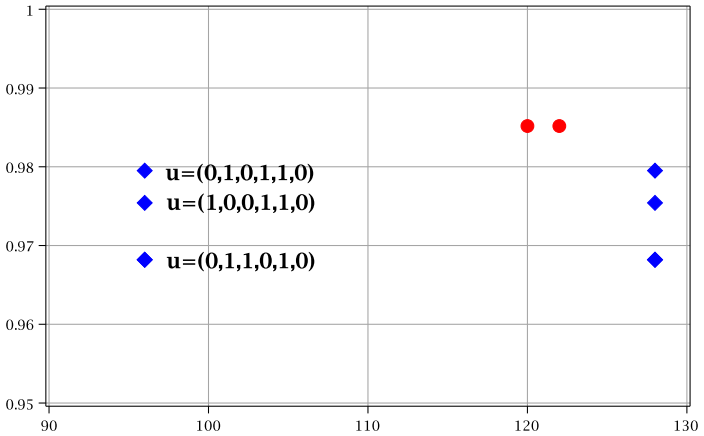


Figure 13: Zoom on the values of $R(\boldsymbol{N}; 0.75) - R(\boldsymbol{N}; 0.25)$ in the vicinity of the maximum values

Table 3: Figure-of-merit for $\boldsymbol{C} \in \mathcal{C}_{8,8}$, $\boldsymbol{H}_{8,8}$, and $\boldsymbol{H}^{+}_{8,8}$.

ω	$\boldsymbol{N}$	p_0	$\max_{p\in[0,1]} R'(\boldsymbol{N};p)$	$\dfrac{\max_{p\in[0,1]} R'(\boldsymbol{N};p)}{\lvert p_0-0.5\rvert}$	$R(\boldsymbol{N};0.75)-R(\boldsymbol{N};0.25)$
120	$\boldsymbol{H}_{8,8}$	0.498	3.7525	1876.25	0.985173
122	$\boldsymbol{H}^{+}_{8,8}$	0.502	3.7525	1876.25	0.985173
72	$\boldsymbol{C}^{(1,1,1,0,0,0)}$	0.760	4.1035	15.78	0.569843
80	$\boldsymbol{C}^{(1,1,0,1,0,0)}$	0.710	3.9273	18.70	0.736601
	$\boldsymbol{C}^{(1,0,1,1,0,0)}$	0.665	3.7709	22.85	0.848154
	$\boldsymbol{C}^{(0,1,1,1,0,0)}$	0.626	3.5995	28.56	0.905953
96	$\boldsymbol{C}^{(1,1,0,0,1,0)}$	0.643	3.6861	25.77	0.891705
	$\boldsymbol{C}^{(1,0,1,0,1,0)}$	0.596	3.6409	37.92	0.947539
	$\boldsymbol{C}^{(0,1,1,0,1,0)}$	0.555	3.5568	64.66	0.968192
	$\boldsymbol{C}^{(1,0,0,1,1,0)}$	0.536	3.5354	98.20	0.975413
	$\boldsymbol{C}^{(0,1,0,1,1,0)}$	0.494	3.5254	587.56	0.979507
	$\boldsymbol{C}^{(0,0,1,1,1,0)}$	0.445	3.4606	62.92	0.968192
128	$\boldsymbol{C}^{(1,1,0,0,0,1)}$	0.555	3.4606	62.92	0.968192
	$\boldsymbol{C}^{(1,0,1,0,0,1)}$	0.506	3.5254	587.56	0.979507
	$\boldsymbol{C}^{(0,1,1,0,0,1)}$	0.464	3.5354	98.20	0.975413
	$\boldsymbol{C}^{(1,0,0,1,0,1)}$	0.445	3.5568	64.66	0.968192
	$\boldsymbol{C}^{(0,1,0,1,0,1)}$	0.404	3.6409	37.92	0.947538
	$\boldsymbol{C}^{(0,0,1,1,0,1)}$	0.357	3.6861	25.77	0.891705
	$\boldsymbol{C}^{(1,0,0,0,1,1)}$	0.374	3.5995	28.56	0.905953
	$\boldsymbol{C}^{(0,1,0,0,1,1)}$	0.335	3.7709	22.85	0.848154
	$\boldsymbol{C}^{(0,0,1,0,1,1)}$	0.290	3.9273	18.70	0.736601
	$\boldsymbol{C}^{(0,0,0,1,1,1)}$	0.240	4.1035	15.78	0.569843

Strong points of compositions

Notice that for $\boldsymbol{u} = (0,1,0,1,1,0)$ the corresponding composition comes really close to $\boldsymbol{H}_{8,8}$ and has an advantage over $\boldsymbol{H}_{8,8}$, in that $\boldsymbol{C}^{(0,1,0,1,1,0)}$ has only 96 wires, while $\boldsymbol{H}_{8,8}$ has 120. From a computational point of view, $R(\boldsymbol{C}^{\boldsymbol{u}};p)$ has several advantages over $R(\boldsymbol{H}_{8,8};p)$.

- Firstly, notice that the order of magnitude of the largest coefficient is 10^9 for $\boldsymbol{C}^{\boldsymbol{u}}$ compared with 10^{14} for $\boldsymbol{H}_{8,8}$.

- Secondly, the reliability polynomials for compositions are sparser than $R(\boldsymbol{H}_{8,8};p)$. This is due to the fact that $R(\boldsymbol{C}^{\boldsymbol{u}};p)$ have non-zero coefficients only for even powers of p, i.e., 29 non-zero coefficients versus 57 for $R(\boldsymbol{H}_{8,8};p)$.

- Thirdly the absolute values of the coefficients of $R(\boldsymbol{H}_{8,8};p)$ are larger on average than the coefficients of $R(\boldsymbol{C}^{\boldsymbol{u}};p)$. For the case $m = 6$, the average value of a coefficient is of the order 1.3×10^{13} for hammocks, compared with 8.6×10^{7} for compositions.

From the computational point of view all these arguments favor compositions over hammocks.

5 Conclusions

In this article we have proposed and analyzed two-terminal networks generated through the repeated composition of the simplest series and parallel networks. We have detailed several structural properties of such networks and have presented an efficient method for computing their associated reliability polynomials.

Compositions were compared with hammocks according to three FoMs: RII, the slope of the reliability polynomials and their variations. For the particular cases considered here we have observed that compositions come very close to hammocks, without surpassing them. Still, compositions of series and parallel present several advantages. There are compositions performing almost as well as hammocks, while having fewer wires than hammocks, for the same number of devices. We also noticed that there is a significant computational gap, the reliability polynomials of compositions being much simpler/easier to compute and analyze exactly.

Acknowledgement

Research supported in part by the European Union (EU) through the European Research Development Fund (ERDF) under the Competitiveness Operational Program (COP): *BioCell-NanoART = Novel Bio-inspired Cellular Nano-architectures*, POC-A1-A1.1.4-E nr. 30/01.09.2016.

Bibliography

[1] Beiu, V.; Cowell, S.R.; Drăgoi, V.; Hoară, S.; Gaşpar, P. (2018); Hammocks versus hammock, *2018 7th International Conference on Computers Communications and Control (ICCCC), Proc. of,* Oradea, Romania, May 2018, Publisher: IEEE, 119–123, 2018. DOI: 10.1109/ICCCC.2018.8390447

[2] Beiu, V.; L. Dăuş, L.; Rohatinovici, N.-C.; Bălaş, V. E. (2018); Transport reliability on axonal cytoskeleton, *Proc. Intl. Conf. Eng. Modern Electr. Syst. (EMES), Oradea, Romania, Jun. 2017*, 160–163, 2017.

[3] Ball, M.O.; Colbourn, C.J.; Provan, J.S. (1992); Network reliability, *Tech. Rep. TR 92-74*, Systems Research Center/ Institute for System Research, University of Maryland, College Park, MD, USA, June 1992.

[4] Barlow, R. E.; Proschan, F. (1965); *Mathematical Theory of Reliability*, John Wiley & Sons, New York, NY, 1965.

[5] Colbourn, C.J. (1991); Combinatorial aspects of network reliability, *Annals of Operations Research*, 33(1), 3 – 15, Jan. 1991.

[6] Courtland, R. (2016); The next high-performance transistor, *IEEE Spectr.*, 53(10), 11–12, Oct. 2016.

[7] Cowell, S.R.; Beiu, V.; Dăuş, L.; Poulin, P. (2018); On the exact reliability enhancements of small hammock networks, *IEEE Access*, 6(1), 25411–25426, Apr. 2018. [Early version as "On hammock networks – Sixty years after", *Proc. Intl. Conf. Design & Technol. Integr. Syst. Nanoscale Era (DTIS)*, Palma de Mallorca, Spain, Apr. 2017, art. 7929871]

[8] Cowell, S.R.; Beiu, V.; Dăuş, L.; Poulin, P. (2017); On cylindrical hammock networks, *Proc. Intl. Conf. Nanotech. (IEEE-NANO), Pittsburgh, PA, USA, Jul. 2017*, 185–188, 2017.

[9] Deng, H.; Chen, J.; Q. Li,Q.; Li,R.; Gao, Q. (2004); On the construction of most reliable networks, *Discr. Appl. Maths.*, 140(1-3), 19–33, 2004.

[10] Drăgoi, V.; Cowell, S.R.; Hoară, S.; Gaşpar, P.; Beiu, V. (2018); Can series and parallel compositions improve on hammocks?, *Proc. of 2018 7th International Conference on Computers Communications and Control (ICCCC)*, Oradea, Romania, May 2018, Publisher: IEEE, 124–130, 2018. DOI: 10.1109/ICCCC.2018.8390448

[11] Duffin, R.J. (1965); Topology of series-parallel networks, *Journal of Mathematical Analysis and Applications*, 10(2), 303–318, 1965.

[12] Geppert, L. (2002); The amazing vanishing transistor act, *IEEE Spectr.*, 239(10), 8–33, 2002.

[13] IEEE Rebooting Computing, https://rebootingcomputing.ieee.org/

[14] International Roadmap for Devices and Systems, *(IRDS)*, 2017 [Online]. Available: https://irds.ieee.org/roadmap-2017

[15] Klaschka, T.F. (1967); Two contributions to redundancy theory, *Proc. Annual Symposium on Switching and Automata Theory (SWAT)*, Austin, TX, USA, Oct. 1967, 175–183, 1967. doi: 10.1109/FOCS.1967.36

[16] Klaschka, T.F. (1971); A method for redundancy scheme performance assessment, *IEEE Transactions on Computers*, C-20(11), 1371–1376, 1971. doi: 10.1109/T-C.1971.223141

[17] Kuo, W.; Zuo, M.J. (2003); *Optimal Reliability Modeling: Principles and Applications*, J. Wiley & Sons, Hoboken, NJ, USA, 2003.

[18] Lee, C.Y. (1955); Analysis of switching networks, *Bell System Technical Journal*, 34(6), 1287–1315, 1955. doi:10.1002/j.1538-7305.1955.tb03799

[19] Li, Q.; Li, Q. (1998); Reliability analysis of circulant graphs, *Networks*, 31(2), 61–65, Mar. 1998.

[20] Moore, E.F.; Shannon, C.E. (1956); Reliable circuits using less reliable relays – Part I, *J. Frankl. Inst.*, 262(3), 191–208, 1956.

[21] Moore, E.F.; Shannon, C.E. (1956); Reliable circuits using less reliable relays – Part II, *J. Frankl. Inst.*, 262(4), 281–297, 1956.

[22] von Neumann, J. (1952, 1956); Probabilistic logics and the synthesis of reliable organisms from unreliable components, Jan. 1952 . Also in C. E. Shannon, and J. McCarthy (Eds.): *Automata Studies*, Princeton Univ. Press, Princeton, NJ, USA, 43–98, Apr. 1956.

[23] Theis, T.N.; Wong, H.-S.P. (2017); The end of Moore's law: A new beginning for information technology, *Comput. Sci. & Eng.*, 19(2), 41–50, 2017.

[24] Wald, J.A.; Colbourn, C.J. (1983); Steiner trees, partial 2-trees, and minimum IFI networks, *Networks*, vol. 13, no. 2, pp. 13(2), 159–167, 1983. doi:10.1002/net.3230130202

[25] Weichenberg, G.E.; Chan, V.W.S.; Medard, M. (2004); High-reliability topological architectures for networks under stress, *IEEE J. Sel. Areas Comm.*, 22(9), 1830–1845, 2004.

[26] Williams, R.S. (2017); What's next?, *Comput. Sci. & Eng.*, 19(2), 7–13, 2017.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 792-807, October 2018.

Generalized Ordered Propositions Fusion Based on Belief Entropy

Y. Li, Y. Deng

Yangxue Li, Yong Deng*
Institute of Fundamental and Frontier Science
University of Electronic Science and Technology of China, Chengdu
*Corresponding author: dengentropy@uestc.edu.cn

Abstract: A set of ordered propositions describe the different intensities of a characteristic of an object, the intensities increase or decrease gradually. A basic support function is a set of truth-values of ordered propositions, it includes the determinate part and indeterminate part. The indeterminate part of a basic support function indicates uncertainty about all ordered propositions. In this paper, we propose generalized ordered propositions by extending the basic support function for power set of ordered propositions. We also present the entropy which is a measure of uncertainty of a basic support function based on belief entropy. The fusion method of generalized ordered proposition also be presented. The generalized ordered propositions will be degenerated as the classical ordered propositions in that when the truth-values of non-single subsets of ordered propositions are zero. Some numerical examples are used to illustrate the efficiency of generalized ordered propositions and their fusion.
Keywords: Dempster-Shafer evidence theory, ordered proposition, uncertainty measure, belief entropy, information fusion.

1 Introduction

In resent year, with the intensification of competition in the modern information war, information technology has developed rapidly, and the amount of information has increased explosively. Thus, as the critical technologies for information collection, storage and processing, the essentiality of information modeling and fusion has gradually increased.

There are many methods to model information, such as probability theory [10], Dempster-Shafer evidence theory [7,23], rough sets [40], fuzzy sets [6,7,9,19,19,22–24,24], Z-numbers [17, 37,37], D numbers [1,5,76] and as so on. A specialized fusion algorithm is used for each method. Ordered proposition is a new approach to model information which is proposed by Liu *et al.* [35]. A set of ordered propositions describe the different intensities of a characteristic of a objects, the intensities increase or decrease gradually. For example, consumers evaluate the quality of a product on a rank of "Wonderful, Good, Indifferent, Weak". A set of ordered propositions can be expressed as a basic support function (similar to belief function in Dempster-Shafer evidence theory), whose elements represent the truth-value (belief value) of each proposition. The truth-values of a basic support function must be convex, because a subject cannot be two degrees in same characteristic. Such as, we cannot say the quality of a product is both wonderful and indifference simultaneously.

A basic support function is divided into determinate part and indeterminate part [35]. The determinate part is the sum of truth-values of each ordered proposition. The sum of indeterminate part and determinate part is one. In the ordered propositions fusion, the indeterminate part is prorated to each proposition and itself. Therefore, the indeterminate part can express the uncertainty for all ordered propositions. In this paper, we define the generalized ordered propositions, they extend the indeterminate part to all non-single subsets of ordered propositions. The truth-value of a non-single subset expresses the uncertainty of the propositions in

it. For example, the "Wonderful, Good" express "the quality of this product is wonderful" or "the quality of this product is good". In order to ensure the convex property of a basic support function, the indeterminate part is listed separately. The generalized ordered propositions will be degenerated as the classical ordered propositions in that when the truth-values of non-single subsets of ordered propositions are zero.

The ordered propositions fusion is an important and extensive problem [35]. Previously, a fusion algorithm based on centroid is proposed [42], which fuse the basic support functions of two sets of ordered propositions and ensure the convexity. However, this approach has a few shortages [35]. In order to address these shortages, a new fusion method based on consistency and uncertainty measurements was presented by Liu *et al.* for the fusion of ordered proposition [35]. They also introduced entropy to measure the uncertainty of the basic support function based on Shannon entropy [35]. But this entropy only considered the determinate part of a basic support function, the indeterminate part is ignored. In Dempster-Shafer evidence theory, an entropy is presented to measure the uncertainty of a belief function, named Deng entropy [6]. When we add the groups of propositions in ordered propositions, the basic support function is more similar with the belief function. In this paper, we introduce a new entropy to measure the uncertainty of a basic support function based on belief entropy. It will be degenerated as the entropy which is proposed by Liu *et al.* in that when the indeterminate part of a basic support function is zero. Additionally, the fusion method of generalized ordered propositions based on consistency and uncertainty measurements is introduced. When the truth-values of non-single subsets of ordered propositions are zero, the fusion result is same as the fusion result of Liu *et al.*'s method.

The rest part of this paper is organized as follows. Section 2 briefly discusses the definitions and properties of ordered propositions, Dempster-Shafer evidence theory and belief entropy. Section 3 introduces the definition and properties of generalized ordered propositions. Section 4 discusses the proposed method for measuring uncertainty of a basic support function. The fusion method of generalized ordered proposition is described in Section 5. Section 6 presents some numerical examples. Finally, this paper is concluded in Section 7.

2 Preliminaries

2.1 Ordered propositions

In this section, some background knowledge about ordered propositions is briefly introduced [35].

Definition 1 (Ordered propositions)**.** For a set of propositions $p_1, p_2, \cdots, p_n$, the truth-value of p_i is denoted as $\lambda(p_i)$. $\lambda(p_k) = max\{\lambda(p_1), \cdots, \lambda(p_n)\}$. $p_1, p_2, \cdots, p_n$ are ordered propositions if [35]

(1) $\forall i = 1, 2, \cdots, n$, all subjects described in p_i are S;

(2) $\forall i = 1, 2, \cdots, n$, s_i describes the same characteristics or features of S;

(3) $\forall i = 1, 2, \cdots, k-1$, $\lambda(p_i) \leq \lambda(p_{i+1})$; and $\forall i = k, k+1, \cdots, n-1$, $\lambda(p_i) \geq \lambda(p_{i+1})$.

Definition 2 (Basic support function)**.** For a set of ordered propositions $P = \{p_1, p_2, \cdots, p_n\}$, a function λ is called the basic support function of the ordered propositions if [35]

(1) λ is defined on $\{\overline{P}\} \cup \{\{p_i\} | 1 \leq i \leq n\}$, where $\overline{P}$ indicates indeterminacy;

(2) $\lambda(p_i) \geq 0, 1 \leq i \leq n$;

(3) $\sum_{1\leq i\leq n}\lambda(p_i)\leq 1$;

(4) $\lambda(\overline{P})=1-\sum_{1\leq i\leq n}\lambda(p_i)$.

Definition 3 (Determinate part and indeterminate part)**.** For a basic support function λ, the determinate part $\lambda(P)$ and indeterminate part $\lambda(\overline{P})$ are defined as [35]

$$\lambda(S)=\sum_{i=1,\cdots,n}\lambda(p_i),\quad \lambda(\overline{P})=1-\lambda(P). \tag{1}$$

Definition 4 (Mean value)**.** The mean value of a basic support function λ is defined as [35]

$$\overline{\lambda}=\frac{\sum_{i=1}^{n}\lambda(p_i)}{n}. \tag{2}$$

Definition 5 (Measure of convexity)**.** The measure of convexity of a basic support function λ is defined as [35]

$$convex(\lambda)=max\{\lambda(p_1),\lambda(p_2),\cdots,\lambda(p_n)\}-\overline{\lambda}. \tag{3}$$

It was clear that the maximum of the measure of convexity is $1-\overline{\lambda}$. Thus, the normalized $convex(\lambda)$ as follows: [35]

$$NC(\lambda)=(max\{\lambda(p_1),\lambda(p_2),\cdots,\lambda(p_n)\}-\overline{\lambda})/(1-\overline{\lambda}). \tag{4}$$

Definition 6 (Center of a basic support function)**.** For a basic support function $\lambda=(\lambda(p_1),\lambda(p_2),\cdots,\lambda(p_n))$, the center of λ is defined as [35]

$$CI(\lambda)=\begin{cases} argmax_{i=1,\cdots,n}\lambda(p_i), & NC(\lambda)\geq\theta \\ \dfrac{\sum_{i=1,\cdots,n\wedge\lambda(p_i)\geq\tau\cdot\overline{\lambda}}\lambda(p_i)\times i}{\sum_{i=1,\cdots,n\wedge\lambda(p_i)\geq\tau\cdot\overline{\lambda}}\lambda(p_i)}, & otherwise, \end{cases} \tag{5}$$

θ is set to 0.55 in [35], $1<\tau\leq 1.5$.

In order to model the complex information of interaction, complex networks are proposed [4, 20,21,40,65,69]. The measure of consistency is essential to information, affected by the reliability of the information source [9, 11, 17, 32, 46, 64, 66, 74]. The reliability of obtaining data is very important for information fusion [41].

Definition 7 (Measure of consistency)**.** If $CI(\lambda_1)$ and $CI(\lambda_2)$ are the centers of the basic support functions λ_1 and λ_2.The consistency between λ_1 and λ_2 is defined as [35]

$$\Delta G(\lambda_1,\lambda_2)=|CI(\lambda_1)-CI(\lambda_2)|/(n-1). \tag{6}$$

If $\Delta G=1$, then λ_1 and λ_2 are totally conflicting. If $\Delta G=0$, then λ_1 and λ_2 are consistent. Otherwise, if

$0<\Delta G<1$, then λ_1 and λ_2 are partially conflicting. The consistency between λ_1 and λ_2 can be divided into 3 degrees [35].

$0\leq\Delta G\leq\delta_1$ indicates the consistency between λ_1 and λ_2 is high.

$\delta_1\leq\Delta G\leq\delta_2$ indicates the consistency between λ_1 and λ_2 is medium.

$\delta_2\leq\Delta G\leq 1$ indicates the consistency between λ_1 and λ_2 is poor.

2.2 Dempster-Shafer evidence theory

Evidence theory is widely used in many applications such as target recognition [29,30], decision making [1,11], uncertain processing [3,13,16,16,20,21,26–28,31,35], risk management [18,36], fault diagnosis [4, 15, 25, 56, 60] and as so on. The frame of discernment Θ is the exhaustive hypotheses of variable, X.
$\Theta = \{x_1, x_2, \cdots, x_i, \cdots, x_n\}$. The power set of Θ is $2^\Theta = \{\emptyset, \{x_1\}, \cdots, \{x_n\}, \{x_1, x_2\}, \cdots, \{x_1, x_2, \cdots, x_i\}, \cdots, \Theta\}$, where $\emptyset$ is an empty set [7, 23].

Definition 8 (Basic probability assignment (BPA))**.** A basic probability assignment function $m : 2^\Theta \to [0, 1]$, which satisfies [7, 23]:

$$m(\Theta) = 0 \quad \sum_{A \in 2^\Theta} m(A) = 1 \quad 0 \leq m(A) \leq 1, \tag{7}$$

the mass $m(A)$ indicates how strongly the evidence supports A.

2.3 Belief entropy

Shannon entropy is widely used to measure the uncertainty of a probability. In addition, a belief entropy named Deng entropy is proposed to measure the uncertainty of a BPA [6].

Definition 9 (Belief entropy)**.** For a BPA, m, defined on the frame of discernment Θ, it's belief entropy is defined as [6]

$$E_d(m) = -\sum_{A \subseteq \Theta} m(A) \ln \frac{m(A)}{2^{|A|} - 1}, \tag{8}$$

where A is the focal element of m, $|A|$ is the cardinality of A.

3 Generalized ordered propositions

3.1 Definitions

Definition 10 (Generalized ordered propositions)**.** For a set of propositions $p_1, p_2, \cdots, p_n$, it's power set, $\{\emptyset, \{p_1\}, \{p_2\}, \cdots, \{p_n\}, \{p_1, p_2\}, \cdots, \{p_1, \cdots, p_n\}\}$, let $\lambda(p_i, p_j \cdots)$ represent the truth-value of $\{p_i, p_j, \cdots\}$ and $\lambda(p_k) = max\{\lambda(p_1), \cdots, \lambda(p_n)\}$. $p_1, p_2, \cdots, p_n$ are generalized ordered propositions, if

(1)$\forall i = 1, 2, \cdots, n$, all subjects described in p_i are S;
(2)$\forall i = 1, 2, \cdots, n$, p_i describes the same characteristics or features of S;
(3)$\forall i = 1, 2, \cdots, m-1$, $\lambda(p_i) \leq \lambda(p_{i+1})$; and $\forall i = m, m+1, \cdots, n-1$, $\lambda(p_i) \geq \lambda(p_{i+1})$.

Definition 11 (Basic support function of the generalized ordered propositions)**.** For a set of generalized ordered propositions $P = \{p_1, p_2, \cdots, p_n\}$, it's power set $2^P = \{\emptyset, \{p_1\}, \{p_2\}, \cdots, \{p_n\}, \{p_1, p_2\}, \{p_1, p_3\}, \cdots, \{p_1, p_2, \cdots, p_n\}\}$ a function λ is called a basic support function of the generalized ordered propositions if

(1) λ is defined on 2^P;
(2) $\lambda(A) \geq 0, A \subseteq P$;
(3) $\lambda(\emptyset) = 0$;
(4) $\sum_{1 \leq i \leq n} \lambda(A) = 1$, where $A \subseteq P$;

Take the example of "the quality of a product", the basic support function is $\{(0.1, 0.3, 0.2, 0.0), (\lambda(p_1, p_2) = 0.2, \lambda(p_2, p_3) = 0.2)\}$.

$\lambda(p_1) = 0.1$ means the truth-value of 1st proposition "the quality of a product is wonderful" is 0.1;

$\lambda(p_2) = 0.3$ means the truth-value of 2nd proposition "the quality of a product is good" is 0.3;

$\lambda(p_3) = 0.2$ means the truth-value of 3rd proposition "the quality of a product is indifference" is 0.2;

$\lambda(p_4) = 0.0$ means the truth-value of 4rd proposition "the quality of a product is weak" is 0.0;

$\lambda(p_1, p_2) = 0.2$ means the uncertain truth-value of 1st proposition and 2nd proposition is 0.2;

$\lambda(p_2, p_3) = 0.2$ means the uncertain truth-value of 2st proposition and 3rd proposition is 0.2.

3.2 Properties

Definition 12 (Determinate part and indeterminate part)**.** For a basic support function of generalized ordered proposition, the determinate part and indeterminate part is

$$\lambda(P) = \sum_{i=1}^{n} \lambda(p_i), \quad \lambda(\overline{P}) = \sum_{A \subseteq P \wedge A \neq \{q_1\}, \cdots, \{q_n\}} \lambda(A) = 1 - \lambda(P). \tag{9}$$

Definition 13 (Mean value)**.** The mean value of a basic support function λ of generalized ordered propositions is

$$\overline{\lambda} = \frac{\sum_{i=1}^{n} \lambda(p_i)(1 + \sum_{p_i \subset A} \lambda(A))}{n}, \tag{10}$$

where $A \subsetneq \{p_1, p_2, \cdots, p_n\}$.

Definition 14 (Degree of convexity)**.** The degree of convexity of a basic support function λ of generalized ordered propositions is:

$$convex(\lambda) = max_{i=1,\cdots,n}\{\lambda(p_i)(1 + \sum_{p_i \subset A} \lambda(A))\} - \overline{\lambda}, i = 1, 2, \cdots, n, \tag{11}$$

where $A \subsetneq \{p_1, p_2, \cdots, p_n\}$.

The normalized $convex(\lambda)$ is

$$NC(\lambda) = (max_{i=1,\cdots,n}\{\lambda(p_i)(1 + \sum_{p_i \subset A} \lambda(A))\} - \overline{\lambda})/(1 - \overline{\lambda}), i = 1, 2, \cdots, n. \tag{12}$$

Definition 15 (Center of a basic support function)**.** A basic support function of generalized ordered propositions $\lambda = \{(\lambda(p_1), \lambda(p_2), \cdots, \lambda(p_n)), (\lambda(p_1, p_2), \cdots, \lambda(p_1, p_2, \cdots, p_n))\}$, the center of λ is

$$CI(\lambda) = \begin{cases} argmax_{i=1,\cdots,n} \lambda(p_i)(1 + \sum_{p_i \subset A} \lambda(A)), & NC(\lambda) \geq \theta \\ \dfrac{\sum_{i=1,\cdots,n \wedge \lambda(p_i) \geq \tau \cdot \overline{\lambda}} \lambda(p_i)(1 + \sum_{p_i \subset A} \lambda(A)) \times i}{\sum_{i=1,\cdots,n \wedge \lambda(p_i)(1 + \sum_{p_i \subset A} \lambda(A)) \geq \tau \cdot \overline{\lambda}} \lambda(p_i)(1 + \sum_{p_i \subset A} \lambda(A))}, & otherwise, \end{cases} \tag{13}$$

where $A \subsetneq \{p_1, p_2, \cdots, p_n\}$.

4 Uncertainty measure

Uncertainty can evaluate the quality of information [2,3,13,15,16,31,32,39,39,47,48,50,61,71]. The more uncertainty, the less information [7,8]. A method to measure the uncertainty of a basic support function of ordered propositions based on Shannon entropy is proposed by Liu *et al.* [35].

Definition 16 (Liu *et al.*'s entropy)**.** For a basic support function $\lambda = (\lambda(p_1), \lambda(p_2), \cdots, \lambda(p_n))$, $\lambda \neq (\lambda(p_1) = 0, \lambda(p_2) = 0, \cdots, \lambda(p_n) = 0)$ and $n \geq 2$. Let $\lambda(p_k) = max\{\lambda(p_1), \lambda(p_2), \cdots, \lambda(p_n)\}$, $1 \leq k \leq n$. If $\beta\lambda(p_k) \leq \lambda(p_j) \leq \lambda(p_k), \beta \geq 0.9$ and $1 \leq j \leq n$, then $\lambda(p_j)$ is quasi-maxima. Let n' is the total number of maxima and quasi-maxima. The Liu *et al.*'s entropy of λ is defined as: [35]

$$E(\lambda) = \begin{cases} -\sum_{i=1}^{n} \lambda(p_i) \ln \lambda(p_i), & n' = 1, \\ -\sum_{i=1}^{n} \lambda(p_i) \ln \lambda(p_i) + (\ln n + \sum_{i=1}^{n} \lambda(p_i) \ln \lambda(p_i)))(n'/n)^{\alpha}, & 2 \leq n' \leq n, \end{cases} \tag{14}$$

where $\alpha = 0.1$.

When indeterminate part of a basic support function is equal to zero, this entropy can accurately measure the uncertainty of a basic support function. For example, given two basic support functions $\mu_1 = (0.005, 0.99, 0.005, 0.0, 0.0)$, $\mu_2 = (0.0049995, 0.990001, 0.0049995, 0.0, 0.0)$, we can given $E(\mu_1) = 0.062933$ and $E(\mu_2) = 0.062928$ using Eq. (14). $E(\mu_1)$ is greater than $E(\mu_2)$, this means that the uncertainty of μ_1 is higher than the uncertainty of μ_2. The result is reasonable. When there are multiple maxima of a basic support function, Liu *et al.*'s method can also measure uncertainty accurately. Take two basic support functions $\mu_3 = (0.5, 0.5, 0.0, 0.0)$, $\mu_4 = (0.15, 0.7, 0.1, 0.05)$, then $E(\mu_3) = 1.34$ and $E(\mu_4) = 0.914$. It is reasonable that $E(\mu_3) > E(\mu_4)$.

However, when indetermination part of a basic support function is not equal to zero, this entropy doesn't apply to measure uncertainty of a basic support function. For example, for two basic support functions $\mu_5 = (0.2, 0.3, 0.0, 0.0)$ and $\mu_6 = (0.7, 0.1, 0.1, 0)$, then $E(\mu_5) = 0.6831$, $E(\mu_6) = 0.7103$. $E(\mu_5) < E(\mu_6)$, this means that the degree of uncertainty of μ_6 is higher. It is obviously counterintuitive. In order to take into considered not only the determinate part but also indeterminate part, we present the a new method to measure uncertainty of a basic support function of generalized ordered proposition based on belief entropy [1,6].

Definition 17 (The entropy based on belief entropy)**.** For a basic support function of generalized ordered propositions $\lambda = \{(\lambda(p_1), \lambda(p_2), \cdots, \lambda(p_n)), (\lambda(p_1, p_2), \lambda(p_1, p_3), \cdots, \lambda(p_1, p_2, \cdots, p_n))\}$, $\lambda \neq (\lambda(p_1) = 0, \lambda(p_2) = 0, \cdots, \lambda(p_n) = 0)$ and $n \geq 2$. Let $\lambda(p_k) = max\{\lambda(p_1), \lambda(p_2), \cdots, \lambda(p_n)\}$, $1 \leq k \leq n$. If $\beta\lambda(p_k) \leq \lambda(p_j) \leq \lambda(p_k), \beta \geq 0.9$ and $1 \leq j \leq n$, then $\lambda(q_j)$ is quasi-maxima. Let n' is the total number of maxima and quasi-maxima. The entropy of λ is defined as:

$$E_d(\lambda) = \begin{cases} -\sum_{i=1}^{n} \lambda(A) \ln(\frac{\lambda(A)}{2^{|A|-1}}), & n' = 1, \\ -\sum_{i=1}^{n} \lambda(A) \ln(\frac{\lambda(A)}{2^{|A|-1}}) + (\ln n + \lambda(A) \ln(\frac{\lambda(A)}{2^{|A|-1}}))(n'/n)^{\alpha}, & 2 \leq n' \leq n, \end{cases} \tag{15}$$

where $A \subseteq \{q_1, 1_2, \cdots, q_n\}$, $|A|$ is the number of elements of A, $\alpha = 0.1$.

Using Eq.(15) to calculate the uncertainty of μ_5 and μ_6, the results are $E_d(\mu_5) = 2.3837$, $E_d(\mu_6) = 1.2114$. $E_d(\mu_5) > E_d(\mu_6)$, it is reasonable. For two basic support functions of generalized ordered propositions $\mu_7 = \{(0.2, 0.5, 0.1, 0.0), (\mu_7(p_1, p_2) = 0, 1, \mu_7(p_2, p_3) = 0.1)\}$ and $\mu_8 =$

$\{(0.2, 0.6, 0.1, 0.0), (\mu_8(p_1, p_2) = 0.1)\}$. The results are $E_d(\mu_7 = 1.5790$ and $E_d(\mu_8) = 1.1988$ using Eq. (12). $E_d(\mu_7) > E_d(\mu_8)$, this means that the degree of uncertainty of μ_7 is higher than μ_8.

5 Fusion of generalized ordered propositions

For a set of generalized ordered propositions $P = \{p_1, p_2, \cdots, p_n\}$, let λ_1 and λ_2 are two basic support functions of P. Denote the fusion result of λ_1 and λ_2 is ω. The processes of method for fusion of basic support functions of generalized ordered propositions is shown in Fig. 5. The steps of this method can be explained as follows:

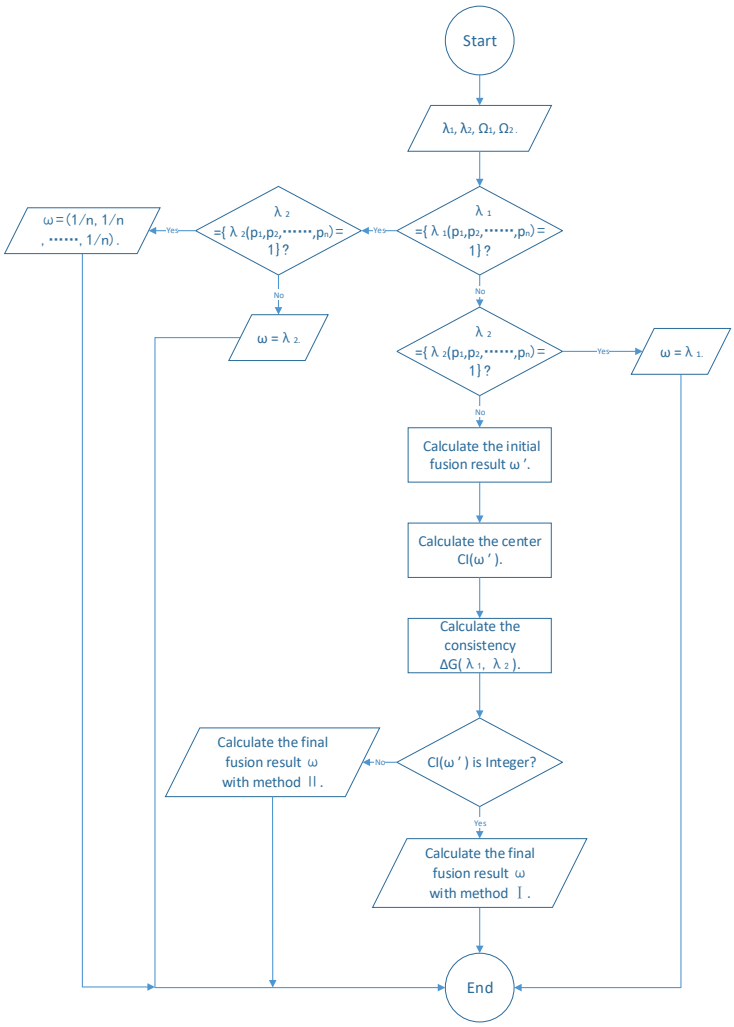


Figure 1: The processes of proposed method

Step 1: Give two basic support functions λ_1, λ_2 of a set of generalized ordered propositions $P = \{p_1, p_2, \cdots, p_n\}$, and the weights Ω_1, Ω_2 of two basic support functions respectively.

Table 1: Process of calculating ω' by Eq. (16).

	λ_1 to ω'	λ_2 to ω'	ω'
Truth-value obtained by $\omega'(p_1)$	0.025	0.08625	0.11125
Truth-value obtained by $\omega'(p_2)$	0.345	0.345	0.69
Truth-value obtained by $\omega'(p_3)$	0.08625	0.025	0.11125
Truth-value obtained by $\omega'(p_4)$	0.025	0.025	0.05
Truth-value obtained by $\omega'(p_1, p_2)$	0	0.01875	0.01875
Truth-value obtained by $\omega'(p_2, p_3)$	0.01875	0	0.01875

Step 2: Determine whether λ_1 is equal to $\{\lambda_1(p_1, p_2, \cdots, p_n) = 1\}$ and if λ_2 is equal to $\{\lambda_2(p_1, p_2, \cdots, p_n) = 1\}$.

If $\lambda_1 = \{\lambda_1(p_1, p_2, \cdots, p_n) = 1\}$ and $\lambda_2 = \{\lambda_2(p_1, p_2, \cdots, p_n) = 1\}$, the fusion result $\omega = (1/n, 1/n, \cdots, 1/n)$. If $\lambda_1 = \{\lambda_1(p_1, p_2, \cdots, p_n) = 1\}$ but $\lambda_2 \neq \{\lambda_2(p_1, p_2, \cdots, p_n) = 1\}$, the fusion result $\omega = \lambda_2$. If $\lambda_2 = \{\lambda_2(p_1, p_2, \cdots, p_n) = 1\}$ but $\lambda_1 \neq \{\lambda_1(p_1, p_2, \cdots, p_n) = 1\}$, the fusion result $\omega = \lambda_1$. If $\lambda_1 \neq \{\lambda_1(p_1, p_2, \cdots, p_n) = 1\}$ and $\lambda_2 \neq \{\lambda_2(p_1, p_2, \cdots, p_n) = 1\}$, take the next step.

Step 3: Calculate the initial fusion result.

$$\omega'(A) = \begin{cases} \Omega_1 \cdot \lambda_1(A)(1 + \sum\limits_{A \subset B} \lambda_1(B)) + \Omega_2 \cdot \lambda_1(A)(1 + \sum\limits_{A \subset C} \lambda_1(C)), & |A| = 1, \\ \Omega_1 \cdot \lambda_1(A)(1 - \sum\limits_{p_i \subset A} \lambda_1(p_i)) + \Omega_2 \cdot \lambda_2(A)(1 - \sum\limits_{p_i \subset A} \lambda_1(p_i)), & 1 < |A| \leq n, \end{cases} \tag{16}$$

where $A, B, C \subseteq \{p_1, p_2, \cdots, p_n\}$, $i = 1, 2, \cdots, n$, $\Omega_1 + \Omega_2 = 1$.

For example, there are two basic support functions $\lambda_1 = \{(0.05, 0.6, 0.15, 0.05), (\lambda_1(p_2, p_3) = 0.15\}$ and $\lambda_2 = \{(0.15, 0.6, 0.05, 0.05), (\lambda_2(p_1, p_2) = 0.15)\}$. The weights are $\Omega_1 = \Omega_2 = 0.5$. The process of calculating initial fusion result ω' by using Eq. (16) is illustrated in Table 1.

Step 4: Calculate the center of initial fusion result ω' with Eq. (13), $CI(\omega')$.

Step 5: Calculate the consistency between λ_1 and λ_2 with Eq. (6), $\Delta G(\lambda_1, \lambda_2)$.

Step 6: Determine whether the center of initial fusion result $CI(\omega')$ is Integer. If $CI(\omega')$ is Integer, take the step 7, otherwise take the step 8.

Step 7: Calculate the final fusion result ω with method I.

Step 7.1: Positive regulation.

$$\omega(p_i) = \begin{cases} \sum\limits_{k=1}^{i} \dfrac{\omega'(p_k)[1 + \varphi(i-k)]}{\sum_{j=0}^{CI(\omega')-k}(1 + j\varphi)}, & if\ i < CI(\omega'), \\ \sum\limits_{k=1}^{CI(\omega)} \dfrac{\omega'(p_k)[1 + \varphi(i-k)]}{\sum_{j=0}^{CI(\omega')-k}(1 + j\varphi)} + \sum\limits_{k=CI(\omega')+1}^{n} \dfrac{\omega'(p_k)[1 + \varphi(k - CI(\omega'))]}{\sum_{j=0}^{k-CI(\omega')}(1 + j\varphi)} & if\ i = CI(\omega'), \\ \sum\limits_{k=i}^{n} \dfrac{\omega'(p_k)[1 + \varphi(k-i)]}{\sum_{j=0}^{k-CI(\omega')}(1 + j\varphi)} & if\ i > CI(\omega'), \end{cases} \tag{17}$$

$$\omega(A) = \omega'(A),$$

where $\varphi = 0.2, 0.1, 0$ when the consistency between two basic support function is high, medium, poor respectively, A is the non-simple subset of P.

Step 7.2: Negative regulation.

When the consistency between two basic support functions is poor, the measure of uncertainty is used to compress the curve of truth-value of ω vertically until the entropy of ω approximately equals the entropy of ω', that is $|E(\omega) - E(\omega')| \leq \epsilon$. This process is called negative regulation and outlined in Algorithm 1.

Algorithm 1 The procedure of negative regulation.

Input: The initial fusion result ω' and basic support function ω after positive regulation
Output: The final fusion result ω

$\delta \leftarrow 1$
while $|E(\omega) - E(\omega')| \leq \epsilon$ **do**
  $I \leftarrow$ index of maximum truth-value of ω
  $k \leftarrow 1$
  for $k = I$ to $n - 1$ **do**
    if $\omega(p_k) > \omega(p_{k+1})$ **then**
      $\omega(p_k) = \omega(p_k) - \frac{\delta\omega(p_{k+1})(\omega(p_k)-\omega(p_{k+1}))}{\omega(p_k)+\omega(p_{k+1})}$
      $\omega(p_{k+1}) = \omega(p_{k+1}) + \frac{\delta\omega(p_{k+1})(\omega(p_k)-\omega(p_{k+1}))}{\omega(p_k)+\omega(p_{k+1})}$
    end if
  end for
  for $k = I; k > 1; k - -$ **do**
    if $\omega(p_k) > \omega(p_{k-1})$ **then**
      $\omega(p_k) = \omega(p_k) - \frac{\delta\omega(p_{k-1})(\omega(p_k)-\omega(p_{k-1}))}{\omega(p_k)+\omega(p_{k-1})}$
      $\omega(p_{k-1}) = \omega(p_{k-1}) + \frac{\delta\omega(p_{k-1})(\omega(p_k)-\omega(p_{k-1}))}{\omega(p_k)+\omega(p_{k-1})}$
    end if
  end for
  if $E(\omega) < E(\omega') - \epsilon$ **then**
    $\delta \leftarrow 1$
  end if
  if $E(\omega) > E(\omega') + \epsilon$ **then**
    $\delta \leftarrow \delta/2$
  end if
end while

Step 8: Calculate the final fusion result ω with method II.

Step 8.1: Positive regulation.

Denote $a = \lceil CI(\omega) \rceil$, $b = \lfloor CI(\omega') \rfloor$ for convenience, thus

$$\omega(p_i) = \begin{cases} \sum_{k=1}^{i} \frac{\omega'(p_k)[1+\varphi(i-k)]}{((\sum_{j=0}^{a-k}(1+j\varphi)) - \varphi)}, & if\ i < b, \\ \omega'(p_b) + \Gamma(a - CI(\omega')), & If\ i = b \wedge \omega'(p_b) \neq \omega'(p_a), \\ \omega'(p_a) + \Gamma(CI(\omega') - b), & If\ i = a \wedge \omega'(p_b) \neq \omega'(p_a), \\ \omega'(p_b) + \Gamma/2, & If\ i = b \wedge \omega'(p_b) = \omega'(p_a), \\ \omega'(p_a) + \Gamma/2, & If\ i = a \wedge \omega'(p_b) = \omega'(p_a), \\ \sum_{k=i}^{n} \frac{\omega'(p_k)[1+\varphi(k-i)]}{((\sum_{j=0}^{k-b}(1+j\varphi)) - \varphi)}, & if\ i > a, \end{cases} \tag{18}$$

$$\omega(A) = \omega'(A),$$

Table 2: The fusion process and result of example (1)

variables	values
λ_1	$\{(0.1, 0.4, 0.2, 0.1), (\lambda_1(p_1, p_2) = 0.1, \lambda_1(p_2, p_3) = 0.1)\}$
λ_2	$\{(0.1, 0.4, 0.2, 0.1), (\lambda_2(p_1, p_2) = 0.1, \lambda_2(p_2, p_3) = 0.1)\}$
$\overline{\lambda_1} = \overline{\lambda_2}$	0.2275
$NC(\lambda_1) = NC(\lambda_2)$	0.3269
$CI(\lambda_1) = CI(\lambda_2)$	2.4615
$\Delta G(\lambda_1, \lambda_2)$	0
ω'	$\{(0.11, 0.48, 0.22, 0.1), (\omega'(p_1, p_2) = 0.05, \omega'(p_2, p_3) = 0.04)\}$
$\overline{\omega'}$	0.2419
$NC(\omega')$	0.3711
$CI(\omega')$	2.3407
ω	$\{(0.0324, 0.5777, 0.2705, 0.1482), (\omega(p_1, p_2) = 0.05, \omega(p_2, p_3) = 0.04)\}$

where

$$\Gamma = \Gamma_1 + \Gamma_2, \quad \Gamma_1 = \sum_{k=1}^{b-1} \frac{\omega'(p_k)[1 + \varphi(b-k)]}{(\sum_{j=0}^{a-k}(1 + j\varphi)) - \varphi} + \sum_{a+1}^{n} \frac{\omega'(p_k)[1 + \varphi(k-a)]}{(\sum_{j=0}^{k-b}(1 + \varphi j)) - \varphi},$$

$$\Gamma_2 = \sum_{k=1}^{b-1} \frac{\omega'(p_k)[1 + \varphi(a-k) - \varphi]}{(\sum_{j=0}^{a-k}(1 + j\varphi)) - \varphi} + \sum_{a+1}^{n} \frac{\omega'(p_k)[1 + \varphi(k-a)]}{(\sum_{j=0}^{k-b}(1 + \varphi j)) - \varphi},$$

$\varphi = 0.2, 0.1, 0$ when the consistency between two basic support function is high, medium, poor respectively, A is the non-simple subset of P.

Step 8.2: Negative regulation.

It is same as Step 7.2.

6 Numerical examples

(1) Two basic support functions are $\lambda_1 = \{(0.1, 0.4, 0.2, 0.1), (\lambda_1(p_1, p_2) = 0.1, \lambda_1(p_2, p_3) = 0.1)\}$ and $\lambda_2 = \{(0.1, 0.4, 0.2, 0.1), (\lambda_2(p_1, p_2) = 0.1, \lambda_2(p_2, p_3) = 0.1)\}$. The weights of λ_1 and λ_2 are $\Omega_1 = \Omega_2 = 0.5$. The fusion processes and results are shown in Table 2. λ_1 and λ_2 are consistent, and they all mean that the 2nd proposition is most likely to be correct. So the fusing basic support function should reach the maximum truth-value at the index 2. The results are reasonable.

(2) Two basic support functions are $\lambda_1 = (0, 0.1, 0.2, 0.7)$ and $\lambda_2 = \{(0.1, 0.1, 0.1, 0.6), (\lambda_2(p_3, p_4) = 0.1)\}$. The weights of λ_1 and λ_2 are $\Omega_1 = \Omega_2 = 0.5$. The fusion result is $\omega = \{(0.0096, 0.0394, 0.1172, 0.8188), (\omega(p_3, p_4) = 0.015)\}$. λ_1 and λ_2 are not exactly the same, but $NC(\lambda_1) = 0.6 > 0.55$ and $NC(\lambda_2) = 0.5512 > 0.55$, so $CI(\lambda_1) = CI(\lambda_2) = 4$ and $\Delta G(\lambda_1, \lambda_2) = 0$. Similar to the previous example, the fusing basic support function should reach the maximum truth-value at the index 4. So the results are reasonable.

(3) Two basic support functions are $\lambda_1 = (0.7, 0.2, 0.1, 0)$ and $\lambda_2 = \{(0.1, 0.1, 0.1, 0.6), (\lambda_2(p_3, p_4) = 0.1)\}$. The weights of λ_1 and λ_2 are $\Omega_1 = \Omega_2 = 0.5$. The fusion results are shown in Table 4. λ_1 and λ_2 are totally conflicting and the fusion result is $\omega = \{(0.1333, 0.4737, 0.2680, 0.11), (\omega(p_3, p_4) = 0.015)\}$. The result shows that the 2nd proposition is most likely to be true, which is logical. It is reasonable that the uncertainty of the result is high.

Table 3: The fusion process and result of example (3)

variables	values
λ_1	$(0.7, 0.2, 0.1, 0)$
λ_2	$\{(0.1, 0.1, 0.1, 0.6), (\lambda_2(p_3, p_4) = 0.1)\}$
ω'	$\{(0.4, 0.15, 0.105, 0.33), (\omega'(p_3, p_4) = 0.015\}$
$E_d(\omega')$	1.5185
ω	$\{(0.1333, 0.4737, 0.2680, 0.11), (\omega(p_3, p_4) = 0.015)\}$
$E_d(\omega)$	1.4832

7 Conclusion

In order to better model the uncertain information of the characteristics of a subject, we proposed the generalized ordered propositions based on classical ordered propositions. The generalized ordered propositions extended the indeterminate part of a basic support function to all groups of propositions, not just the universal set of propositions. Then we considered the determinate part, indeterminate part, mean, degree of convexity and center of a basic support function in the situation of generalized ordered propositions. These properties can also be applied in classical ordered propositions. Additionally, we found the existing entropy of a basic support function does not apply when the indeterminate part is not zero. To address this shortage, we presented a new entropy based on belief entropy. This entropy measures not only the uncertainty of the determinate part but also indeterminate part of a basic support function. When the indeterminate part equals to zero, this entropy is degenerated into the existing entropy. Finally, we instructed the fusion method of basic support functions in generalized ordered propositions based on consistency and uncertainty. The experimental results show that the method is effective.

Acknowledgment

The authors greatly appreciate the reviews' suggestions and the editor's encouragement. The work is partially supported by National Natural Science Foundation of China (Grant Nos. 61573290, 61503237).

Conflict of interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Bibliography

[1] Abelln, J. (2017); Analyzing properties of deng entropy in the theory of evidence, *Chaos, Solitons & Fractals*, 95, 195–199, 2017.

[2] Azadi, M.; Jafarian, M.; Saen, R. F. ; Mirhedayatian , S. M. (2014); A new fuzzy dea model for evaluation of ef?ciency and effectiveness of suppliers in sustainable supply chain management context, *Computers & Operations Research*, 54 , 274–285, 2014.

[3] Bi, W.; Zhang, A.; Yuan, Y. (2017); Combination method of conflict evidences based on evidence similarity, *Journal of Systems Engineering and Electronics*, 28 (3), 503–513, 2017.

[4] Bian, T.; Deng, Y. (2018); Identifying influential nodes in complex networks: A node information dimension approach, *Chaos*, 28, 10.1063/1.5030894, 2018.

[5] Bian, T.; Zheng, H.; Yin, L.; Deng, Y. (2018); Failure mode and effects analysis based on Dnumbers and topsis, *Quality and Reliability Engineering International*, 34(4), 501-515, 2018.

[6] Bogdana, S.; Ioan, D.; Simona, D. (2015); On the ratio of fuzzy numbers - exact membership function computation and applications to decision making, *Technological and Economic Development of Economy*, doi:10.3846/20294913.2015.1093563, 21(5), 815–832, 2015.

[7] Cao, Z.; Lai, K.L.; Lin, C.T.; Chuang, C.H.; Chou, C.C.; Wang, S.J. (2018); Exploring resting-state EEG complexity before migraine attacks, *Cephalalgia*, 38(7), 1296–1306, 2018.

[8] Cao, Z.; Lin, C.T. (2018); Inherent Fuzzy Entropy for the Improvement of EEG Complexity Evaluation, *IEEE Transactions on Fuzzy Systems*, doi:10.1109/TFUZZ.2017.2666789, 26 (2), 1032–1035, 2018.

[9] Chao, X. R.; Kou, G.; Peng, Y. (2017); A Similarity Measure-based Optimization Model for Group Decision Making with Multiplicative and Fuzzy Preference Relations. *International Journal of Computers, Communications & Control*, 12(1), 2017.

[10] Chatterjee, K.; Zavadskas, E.K.; Tamoaitien, J.; Adhikary, K.; Kar, S. (2018); A hybrid MCDMtechnique for risk management in construction projects, *Symmetry*, doi:10.3390/sym10020046, 10 (2), 46, 2018.

[11] Chen, L.; Deng, X. (2018); A modfied method for evaluating sustainable transport solutions based on ahp and dempstercshafer evidence theory, *Applied Sciences*, doi:10.3390/app8040563, 8 (4), Article ID 563, 2018.

[12] Chin, K.S.; Fu, C. (2015); Weighted cautious conjunctive rule for belief functions combination, *Information Sciences*, 325, 70–86, 2015.

[13] Dahooie, J. H.; Zavadskas, E. K.; Abolhasani, M.; Vanaki ,A.; Turskis ,Z. (2018); A novel approach for evaluation of projects using an intervalvalued fuzzy additive ratio assessment ARAS method: A case study of oil and gas well drilling projects, *Symmetry*, doi:10.3390/sym10020045, 10 (2), 45, 2018.

[14] Dempster, A. P. (1967); Upper and lower probabilities induced by a multivalued mapping, *The annals of mathematical statistics*, 325–339, 1967.

[15] Deng, W.; Lu, X.; Deng, Y. (2018); Evidential Model Validation under Epistemic Uncertainty, *Mathematical Problems in Engineering*, doi:10.1155/2018/6789635, 2018.

[16] Deng, X. (2018); Analyzing the monotonicity of belief interval based uncertainty measures in belief function theory, *International Journal of Intelligent Systems (2018) Published online*, doi: 10.1002/int.21999, 2018.

[17] Deng, X.; Deng, Y. (2018); D-AHP method with different credibility of information, *Soft Computing (2018) Published online*, doi: 10.1007/s00500–017–2993–9, 2018.

[18] Deng, Y. (2016), Deng entropy, *Chaos, Solitons & Fractals*, 91, 549–553, 2016.

[19] Ding, W.; Lin, C.T.; Cao, Z. (2018); Deep Neuro-Cognitive Co-Evolution for Fuzzy Attribute Reduction by Quantum Leaping PSO With Nearest-Neighbor Memeplexes, *IEEE Transactions on Cybernetics*, 1–14, 2018.

[20] Ding, W.; Lin, C.T.; Cao, Z. (2018); shared Nearest Neighbor Quantum Game-based Attribute Reduction with Hierarchical Co-evolutionary Spark and Application in Consistent Segmentation of Neonatal Cerebral Cortex, *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[21] Ding, W.; Lin C. T.; Prasad, M.; Cao, Z.; Wang, J. (2018); A Layered-Coevolution-Based Attribute-Boosted Reduction Using Adaptive Quantum-Behavior PSO and Its Consistent Segmentation for Neonates Brain Tissue, *IEEE Transactions on Fuzzy Systems*, 26(3), 1177–1191, 2018.

[22] Duan, Y.; Cai, Y.; Wang, Z.; Deng, X. (2018); A novel network security risk assessment approach by combining subjective and objective weights under uncertainty, *Applied Sciences*, doi:10.3390/app8030428, 8 (3), Article ID 428, 2018.

[23] Dzitac, I., Filip, F. G., Manolescu, M. J. (2017); Fuzzy Logic Is Not Fuzzy: World-renowned Computer Scientist Lotfi A. Zadeh. *International Journal of Computers Communications & Control*, 12(6), 748–789, 2017.

[24] Fei, L.; Wang, H.; Chen, L.; Deng, Y. (2017); A new vector valued similarity measure for intuitionistic fuzzy sets based on OWA operators, *Iranian Journal of Fuzzy Systems*, accepted, 2017.

[25] Feller, W. (2008); *An introduction to probability theory and its applications*, Vol. 2, John Wiley & Sons, 2008.

[26] Fu, C.; Yang, J.B.; Yang, S.L. (2015); A group evidential reasoning approach based on expert reliability, *European Journal of Operational Research*, 246 (3), 886–893, 2015.

[27] Gong, Y.; Su, X.; Qian, H.; Yang, N. (2017); Research on fault diagnosis methods for the reactor coolant system of nuclear power plant based on D-S evidence theory, *Annals of Nuclear Energy*, DOI: 10.1016/j.anucene.2017.10.026, 2017.

[28] Han, Y.; Deng, Y. (2018); A hybrid intelligent model for Assessment of critical success factors in high risk emergency system, *Journal of Ambient Intelligence and Humanized Computing*, doi: 10.1007/s12652-018-0882-4, 2018.

[29] Han, Y.; Deng, Y. (2018); An enhanced fuzzy evidential DEMATEL method with its application to identify critical success factors, *Soft computing*, 10.1007/s00500-018-3311-x, 2018.

[30] Han, Y.; Deng, Y. (2018); An Evidential Fractal AHP target recognition method, *Defence Science Journal*, doi:10.14429/dsj.68.11737, Vol. 68, No. 4, July 2018, 367–373, 2018.

[31] Huynh, V.; Nakamori, Y.; Ho, T.; Murai, T. (2006); Multiple-attribute decision making under uncertainty: The evidential reasoning approach revisited, *IEEE Transaction on Systems Man and Cybernetics Part A-Systems and Humans*, 36 (4), 804–822, 2006.

[32] Jiang, W.; Wang, S. (2017); An uncertainty measure for interval-valued evidences, *International Journal of Computers Communications & Control*, 12 (5), 631–644, 2017.

[33] Jiang, W.; Wang, S.; Liu, X.; Zheng, H.; Wei, B. (2017); Evidence confict measure based on OWA operator in open world, *PloS one*, 12 (5), e0177828, 2017.

[34] Jiang, W.; Xie, C.; Zhuang, M.; Tang, Y. (2017); Failure mode and effects analysis based on a novel fuzzy evidential method, *Applied Soft Computing*, 57, 672–683, 2017.

[35] Jiang, W.; Yang, Y.; Luo, Y.; Qin, X. (2015); Determining basic probability assignment based on the improved similarity measures of generalized fuzzy numbers, *International Journal of Computers Communications & Control*, 10(3), 333–347, 2015.

[36] Kang, B.; Chhipi-Shrestha,G.; Deng, Y.; Hewage,K.; Sadiq, R. (2018); Stable strategies analysis based on the utility of Z-number in the evolutionary games, *Applied Mathematics & Computation*, 324, 202–217, 2018.

[37] Kang, B.; Deng, Y. (2018); Generating Z-number based on OWA weights usingmaximum entropy, *International Journal of Intelligent Systems*, INT21995, 2018.

[38] Li, C.; Mahadevan, S. (2016); Relative contributions of aleatory and epistemic uncertainty sources in time series prediction, *International Journal of Fatigue* , 82, 474–486, 2016.

[39] Li, C.; Mahadevan, S. (2016); Role of calibration, validation, and relevance in multi-level uncertainty integration, *Reliability Engineering & System Safety*, 148, 32–43, 2016.

[40] Li, M.; Zhang, Q.; Deng, Y. (2018); Evidential identification of influential nodes in network of networks, *Chaos, Solitons & Fractals*, 2018.

[41] Lin, C.T.; Chuang, C.H.; Cao, Z.; Singh, A.K.; Hung, C.S.; Yu, Y.H.; Nascimben, M.; and Liu, Y.T.; King, J.T.; Su, T.P.; Wang, S.J. (2017); Forehead EEG in Support of Future Feasible Personal Healthcare Solutions: Sleep Management, Headache Prevention, and Depression Treatment, *IEEE Access*, 5, 10612-10621, 2017.

[42] Liu D.; Yang K.; Tang H. (2000); A convex evidence theory model, *J Comput Res Dev*, 37(2), 175–181, 2000.

[43] Liu, D.; Zhu, Y.; Ni, N.; Liu, J. (2017); Ordered proposition fusion based on consistency and uncertainty measurements, *Science China Information Sciences*, 60 (8), 082103, 2017.

[44] Liu, T.; Deng, Y.; Chan, F. (2018); Evidential supplier selection based on DEMATEL and game theory, *International Journal of Fuzzy Systems*, 20 (4), 1321–1333, 2018.

[45] Liu, Y.T.; Pal, N.R.; Marathe, A.R.; Lin, C.T. (2018); Weighted fuzzy dempster-shafer framework for multimodal information integration, *IEEE Transactions on Fuzzy Systems*, doi:10.1109/TFUZZ.2017.2659764, 26 (1), 338–352, 2018.

[46] Meng, D.; Zhang, H.; Huang, T. (2016); A concurrent reliability optimization procedure in the earlier design phases of complex engineering systems under epistemic uncertainties, *Advances in Mechanical Engineering*, 8(10), 1687814016673976, 2016.

[47] Mo, H.; Deng, Y. (2018); A new MADA methodology based on D numbers, *International Journal of Fuzzy Systems*, 10.1007/s40815-018-0514-3, 2018.

[48] Mohsen, O.; Fereshteh, N. (2017); An extended vikor method based on entropy measure for the failure modes risk assessment - a case study of the geothermal power plant (gpp), *Safety Science*, 92, 160–172, 2017.

[49] Pawlak, Z.; Grzymala-Busse, J.; Slowinski, R.; Ziarko, W. (1995); Rough sets, *Communications of the ACM* 38(11), 88–95, 1995.

[50] Sabahi, F. (2016); A novel generalized belief structure comprising unprecisiated uncertainty applied to aphasia diagnosis, *Journal of Biomedical Informatics*, 62, 66–77, 2016.

[51] Shafer, G. (1976); *A mathematical theory of evidence*, Vol. 1, Princeton university press Princeton, 1976.

[52] Song, Y.; Wang, X.; Lei, L.; Xing, Y. (2015); Credibility decay model in temporal evidence combination, *Information Processing Letters*, 115 (2), 248–252, 2015.

[53] Xiao, F. (2014); Multi-sensor data fusion based on the belief divergence measure of evidences and the belief entropy, *Information Fusion*, 46 (2019), 23–32, 2019.

[54] Xiao, F. (2016); An intelligent complex event processing with D numbers under fuzzy environment, *Mathematical Problems in Engineering*, 2016 (1), 1–10, 2016.

[55] Xiao, F. (2017); A novel evidence theory and fuzzy preference approach-based multi-sensor data fusion technique for fault diagnosis, *Sensors*, 17(11), 2504, 2017.

[56] Xiao, F. (2017); An improved method for combining conficting evidences based on the similarity measure and belief function entropy, *International Journal of Fuzzy Systems*, doi: 10.1007/s40815–017–0436–5, 20(4), 1256–1266, 2017.

[57] Xiao, F. (2018). A Hybrid Fuzzy Soft Sets Decision Making Method in Medical Diagnosis, *IEEE Access* , 6, 25300–25312, 2018.

[58] Xiao, F. (2018); A novel multi-criteria decision making method for assessing health-care waste treatment technologies based on D numbers, *Engineering Applications of Artificial Intelligence* , 71, 216–225, 2018.

[59] Xu, H.; Deng, Y. (2018); Dependent evidence combination based on shearman coefficient and pearson coeficient, *IEEE Access*, 10.1109/ACCESS.2017.2783320, 2018.

[60] Xu, X.; Li, S.; Song, X.; Wen, C.; Xu, D. (2016); The optimal design of industrial alarm systems based on evidence theory, *Control Engineering Practice*, 46, 142–156, 2016.

[61] Yager, R.R. (2016); On viewing fuzzy measures as fuzzy subsets, *IEEE Transactions on Fuzzy Systems*, 24 (4), 811–818, 2016.

[62] Yager, R.R. (2016); Uncertainty modeling using fuzzy measures, *Knowledge-Based Systems*, 92, 1–8, 2016.

[63] Yager, R.R.; Elmore, P.; Petry, F. (2017); Soft likelihood functions in combining evidence, *Information Fusion*, 36, 185–190, 2017.

[64] Yin, L.; Deng, Y. (2018); Measuring transferring similarity via local information, *Physica A: Statistical Mechanics and its Applications*, 498, 102–115, 2018.

[65] Yin, L.; Deng, Y. (2018); Toward uncertainty of weighted networks: An entropy-based model, *Physica A: Statistical Mechanics and its Applications*, 2018.

[66] Yuan, R.; Meng, D.; Li, H. (2016); Multidisciplinary reliability design optimization using an enhanced saddlepoint approximation in the framework of sequential optimization and reliability analysis, *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 230(6), 570-578, 2016.

[67] Zadeh, L.A. (1996) ; Fuzzy sets, in: Fuzzy Sets, Fuzzy Logic, And Fuzzy Systems: Selected Papers by Lotfi A Zadeh, *World Scientisc*, 394–432, 1996.

[68] Zadeh, L.A. (2011); A note on z-numbers, *Information Sciences*, 181 (14), 2923–2932, 2011.

[69] Zhang, Q.; Li, M.; Deng, Y. (2018); Measure the structure similarity of nodes in complex networks based on relative entropy, *Physica A: Statistical Mechanics and its Applications*, 491, 749–763, 2018.

[70] Zhang, R.; Ashuri, B.; Deng, Y. (2017); A novel method for forecasting time series based on fuzzy logic and visibility graph, *Advances in Data Analysis and Classification*, 11.4, 759–783, 2017.

[71] Zhang, X.; Mahadevan, S. (2017); A game theoretic approach to network reliability assessment, *IEEE Transactions on Reliability*, 66 (3), 875–892, 2017.

[72] Zhang, X.; Mahadevan, S.; Deng, X. (2017); Reliability analysis with linguistic data: An evidential network approach, *Reliability Engineering & System Safety*, 162, 111–121, 2017.

[73] Zhang, X.; Mahadevan, S.; Sankararaman, S.; Goebel, K. (2018); Resilience-based network design under uncertainty, *Reliability Engineering & System Safety*, 169, 364-379, 2018.

[74] Zheng, X.; Deng, Y. (2018); Dependence assessment in human reliability analysis based on evidence credibility decay model and iowa operator, *Annals of Nuclear Energy*, 112, 673–684, 2018.

[75] Zheng, H.; Deng, Y. (2018); Evaluation method based on fuzzy relations between Dempster-Shafer belief structure, *International Journal of Intelligent Systems*, doi:10.1002/int.21956, 33(7), 1343–1363, 2018.

[76] Zhou, X.; Hu, Y.; Deng, Y.; Chan, Felix T.S.; Ishizaka, A. (2018); A DEMATEL-Based Completion Method for Incomplete Pairwise Comparison Matrix in AHP, *Annals of Operations Research*, doi:10.1007/s10479-018-2769-3, 2018.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 808-823, October 2018.

Electronic Throttle Valve Takagi-Sugeno Fuzzy Control Based on Nonlinear Unknown Input Observers

W. Gritli, H. Gharsallaoui, M. Benrejeb, P. Borne

Wafa Gritli*, Hajer Gharsallaoui, Mohamed Benrejeb
Université de Tunis El Manar,
Ecole Nationale d'Ingénieurs de Tunis,
LA.R.A Automatique,
BP 37, Le Belvédère, 1002 Tunis, Tunisie.
*Corresponding author: wafa.gritli@enit.rnu.tn,
hajer.gharsallaoui@gmail.com, mohamed.benrejeb@enit.rnu.tn

Pierre Borne
Ecole Centrale de Lille,
Cité scientifique CS20048,
59651 Villeneuve d'Ascq Cedex, France.
pierre.borne@centralelille.fr

Abstract: This paper deals with the synthesis of a new fuzzy controller applied to Electronic Throttle Valve (ETV) affected by an unknown input in order to enhance the rapidity and accuracy of trajectory tracking performance. Firstly, the Takagi-Sugeno (T-S) fuzzy model is employed to approximate this nonlinear system. Secondly, a novel Nonlinear Unknown Input Observer (NUIO)-based controller is designed by the use of the concept of Parallel Distributed Compensation (PDC). Then, based on Lyapunov method, asymptotic stability conditions of the error dynamics are given by solving Linear Matrix Inequalities (LMIs). Finally, the effectiveness of the proposed control strategy in terms of tracking trajectory and in the presence of perturbations is verified in comparison with a control strategy based on Unknown Input Observers (UIO) of the ETV described by a switched system for Pulse-Width-Modulated (PWM) reference signal.
Keywords: electronic throttle valve, switched system, Takagi-Sugeno fuzzy model, nonlinear unknown input observer, Lyapunov method.

1 Introduction

For further improvement of drivability, fuel economic system and emission performance of vehicles, the Electronic Throttle Control (ETC) systems is required to possess fast transient responses without overshoot and high static precision. Hence, obtaining a proper controller with the ability achieving the requirements is a very interesting topic for the ETC system. The challenging issue is that the control performance of the ETC system is adversely affected by the uncertain system physical parameters related to friction, return spring and gear backlash. To solve the parameter uncertainty problem in the controller design of the ETC system, a lot of efforts have been made from two aspects.

On the one hand, a linear model of the system has been used in several existing control design. In [10] and [23], a nonlinear control strategy has been proposed, consisting of a PID controller and a feedback compensator for friction and limp-home effects. A discrete-time sliding mode controller and observer are designed to realize robust tracking control of the valve system in [7] and [10]. In [20], the variable structure concept is used after the use of feedback backstepping techniques in the intermediate stages of ETV design. In [31], [13], [15] and [21], the emphasis is on the development of an adaptive control strategy, which is aimed to enhance the control strategy

robustness with respect to process parameter variations, caused by production deviations and variations of external conditions. In [14] and [18], a PID-type fuzzy logic controller has been proposed. In [16], fault tolerant control has been proposed for the ETV described by a switched discrete-time systems with input disturbances and actuator faults. In [6], the Smith-predictor control has been adapted for controlling the electronic throttle body over a delay-driven network.

On the other hand, among nonlinear control theory, the Takagi-Sugeno (T-S) fuzzy system [36] has been the most active branch of the fuzzy control field, [19], [20], [12], [39] and [13]. The stability and stabilisation of T-S systems has been the subject of many works either in the continuous case or in the discrete one, [2] and [3]. The Parallel Distribution Compensation (PDC) technology has been widely employed to design the fuzzy controller for T-S fuzzy systems in [26] and [28]. The problem of robust tracking control is investigated for a class of nonlinear systems approximated by a fuzzy T-S model in [11]. In [32], a fuzzy H_2 guaranteed cost sampled-data control problem for nonlinear time-varying delay systems is studied. An observer can be used for state estimation when the system states are unmeasurable [27]. The observer-based state feedback PDC controller can be employed to settle the unmeasured state condition as shown in [18] and [40]. In [19], an adaptive observer in the unknown input estimation form is proposed for a system with unmeasured premise variable. In [25], an adaptive observer is designed for the estimation of unmodeled dynamics in a T-S system. In [9], a T-S observer with parameter estimation was designed for a heat exchanger fouling detection problem. In [5], a joint state and parameter estimation observer was proposed for T-S systems whose matrices depend on unknown parameters. In [22], Nonlinear Unknown Input Fuzzy Observer (UIFO) has been used for fuzzy T-S systems to design the fuzzy fault tolerant control. In [7], an approach for Nonlinear Unknown Input Observer (NUIO) design for nonlinear systems has been proposed.

In this paper, a new control strategy based on NUIO is proposed for the ETV described by T-S fuzzy systems. NUIO is used to estimate the position of an automotive throttle valve. Then, an estimated-state feedback control law is developed via PDC. Based on Lyapunov method, asymptotic stability conditions of the error dynamics is given in LMIs to design the observer parameters. The proposed control strategy is then investigated and compared to switching control based on Unknown Input Observer (UIO). The following part of this article is organized as follows: In Section II, the topology of the ETV is presented and modeled. The proposed control strategy is explained and detailed in section III. Section IV is devoted to comparison and discussions, and finally, section VI ends the paper with a conclusion.

2 Electronic throttle valve topology

The studied electronic throttle control system includes an accelerator pedal, an Electronic Control Unit (ECU) and a throttle body, shown in figure 1.
The throttle body is composed of a DC motor, a reduction gear set, a valve plate, a position sensor and two nonlinear return springs [21].
The control signal, provided by the ECU, is the armature voltage of a DC-motor which is controlled by changing the PWM duty cycle. It generates the rotational torque to regulate the throttle plate position. Nomenclature used in the model is presented in the appendix A.

2.1 State space representation

The electrical part of the throttle body is modeled by (1) and the electromechanical part by (2).

$$u = L\dot{i} + Ri + e \tag{1}$$

$$e = K_v\dot{\theta}_m \tag{2}$$

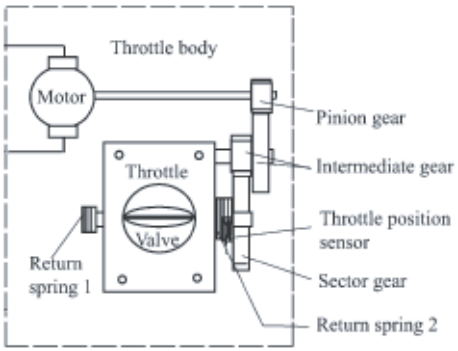


Figure 1: The electronic throttle body

$u(t)$ is the voltage, $i(t)$ the armature current and e the electromotive force, [24].
The electrical torque C_e is considered, in this study, such that

$$C_e = K_t i \tag{3}$$

By considering the stick-slip friction torque $T_f(\omega)$ and the nonlinear spring torque $T_{sp}(\theta)$, the mechanical part of the throttle body is modeled by (4), [24].

$$J_{tot}\dot{\omega} = -B_{tot}\omega - T_f(\omega) - T_{sp}(\theta) + C_e \tag{4}$$

There are many types of friction involved in the motion of throttle plate such as Coulomb, viscous, stribeck, rising static frictions and presliding displacement [20]. In this paper, the Coulomb friction model $T_f(\omega)$ is considered given by

$$T_f(\omega) = F_s \operatorname{sgn}(\omega) \tag{5}$$

where F_s is a positive constant parameter.
The typical feature of the ETV includes a stiff spring, used as a fail-safe mechanism, which forces the valve plate to return to the position slightly above the closed position when no power is applied. Moreover, the motion of the valve plate is limited between θ_{max} and θ_{min} angles. These limitations are realized by a highly stiff spring, ideally with infinite gain. The nonlinear spring expression is written such as

$$T_{sp}(\theta) = m_1(\theta - \theta_0) + D \operatorname{sgn}(\theta - \theta_0) \tag{6}$$

The gear ratio γ is given by (7).

$$\gamma = \frac{\theta_m}{\theta} = \frac{1}{K_{g1}K_{g2}} \tag{7}$$

such that: $K_{g1} = N_p/N_{\text{int}\,l}$ and $K_{g2} = N_{\text{int}\,s}/N_{\text{sec}\,t}$.

From equations (1), (2), (3) and (7) and by subtituting the expressions $T_{sp}(\theta)$ and $T_f(\omega)$ into (4), it comes the following ETV model

$$\begin{cases} \dot{\theta} = K_{g1}K_{g2}\omega \\ \dot{\omega} = -\frac{m_1}{J_{tot}}(\theta - \theta_0) - \frac{D}{J_{tot}}\operatorname{sgn}(\theta - \theta_0) - \frac{B_{tot}}{J_{tot}}\omega - \frac{F_s}{J_{tot}}\operatorname{sgn}(\omega) + \frac{K_t}{J_{tot}}i \\ L\dot{i} = -K_v\omega - Ri + u \end{cases} \tag{8}$$

2.2 System identification

By substituting C_e, T_f and T_{sp} into (4) and by neglecting the torque generated by the airflow C_a, the two nonlinearities $\text{sgn}(\theta-\theta_0)$ and $\text{sgn}(\omega)$ and the constant $\frac{m_1}{J_{tot}}\theta_0$, the ETV can be modelized by the following transfer function $H(s)$ [37].

$$H(s) = \frac{\theta(s)}{u(s)} = \frac{(180/\pi)K_e/\gamma}{LJ_{tot}s^3 + (RJ_{tot} + LB_{tot})s^2 + (RB_{tot} + K_vK_e + K_sL)s + K_sR} \tag{9}$$

with: $K_s = (180/\pi/\gamma)m_1$ and s the Laplace operator.

From equation (9), the ETV can be modelized by two linear models identified from the default position of the throttle plate for two values of the parameter K_s, [37]: a model representing the position of the plate above the position by default and the other the position of the plate below the position by default.

Switching between these two models of the ETV is equivalent to enabling and disabling the current model. Changing model and process structure raise problems such as detection model switching and maintain model tracking. Moreover, it is essential to consider the nonlinearities in the modeling phase, so that the behavior of the real system is described over a wide range of operation. It is, therefore, possible to consider modeling based on the concept of fuzzy logic. Indeed, in this case, a novel Takagi-Sugeno fuzzy model of the ETV is proposed which uses a base of locally linearised models.

2.3 T-S Fuzzy modeling

A reduced model of the ETV is firstly provided. To simplify the analysis, the motor armature inductance L will be assumed negligible then (1) and (2) can be rewritten as

$$-K_v\omega - Ri + u = 0 \tag{10}$$

Let $x(t) = [x_1(t)\ x_2(t)]^T$ and $x_{10} = \theta_0$, with

$$x_1 = \theta,\ x_2 = K_{g1}K_{g2}\omega \tag{11}$$

By substituting the value of i in the throttle valve dynamical system (8), the state space form can be simplified as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = a_{21}(x_1 - x_{10}) - \lambda\,\text{sgn}(x_1 - x_{10}) + (a_{22} - a_{23}\frac{a_{32}}{a_{33}})x_2 - \mu\,\text{sgn}(x_2) - \frac{a_{23}}{a_{33}}u \end{cases} \tag{12}$$

where the coefficients of the ETV model according to the physical parameters are given by: $a_{21} = m_1K_{g1}K_{g2}/J_{tot}$, $a_{22} = -B_{tot}/J_{tot}$, $a_{23} = K_tK_{g1}K_{g2}/J_{tot}$, $a_{32} = -K_v/K_{g1}K_{g2}$, $a_{33} = -R$, $\mu = F_sK_{g1}K_{g2}/J_{tot}$ and $\lambda := DK_{g1}K_{g2}/J_{tot}$.

The number of the local models depends on the nonlinear system complexity and the choice of the activation functions structure [1]. The polytope is obtained with $N = 2^r$ peaks where r is the number of premise variables considered for: $r = 2$.
Then, the ETV system can be transferred as the following T-S models.

$$\begin{array}{c} Rule1:\ IF\ x_1(t) < x_{10}\ AND\ x_2(t) <\ 0\ THEN \\ \begin{cases} \dot{x}(t) = A_1x(t) + B_1u(t) + D_1d(t) \\ y(t) = C_1x(t) \end{cases} \end{array} \tag{13}$$

$$Rule2: \ IF \ x_1(t) > \ x_{10} \ AND \ x_2(t) < \ 0 \ THEN \begin{cases} \dot{x}(t) = A_2x(t) + B_2u(t) + D_2d(t) \\ y(t) = C_2x(t) \end{cases} \tag{14}$$

$$Rule3: \ IF \ x_1(t) < \ x_{10} \ AND \ x_2(t) > \ 0 \ THEN \begin{cases} \dot{x}(t) = A_3x(t) + B_3u(t) + D_3d(t) \\ y(t) = C_3x(t) \end{cases} \tag{15}$$

$$Rule4: \ IF \ x_1(t) > \ x_{10} \ AND \ x_2(t) > \ 0 \ THEN \begin{cases} \dot{x}(t) = A_4x(t) + B_4u(t) + D_4d(t) \\ y(t) = C_4x(t) \end{cases} \tag{16}$$

with

$$A_1 = \begin{pmatrix} 0 & 1 \\ a_{21} + \lambda & a_{22} - a_{23}\frac{a_{32}}{a_{33}} + \mu \end{pmatrix}, \ A_2 = \begin{pmatrix} 0 & 1 \\ a_{21} - \lambda & a_{22} - a_{23}\frac{a_{32}}{a_{33}} + \mu \end{pmatrix}$$

$$A_3 = \begin{pmatrix} 0 & 1 \\ a_{21} + \lambda & a_{22} - a_{23}\frac{a_{32}}{a_{33}} - \mu \end{pmatrix}, \ A_4 = \begin{pmatrix} 0 & 1 \\ a_{21} - \lambda & a_{22} - a_{23}\frac{a_{32}}{a_{33}} - \mu \end{pmatrix}$$

$$B_1 = B_2 = B_3 = B_4 = \begin{pmatrix} 0 \\ -\frac{a_{23}}{a_{33}} \end{pmatrix}$$

$$C_1 = C_2 = C_3 = C_4 = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

$$D_1 = D_2 = D_3 = D_4 = 1e3. \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

The global fuzzy Takagi-Sugeno model is given, for $i \in \xi = \{1, ..., 4\}$, as follows

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{4} h_i(z(t))(A_ix(t) + B_iu(t) + D_id(t)) \\ y(t) = \sum_{i=1}^{4} h_i(z(t))C_ix(t) \end{cases}, \ i \in \xi \tag{17}$$

with

$$h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^{4} w_i(z(t))}, \ w_i(z(t)) = \prod_{j=1}^{2} M_j^i(z_j(t)) \tag{18}$$

M_j^i is the j^{th} fuzzy set of the i^{th} rule, $z_1(t), z_2(t)$ are the known premises variables and $M_j^i(z_j(t))$ the membership value of $z_j(t)$ in M_j^i.
Therefore, from (18) the following properties are satisfied

$$\sum_{i=1}^{4} h_i(z(t)) = 1, \ h_i(z(t)) \geqslant 0 \ \forall i \in \xi \tag{19}$$

The ETV control law should be designed to guarantee the tracking of the throttle movement θ for a desired reference signal with satisfactory transient performance and steady-state position error as well as robustness to nonlinearity parameter variations of friction, nonlinear spring and external disturbance.

3 Proposed T-S fuzzy control strategy

3.1 Structure of the proposed T-S fuzzy control strategy

For the T-S fuzzy model (17), the following control law based on the estimated state is proposed, as shown in figure 2.

$$u_{TS}(t) = -\sum_{i=1}^{N} h_i(z(t))(K_{TS,i}\hat{x}(t) + y^d(t)) \tag{20}$$

The main contribution of this article is to design an estimated feedback control law for the ETV described by T-S fuzzy model. A Nonlinear Unknown Input Observer (NUIO) is used to estimate the system state $\hat{x}(t)$. The control should maintain the system output closed to the desired trajectory $y^d(t)$ even in the presence of unknown input $d(t)$. A supervisor is implemented to calculate the weighting functions $h_i(z(t))$. The design of the fuzzy controller shares the same fuzzy sets as the fuzzy model and the same weights.

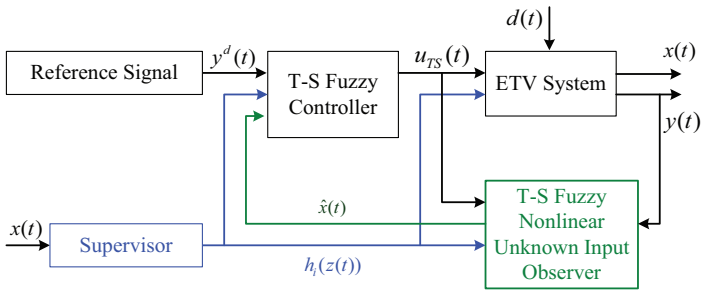


Figure 2: Proposed T-S Fuzzy controller structure for the ETV

$K_{TS,i} \in \mathbb{R}^{p\times n}$, $i \in \xi$, is the i^{th} feedback gain vector and $y^d(t) \in \mathbb{R}^p$ the reference input.
The purpose of the next sub-section is to design an estimated state-feedback control defined via PDC ensuring the stability of the closed-loop system.

3.2 Parallel distributed compensation

The design of the PDC fuzzy controller shares the same fuzzy sets as the fuzzy model and the same weights $w_i(z(t))$ in the premise parts [34]. The state feedback fuzzy controller is constructed via PDC as follows [35]

$$\begin{aligned} &\text{IF } z_1(t) \text{ is } M_1^i \text{ AND...AND } z_r(t) \text{ is } M_r^i \\ &\text{THEN } u(t) = -\sum_{i=1}^{N} h_i(z(t))K_{TS,i}x(t) \end{aligned} \tag{21}$$

The design of the fuzzy regulator is to determine the local feedback gains $K_{TS,i} \in \mathbb{R}^{p\times n}$.
By substituting (21) into (17), the closed loop model is written as

$$\begin{cases} \dot{x}(t) = \sum_{i=1}^{N}\sum_{i=j}^{N} h_i(z(t))h_j(z(t))Q_i x(t) \\ y(t) = \sum_{i=1}^{N} h_i(z(t))C_i x(t) \end{cases} \tag{22}$$

with: $Q_i = (A_i - B_i K_{TS,i})$.
Stability conditions for ensuring stability of (17) is derived using Lyapunov approach for linear continuous systems.

Theorem 1. *The equilibrium of the continuous fuzzy system described by (17) is asymptotically stable in the large if there exists a common positive definite matrix $X1$ such that, [34]*

$$A_i^T X1 + X1 A_i < 0 \tag{23}$$

for $i \in \xi$; that is, for all the subsystems.

By applying theorem 1 to (22), we can derive stability conditions.

Theorem 2. *The equilibrium of the continuous fuzzy control system described by (22) is asymptotically stable in the large if there exists a common positive definite matrix $X1$ such that, [34]*

$$Q_{ii}^T X1 + X1 Q_{ii} < 0 \tag{24}$$

$$\left(\frac{Q_{ij}+Q_{ji}}{2}\right)^T X1 + X1\left(\frac{Q_{ij}+Q_{ji}}{2}\right) \leqslant 0 \;\; i<j \tag{25}$$

for all i and j excepting the pairs (i,j) such that $h_i(z(t))h_j(z(t)) = 0,\ \forall t$.

The stability conditions of theorems 1 and 2 can be expressed as LMIs, [35].
By using: $X = X1^{-1}$ and: $M_i = K_{TS,i}X$, the satisfaction of LMIs conditions needs to find $X > 0$ and M_i such that, $\forall i \in \{1, ..., N\}$,

$$X A_i^T + A_i X - B_i M_i - M_i^T B_i^T < 0 \tag{26}$$

$$X(A_i^T + A_j^T) + (A_i + A_j)X - (B_i M_j + B_j M_i) - (B_i M_j + B_j M_i)^T < 0 \tag{27}$$

In practice, all states are not fully measurable; then, a nonlinear unknown input observer for T-S fuzzy models is proposed in order to implement the estimated state-feedback controller. The concept of PDC is employed to design the following NUIO structure in the next part.

3.3 NUIO design and stability analysis

The concept of PDC is employed to design NUIO for the T-S fuzzy model (17). The i^{th} observer rule is of the following form, [7]

$$\begin{array}{c} \text{IF } z_1(t) \text{ is } M_1^i \text{ AND...AND } z_r(t) \text{ is } M_r^i \\ \text{THEN } \dot{v}(t) = A_i v(t) + G_i u(t) + L_i y(t),\ i \in \xi \end{array} \tag{28}$$

The overall fuzzy observer is given as

$$\begin{cases} \dot{v}(t) = \sum\limits_{i=1}^{N} h_i(z(t))(N_i v(t) + G_i u(t) + L_i y(t)) \\ \hat{x}(t) = v(t) - Ey(t) \end{cases}, \; i \in \xi \tag{29}$$

where N_i, G_i and L_i, $i \in \xi$ and E are unknown matrices to be designed, [7].
Let's define the error $e(t) = \hat{x}(t) - x(t)$, it follows from (17) and (29) that

$$\begin{aligned}\dot{e}(t) &= \sum_{i=1}^{N} h_i(z(t))N_i e(t) \\ &+ \sum_{i=1}^{N} h_i(z(t))\{N_i + K_i C_i - (I + EC_i)A_i\}x(t) \\ &+ \sum_{i=1}^{N} h_i(z(t))\{G_i - (I + EC_i)B_i\}u(t) \\ &- \sum_{i=1}^{N} h_i(z(t))(I + EC_i)D_i d(t)\end{aligned} \tag{30}$$

with

$$K_i = L_i + N_i E \tag{31}$$

A sufficient condition, for the observer given by (29) to be an NUIO, is given as in the following theorem.

Theorem 3. *For the observer given by (29), if K_i, $i \in \xi$ and E are chosen such that*

$$\begin{gathered}N_i = (I + EC_i)A_i - K_i C_i \\ G_i = (I + EC_i)B_i \\ L_i = K_i - N_i E \\ EC_i D_i = -D_i\end{gathered} \tag{32}$$

and if a positive definite symmetric matrix $X2$ can be found to satisfy the following inequalities

$$N_i^T X2 + X2 N_i < 0, \ i \in \xi \tag{33}$$

then the error dynamics given by (30) is asymptotically stable at the origin. Hence the observer given by (29) is an NUIO, that is, $e(t)$ goes to zero asymptotically and is invariant with respect to the unknown inputs $d(t)$, [7].

Proof. The proof is given in the appendix B.

Theorem 4. *For the observer given by (29), if there exist matrices $\bar{K}_i$, $i \in \xi$, a matrix $\bar{Y}$ and a positive definite symmetric matrix X such that the following LMIs are satisfied*

$$\begin{gathered}[(I + UC_i)A_i]^T X2 + X(I + UC_i)A_i \\ +(VC_iA_i)^T\bar{Y}^T + \bar{Y}(VC_iA_i) - C_i^T\bar{K}_i^T - \bar{K}_i C_i < 0\end{gathered} \tag{34}$$

with: $U = -D_i(C_iD_{T1})^+$ and: $V = I - C_iD_{T1}(C_iD_{T1})^+$; then by letting $K_i = X2^{-1}\bar{K}_i$ and $Y = X2^{-1}\bar{Y}$ and computing the observer gains using (39) and (32), the error dynamics given by (30) is asymptotically stable at the origin. Hence, the observer given by (29) is a NUIO, that is, $e(t)$ goes to zero asymptotically and is invariant with respect to the unknown inputs $d(t)$, [7].

Proof. The proof is given in the appendix C.

In the next section, in order to evaluate the proposed control laws performance against the nonlinearities such as friction and limp home spring of ETV, the position is examined for a PWM signal.

4 Results and discussion

In order to test the position of the ETV for a PWM reference signal provided by the T-S fuzzy control, a comparative study is performed. The proposed control strategy is then investigated and compared to switching control based on UIO proposed in our previous work, [16] and [17].

To illustrate the effectiveness of the proposed control strategy, a T-S fuzzy model of the ETV, is firstly, provided. The considered model is given by (12) with the state region $0rad < x_1(t) < \pi/2rad$ and $-80rad/s < x_2(t) < 80rad/s$.
The considered ETV system parameters, for the numerical simulations, are presented in table 4, as given in [20].

Table 1: Parameter values for simplified model

Parameters	**Values**
a_{21}	$1/18$
a_{22}	$-1.6801e3$
a_{23}	-32.9820
a_{32}	-0.0245
a_{33}	-1.0980
μ	$4.7438e2$
λ	$2.1073e3$

Solutions satisfying stability conditions under LMIs of the theorem 2 are found for symmetric definite positive matrix $X1$ given by (35).

$$X1 = \begin{pmatrix} 32.7434 & -8.9104 \\ -8.9104 & 32.7434 \end{pmatrix} \tag{35}$$

Then, the feedback gains are given by

$$K_{TS,1} = \begin{bmatrix} 0.7505 & 0.5061 \end{bmatrix}, K_{TS,2} = \begin{bmatrix} 0.7505 & -0.5716 \end{bmatrix}$$
$$K_{TS,3} = \begin{bmatrix} -1.6091 & 0.5061 \end{bmatrix}, K_{TS,4} = \begin{bmatrix} -0.5050 & -0.2716 \end{bmatrix}$$

Solutions satisfying stability conditions under LMIs of the theorem 3 and 4 are found for symmetric definite positive matrix $X2$ given by (36). Then, the rest parameters of NUIO are calculated.

$$X2 = \begin{pmatrix} 0.8820 & 0 \\ 0 & 0.0002 \end{pmatrix} \tag{36}$$

The throttle plate follows a PWM signal of amplitude ranged from 0 to 1.5708 rad and frequency $f = 0.2Hz$ with a white Gaussian noise $d(t)$ given by figure 3.

The system responses obtained from initial conditions: $x_0(t) = [0.1\ 0]^T$ are shown in figures 4-6.

Figure 4 shows the evolution of the throttle plate angle in the presence of unknown input $d(t)$ with the switched system and the T-S fuzzy system. For the switched system, the throttle plate tracks the reference signal with settling time equal to $2.5s$ and fluctuations due to the noise. Whereas for the same reference input and by using the T-S fuzzy system, the throttle plate tracks the reference signal with settling time equal to $0.25s$, rotor angular velocity average value: $\pm 1.0295 rad/s$, figure 7, and ECU output voltage ranging from: $-2.7\ V$ to $1.6\ V$, figure 6.

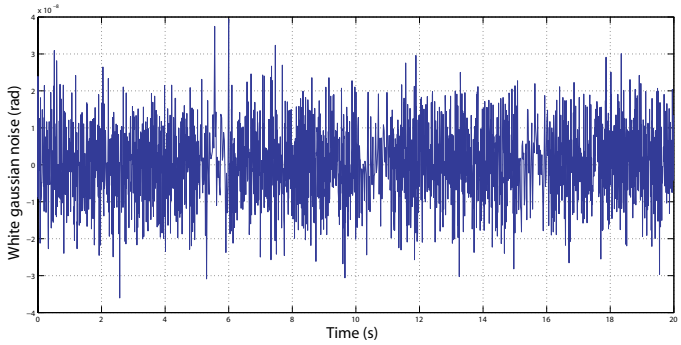


Figure 3: White gaussian noise

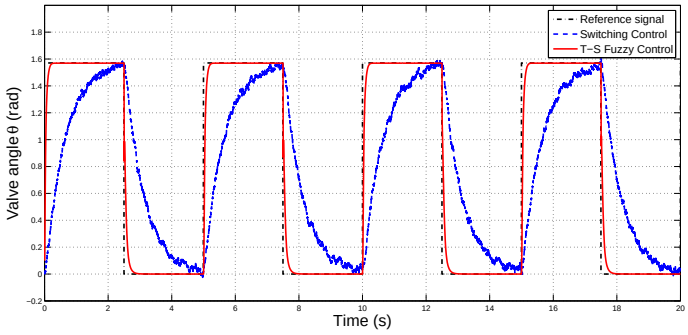


Figure 4: Valve angle position

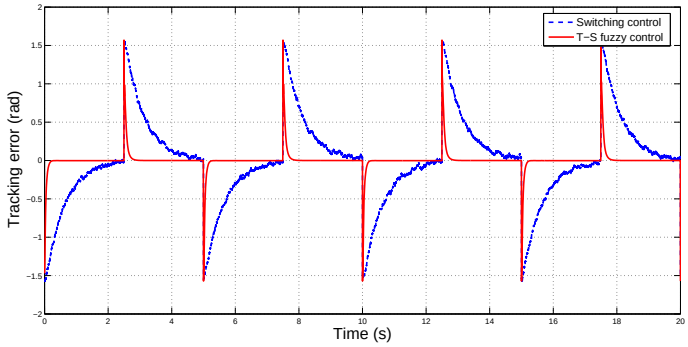


Figure 5: Tracking error

It can be observed, from figure 5, that the error between the throttle plate angle and its steady-state value has been greatly reduced, in terms of average value, to: $\pm 40.10^{-3} rad$ using the proposed strategy against: $\pm 40.10^{-2} rad$ with the switched system technique.

From figure 8, simulation results using the proposed T-S fuzzy control strategy show that the controller yields good tracking performance for a step perturbation amplitude of $0.2 rad$ at $t = 6s$ with a small error between the throttle plate angle and its steady state.
From the simulation results, we consider that the performances of the proposed approach for T-S fuzzy system control are satisfactory and allow normal functioning of the system in spite of the fast acceleration and deceleration process even in the presence of an unknown input $d(t)$ and perturbation. Indeed, the electronic throttle control system has a fast transient response

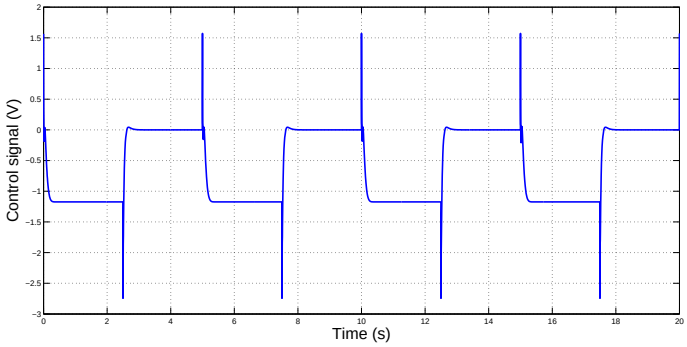


Figure 6: Control signal obtained by T-S fuzzy control

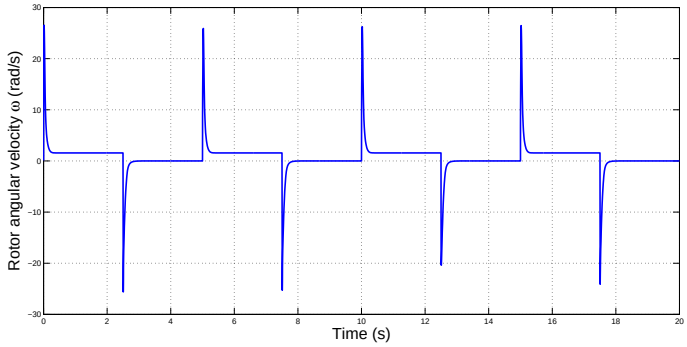


Figure 7: Rotor angular velocity obtained by T-S fuzzy control

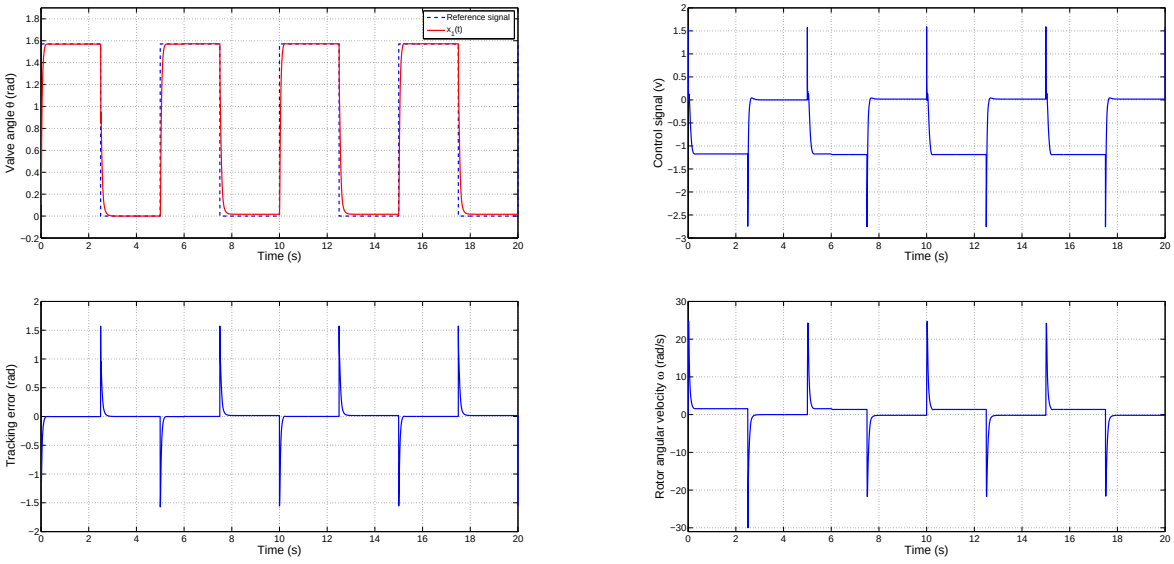


Figure 8: Simulation results with step perturbation

without overshoot and high static precision.

The results of this paper may inspire further research interest. Certainly, actuator and sensor faults occur, a Fault Tolerant Controller (FTC) based on T-S observer strategy would be

designed. It should be noted also that the control strategy would be given from the view of discontinuous systems. Then, experimental validation would be performed to illustrate the performance of the presented throttle control for tracking a reference position.

5 Conclusion

The difficulty in controlling the studied Electronic Throttle Valve (ETV) mainly lies in the nonlinearities related to the friction, the return spring and the gear mechanism. Therefore, a new control strategy based on Nonlinear Unknown Input Observer (NUIO) has been proposed for the ETV. Firstly, a Takagi-Sugeno fuzzy model for the ETV with unknown input has been constructed. Then, NUIO, designed via the Parallel Distributed Compensation (PDC), are used to estimate the unmeasurable system status. Lyapunov theory and Linear Matrix Inequalities (LMIs) have been used for ensuring the stability of the error system. The proposed approach has been achieved in comparison with Unknown Input Observer (UIO)-based control for the ETV described by switched system. UIO is designed and formulated in terms of Lyapunov theory and LMIs technique in order to maintain asymptotic stability. The simulation results have indicated the usefulness of the proposed approach for T-S fuzzy disturbed system control in terms of tracking a desired valve angle position and pertubation rejection with a fast transient response and high static precision for Pulse-Width-Modulated (PWM) reference signal.

Bibliography

[1] Ben Hamouda, L.; Ayadi, M.; Langlois, N. (2016); Fuzzy Fault Tolerant Predictive Control for a Diesel Engine Air Path, *International Journal of Control, Automation and Systems*, 14, 443-451, 2016.

[2] Benrejeb, M.; Soudani, D.; Sakly, A.; Borne, P. (2006); New Discrete Tanaka Sugeno Kang Fuzzy Systems Characterization and Stability Domain, *International Journal of Computers Communications & Control*, 1(4), 9-19, 2006.

[3] Benrejeb, M. (2010); Stability Study of Two Level Hierarchical Nonlinear Systems Plenary lecture, *IFAC Proceedings Volumes*, 43(8), 30-41, 2010.

[4] Bernardo, M.; Gaeta, A.; Montanaro, U.; Santini, S. (2010); Synthesis and experimental validation of the novel LQ-NEMCSI adaptive strategy on an electronic throttle valve, *IEEE Transactions on Control Systems Technology*, 18(6), 1325-1337, 2010.

[5] Bezzaoucha, S.; Marx, B.; Maquin, D.; Ragot, J. (2013); State and parameter estimation for nonlinear systems: a Takagi-Sugeno approach, *In American Control Conference*, Washington, 2013.

[6] Caruntu, C.F.; Vargas, A.N.; Acho, L.; Pujol, G. (2018); Adaptive-Smith Predictor for Controlling an Automotive Electronic Throttle over Network, *International Journal of Computers Communications & Control*, 13(2), 151-161, 2018.

[7] Chen, W.; Saif, M. (2007); Design of a TS Based Fuzzy Nonlinear Unknown Input Observer with Fault Diagnosis Applications, *American Control Conference*, New York, 2007.

[8] Chen, J.; Patton, R. J.; Zhang, H. Y. (1996); Design of unknown input observers and robust fault detection filters, *International Journal of Control*, 63(1), 85-105, 1996.

[9] Delmotte, F.; Dambrine, M.; Delrot, S.; Lalot, S. (2013); Fouling detection in a heat exchanger: A polynomial fuzzy observer approach, *Control Engineering Practice*, 21, 1386-1395, 2013.

[10] Deur, J.; Pavkovic, D.; Peric, N.; Jansz, M.; Hrovat, D. (2004); An electronic throttle control strategy including compensation of friction and limphome effects, *IEEE Transanctions on Industry Applications*, 40(3), 821-834, 2004.

[11] Du, Z.-B.; Lin, T.-C.; Zhao, T.-B. (2015); Fuzzy Robust Tracking Control for Uncertain Nonlinear Time-Delay System, *International Journal of Computers Communications and Control*, 10(6), 812-824, 2015.

[12] Dragos, C.-A.; Precup, R.-E.; Tomescu, M.L.; Preitl, S.; Petriu, E.M.; Radac, M.-B. (2013); An Approach to Fuzzy Modeling of Electromagnetic Actuated Clutch Systems, *International Journal of Computers Communications and Control*, 8(3), 395-406, 2013.

[13] Dzitac, I.; Filip, F.G.; Manolescu, M.J. (2017); Fuzzy Logic Is Not Fuzzy: World-renowned Computer Scientist Lotfi A. Zadeh, *International Journal of Computers Communications & Control*, 12(6), 748-789, 2017.

[14] Gritli, W.; Gharsallaoui, H.; Benrejeb, M. (2017); A New Methodology for Tuning PID-Type Fuzzy Logic Controllers Scaling Factors Using Genetic Algorithm of a Discrete-Time System, *Modern Fuzzy Control Systems and Its Applications, InTech*, 5, 89-103, 2017.

[15] Gritli, W.; Gharsallaoui, H.; Benrejeb, M. (2016); PID-type Fuzzy Scaling Factors Tuning Using Genetic Algorithm and Simulink Design Optimization for Electronic Throttle Valve, 3^{rd} *International Conference on Control, Decision and Information Technologies CoDIT*, Malta, 2016.

[16] Gritli, W.; Gharsallaoui, H.; Benrejeb, M. (2017); Fault Tolerant Control Based on PID-type Fuzzy Logic Controller for Switched Discrete-time Systems: An Electronic Throttle Valve Application, *Advances in Science, Technology and Engineering Systems Journal*, 2(6), 186-193, 2017.

[17] Gritli, W.; Gharsallaoui, H.; Benrejeb, M. (2017); Fault Detection Based on Unknown Input Observers for Switched Discrete-Time Systems, *International Conference on Advanced Systems and Electric Technologies IC-ASET*, Hammamet, 2017.

[18] He, S.P.; Liu, F. (2012); Finite-time H^{∞} fuzzy control of nonlinear jump systems with time delays via dynamic observer-based state feedback, *IEEE Transactions on Fuzzy Systems*, 20(4), 605-614, 2012.

[19] Ichalal, D.; Marx, B.; Ragot, J.; Maquin, D. (2009); State and unknown input estimation for nonlinear systems described by Takagi-Sugeno models with unmeasurable premise variables, *In 17th Mediterranean Conference on Control and Automation*, Thessaloniki, 2009.

[20] Jiao, X.; Zhang, J.; Shen, T. (2008); Variable-Structure Control of Electronic Throttle Valve, *IEEE Transactions on Industrial Electronics*, 55(11), 2008.

[21] Jiao, X.; Zhang, J.; Shen, T. (2014); An Adaptive Servo Control Strategy for Automotive Electronic Throttle and Experimental Validation, *IEEE Transactions on Industrial Electronics*, 61(11), 2014.

[22] Kamal, E.; Aitouche, A.; Ghorbani, R.; Bayart, M. (2012); Unknown Input Observer with Fuzzy Fault Tolerant Control for Wind Energy System, *8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Mexico City, 2012.

[23] Kitahara, A; Sato, A.; Hoshino, M.; Kurihara, N.; Shin, S. (1996); LQG based electronic throttle control with a two degree of freedom structurev, *Proceedings 35th IEEE Conference Decision Control*, 6(3), 1785-1789, Kobe, 1996.

[24] Lebbal, M.; Chafouk, H.; Hoblos, G.; Lefebvre, D. (2007); Modelling and Identification of Non-Linear Systems by a Multimodel Approach: Application to a Throttle Valve, *International Journal Information and Systems Science*, 3, 67-87, 2007.

[25] Lendek, Z.; Guerra, T.M.; De Schutter, B. (2010); Stability analysis and nonlinear observer design using Takagi-Sugeno fuzzy models, *Springer*, 2010.

[26] Li, H.Y.; Gao, Y.B.; Wu, L.G. Lam, H.K. (2015); Fault detection for T-S fuzzy time-delay systems: Delta operator and input-output methods, *IEEE Transactions on Cybernetics*, 45(2), 229-241, 2015.

[27] Li, H.Y.; Shi, P.; Yao, D.Y.; Wu, L.G. (2016); Observer based adaptive sliding mode control for nonlinear Markovian jump systems, *Automatica*, 64, 133-142, 2016.

[28] Manai, Y.; Benrejeb, M. (2011); New Condition of Stabilisation for Continuous Takagi-Sugeno Fuzzy System based on Fuzzy Lyapunov Function, *International Journal of Control and Automation*, 4(3), 2011.

[29] Nakano, K.; Sawut, U.; Higuchi, K.; Okajima, Y. (2006); Modelling and observer-based sliding-mode control of electronic throttle systems, *ECTI Transactions on Electrical Engineering, Electronics, and Communications*, 4(1), 22-28, 2006.

[30] Ozguner, U.; Hong, S.; Pan, Y. (2001); Discrete-time sliding mode control of electronic throttle valve, *Proceedings 40th IEEE Conference Decision Control*, 1819-1824, Orlando, FL, 2001.

[31] Pavkovic, D.; Deur, J.; Jansz. M.; Peric, N. (2006); Adaptive control of automotive electronic throttle, *Control Engineering Practice*, 14(2), 121-136, 2006.

[32] Qu, Z.-F.; Du, Z.-B. (2016); Fuzzy H_2 Guaranteed Cost Sampled-Data Control of Nonlinear Time-Varying Delay Systems, *International Journal of Computers Communications & Control*, 11(5), 708-719, 2016.

[33] Su, X.J.; Shi, P.; Wu, L.G.; Song, Y.-D. (2013); A novel control design on discrete-time Takagi-Sugeno fuzzy systems with time-varying delays, *IEEE Transactions on Fuzzy Systems*, 21(4), 655-671, 2013.

[34] Tanaka, K; Sugeno, M. (1992); Stability Analysis and Design of Fuzzy Control Systems, *Fuzzy Sets and Systems*, 45(2), 135-156, 1992.

[35] Tanaka, K.; Ikeda, T.; Wang, H.O. (1998); Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs, *IEEE Transanctions on Fuzzy Systems*, 6(2), 1-16, 1998.

[36] Takagi, T.; Sugeno, M. (1985); Fuzzy identification of systems and its application to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics*, 15, 116-132, 1985.

[37] Yang, C. (2004); Model-based analysis and tuning of electronic throttle controllers, *Visteon Corporation*, SAE 2004 World Congress & Exhibition, 63-67, 2004.

[38] Yuan, X.; Wang, Y.; Sun, W.; Wu, L. (2010); RBF networks-based adaptive inverse model control system for electronic throttle, *IEEE Transactions on Control Systems Technology*, 18(3), 750-756, 2010.

[39] Zadeh, L.A.; Tufis, D.; Filip, F.G.; Dzitac, I.; (2008); From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence, *Exploratory Workshop on NL-Computation*, Baile Felix, Oradea, Romania, 2008.

[40] Zhang, J.H.; Shi, P.; Qiu, J.Q.; Nguang, S.K. (2015); A novel observer-based output feedback controller design for discrete-time fuzzy systems, *IEEE Transactions on Fuzzy Systems*, 23(1), 223-229, 2015.

[41] Zhao, X.D.; Zhang, L.X.; Shi, P.; Karimi, H.R. (2014); Novel stability criteria for T-S fuzzy systems, *IEEE Transactions on Fuzzy Systems*, 22(2), 313-323, 2014.

Appendix A. Nomenclature

J_{tot}	Total moment of inertia	$Kg.m^2$
B_{tot}	Total damping constant	$N.m/rad$
N_p	Tooth number of pinion gear	-
$N_{\text{int}\,l}$	Tooth number of large intermediate gear	-
$N_{\text{int}\,s}$	Tooth number of small intermediate gear	-
$N_{\text{sec}\,t}$	Tooth number of sector gear	-
L	Motor inductance	H
R	Motor resistance	Ω
K_t	Motor torque constant	$N.m/A$
K_v	Motor back EMF constant	$V.s/rad$
θ_0	Spring default position	rad
θ_{min}	Spring min position	rad
θ_{max}	Spring max position	rad
θ	Valve plate position	rad
θ_m	Motor rotational position	rad
ω	Rotor angular velocity	rad/s
D	Spring offset	$N.m$
m_1	Spring gain	$N.m/rad$

Appendix B. Proof of theorem 3

Proof: Using (32), it follows from (30) that

$$\dot{e}(t) = \sum_{i=1}^{N} h_i(z(t)) N_i e(t) \tag{37}$$

Using (36), it is now quite standard to prove the stability property.
It is easy to see that (32) implies that

$$EC_i(D_1, \cdots, D_N) = -(D_1, \cdots, D_N) \tag{38}$$

Assumption: $rank(EC_i(D_1, \cdots, D_N)) = rank((D_1, \cdots, D_N))$.
Under assumption, there exists a nonsingular matrix T_D such that $(D_1, \cdots, D_N)T_D = (D_{T1}\ 0)$, where D_{T1} is of full column rank. This implies that $C_i D_{T1}$ is of full column rank. (38) requires all the possible solutions for E to have the following form

$$E = -D_i(C_i D_{T1})^+ + Y(I - C_i D_{T1}(C_i D_{T1})^+) \tag{39}$$

where Y can be any compatible matrix and $X^+ = (X^T X)^{-1} X^T$.
In order to provide an efficient design method, we reformulate the sufficient conditions given by (32) and (36) as LMIs.

Appendix C. Proof of theorem 4

Proof: Using (39), it is easy to show that LMI based conditions given by (34) are equivalent to those conditions required in the theorem 3. The theorem is therefore proved.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 824-836, October 2018.

Optimization of the Latency in Networks SDN

G.A. Keupondjo Satchou, N.G. Anoh, T. N'Takpé, S. Oumtanaga

Gilles Armel Satchou Keupondjo*
Laboratoire de Recherche en Informatique et Télécommunications (LARIT)
Institut National Polytechnique Félix Houphouët Boigny de Yamoussoukro, Côte d'Ivoire
*Corresponding author: armel.keupondjo@inphb.ci

Nogbou Georges Anoh
1. Laboratoire de Recherche en Informatique et Télécommunications (LARIT)
Institut National Polytechnique Félix Houphouët Boigny de Yamoussoukro, Côte d'Ivoire
2. Unité de Recherche et d'Expertise Numérique (UREN)
Université virtuelle, Côte d'Ivoire
georges.anoh@uvci.edu.ci

Tchimou N'Takpé
1. Laboratoire de Recherche en Informatique et Télécommunications (LARIT)
Institut National Polytechnique Félix Houphouët Boigny de Yamoussoukro, Côte d'Ivoire
2. Laboratoire de Mathématique et Informatique (LMI)
Université Nangui Abrogoua , Côte d'Ivoire
tchimou.ntakpe@gmail.com

Souleymane Oumtanaga
Laboratoire de Recherche en Informatique et Télécommunications (LARIT)
Institut National Polytechnique Félix Houphouët Boigny de Yamoussoukro, Côte d'Ivoire
oumtana@gmail.com

Abstract: Unlike traditional networks, software-defined networking (SDN) are characterized by a physical separation of the control and the transfer plan. Thus, a centralized controller communicates the functions of control plan to each device via the OpenFlow Protocol whenever he is asked or that it deems appropriate. This impact strongly the latency time which is important for new services or multimedia applications. In order to optimize the time of transmission in network data with SDN, the proactive approach based on the algorithm back - pressure is usually offered. However, we note that the proactive approach while reducing strongly this time, does not account settings such as the failure of a node part of the way to transfer or the breaking of a bond that greatly increases the latency time. In this document, we will propose a joint routing approach based on proactive and reactive routing. This in order to optimize the routing functions by simply placing the traffic where capacity allows, in order to avoid congestion of highly stressed parts of the network taking into account the failures and significantly reduce the time latency. Simulation results show that our proposal allows a considerable reduction of latency even when there are failures in the network.
Keywords: Software-defined Networking (SDN), routing, multi-path, latency.

1 Introduction

Conceptually, a router is characterized by a control plan, and a transfer plan. As a result, the routers are responsible for the supervision of the topology of the network and the transfer of packages using static, or dynamic routing protocols. What is often at the origin of a load of significant calculation. In networks SDN (Software-Defined Networking) this charge is the responsibility of one or several controllers [2, 9, 12, 14].

As a result, a centralized controller communicates control plane functions to each device via the OpenFlow protocol [17] whenever it is requested or deemed appropriate [4, 16]. This has a significant impact on latency, which is important for multimedia applications. In order to optimize the transmission delay in data networks with SDN, two routing approaches are generally proposed; such as the proactive approach and the reactive approach. However, with the reactive approach, resource utilization is optimal and this approach is resilient to outages; which is not the case for the proactive approach which greatly reduces the transmission delay. However, the proactive approach does not take into account parameters such as the failure of a node in the transfer path or the break of a link. But these failures greatly increase the latency. Despite the incremental deployment of new routing solutions adapted to modern uses of the Internet [5] facilitated by SDN networks. It is in this context that we propose in this article a mixed routing approach that will reduce latency in case of network failures. This makes it possible to optimize the routing functions by simply placing the traffic where the capacity allows it, in order to avoid the congestions of some parts of the network that are heavily loaded, taking into account faults and considerably reducing the latency.

2 Related works

The algorithm back - pressure [8, 11, 13] is a well-known optimal flow algorithm. It refers to an algorithm to dynamically route traffic over a network to multiple bonds using gradients of congestion.

Its implementation requires that each node maintains a separate queue for each stream on the network, and a single queue is served at a time. As traffic in the data network is usually very large, maintaining the data structure of the queues at each node becomes complex. In [1], the authors propose an approach allowing each node on the network to maintain a single queue that implements the FIFO (First In First Out) approach for each outbound links, instead of keeping a separate queue each stream on the network. They also propose an algorithm that force back - pressure, to use the minimum amount of network resources while maintaining maximum throughput. However, for low flow rates arrived, delays will be much higher. That's why in [15], the authors propose to combine the algorithm of routing of shortest path to back - pressure, in order to maintain several queues at each intermediate node for each stream and ensure that packets of a stream will reach the respective after crossing more than destination n nodes. This allows to optimize the delay of package among the variants of back - pressure presented. But it overload due to queues of more and larger nodes. In [10] , the authors propose a variant of the algorithm of back - pressure based on the LIFO (Last In First Out) approach for minimization of the time. But the main limitation of this method is that some early packages get trapped in the LIFO buffer indefinitely.

With the emergence of SDN and its use of more and more in networks of data in [6], the authors rely on the centralized management of the network with SDN and offer improved routing back - pressure. Associated with techniques of shortest paths to the destination routing to optimize transmission times and delays results in looping of packets on the network. This proposed method shows a significant gain in performance in terms of reduction of delays in packets, length of queue average, average length of jump while retaining ownership of optimality of the routing algorithm flow back - pressure base.

However, suppose that a node reboots into the network. It has more rules of transfer and so cannot make a decision of transfer if a flow happens at that moment as long as the controller will again provide transfer rules. This can significantly impact the time of transmission. And also one of the problems in data with SDN network is control of the size of the table of rules of transfers.

The question that arises is how to optimize latency in the face of the problem of power outages on the network.

3 The SDN paradigm

In networks SDN, one or several controllers have supported the calculation of roads, thereby routers are reduced to simple devices of transmission. Figure 5, shows the routing of a package between a source and a destination in SDN network with Openflow Protocol.
The controller sends a transfer rule to the router A (1) so that when a package arrives at this router (2), with an IP destination which is included in the transfer rule network IP address, the router will know immediately by what interface transfer the package (proactive approach). Router B, having no management rule concerning the package reached his level, will contact the controller (4) to find out the attitude compared to the package. The controller sends a transfer rule (5) concerning the package and the router is then able to transfer the package to the next node of the network (6) (reactive method).

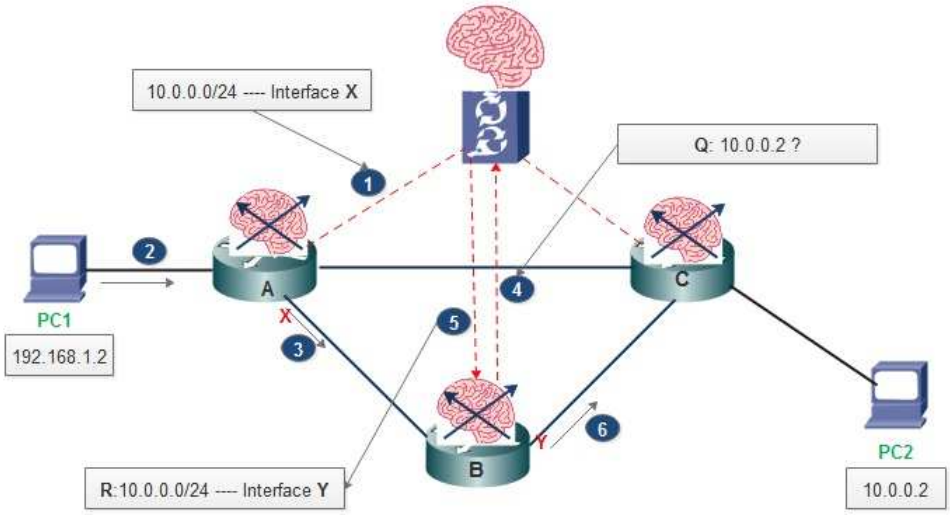


Figure 1: Diagram of a simple SDN network

The main goal of traffic engineering is to avoid congestion of some heavily-used parts of the network by controlling and optimizing routing functions (placing traffic where capacity permits) [3]. The challenge here is to adapt well to the dynamic character of the topology (case of breakdowns) and the demand.

3.1 Analysis of reactive and proactive methods

For the realization of this work we have adopted a methodology following the goal. It is, to make an assessment of the transfer time of the stream as well as load produced for this transfer of flow from the source to the destination, while considering the different routing methods.

Reactive method

In this approach, each time the router or the node receiving a package, reports the event to the controller and receives in return rules in an Exchange, in order to decide on the transfer to the appropriate following router.
An example on an architecture represented by figure 2 below.
When the node (A), receives a package from PC1 to PC2 (1), it passes the request to the

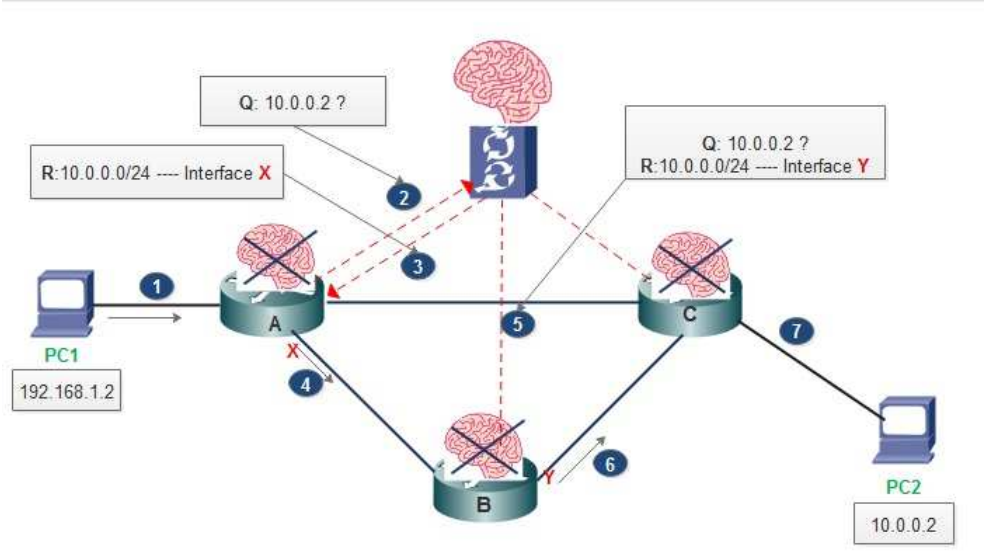


Figure 2: Descriptive diagram approach reactive

controller (2). The controller relies on the current topology of the network and communicates the transfer rule to the node (a) that runs (3).
Operations (1), (2), (3) are thus repeated in each node until the package arrives at its destination (PC2).
We evaluate the deadline for transmission and all the messages exchanged during the transfer of the flow from the source to the destination.
During the application of the rule of transfer to the controller:

- It takes a time $T = t_{jc} + t_{cj}$ for (1) and (2) on different nodes requesting the controller. For n nodes in the path from the source s to destination d, the transfer time is defined as follows:

$$T(s,d) = \sum_{j=1}^{n} (t_{jc} + t_{cj} + t_j) \tag{1}$$

- c represent the controller
- t_{jc} represents the transmission delay of node j to the controller
- t_{cj} represents the transmission delay of the controller at node j
- t_j is the verification time of the transfer rules in order to decide on the transfer

- All of the messages of the path with the source s and the destination d is define by :

$$\alpha_{ch(s,d)} = \sum_{i=1}^{n}(\alpha_{ic} + \alpha_{ci}) + \sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_{ij} \tag{2}$$

- c represent the controller
- α_{ic} the number of messages exchanged between a node i and the controller c
- α_{ci} the number of messages exchanged between the controller c and a node i
- α_{ij} the number of messages exchanged between a node i and a node j

Proactive method

Proactive behavior consists of the transmission of rules by the controller until the router receives the associated packages. Several solutions have been developed based on the algorithm of back - pressure suitable for the data networks very sensitive to latency and especially the QoS [7].

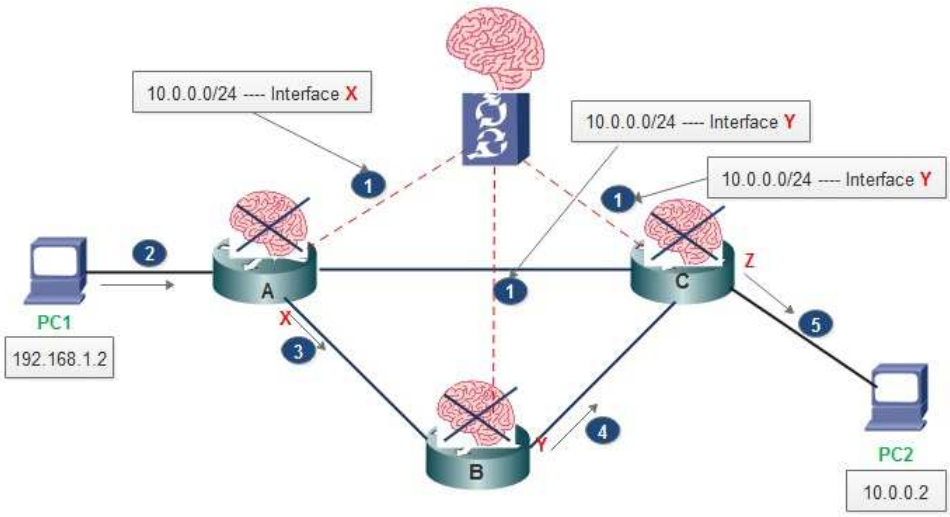


Figure 3: Descriptive diagram proactive approach

Figure 3 representing a network architecture to proactive behavior. When the node (A) of the figure 3 receives a packet from PC1 to PC2 destination (1), he runs the appropriate for this type of package and this transfer rule in its table. Operation (1) is thus repeated until the package arrives at its destination. When checking the rules of transfer from the source to the destination:

- **It takes a time** t_j which varies according to the size of the transfer of each node table and j belonging to the path between s and d.

$$T(s,d) = \sum_{j=1}^{n} t_j \tag{3}$$

- Messages sent on the path to source s and destination d are defined by :

$$\alpha_{ch(s,d)} = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_{ij} \tag{4}$$

where $\alpha_{ij} = \{^{ksilelien(i,j) \in ch_{(s,d)}}_{0,sinon}$

Experimental evaluation of the proactive and reactive approaches

In order to value these different approaches, we emulated the topology (figure 4) WAN, on a physical machine. The latter is characterized by an Intel Core i5 processor 4 cores running at 2.53 Ghz and an equal to 8 GB RAM. The machine that simulates the virtual network uses Mininet in its version 3.2, which allows to create the topology considered SDN. We will evaluate on the architecture figure 4, the impact that can have these two approaches on latency. We ignore possible failures in the network.

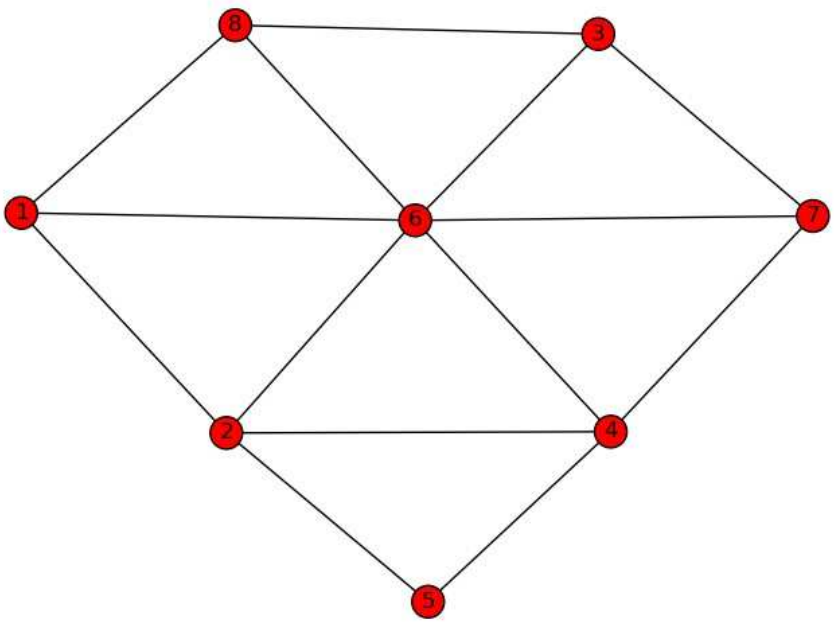


Figure 4: Topology of simulation

Our tests are carried out with the controller Ryu under its 3.2 version. It is based on python and provided APIs to communicate between applications and the network infrastructure. In addition, the protocol used for communication between the controller and the switches is OpenFlow1.3, as well as the Python3 programming language. In addition, we use Ping and Cbench tools to measure network performance metrics.
The simulation followed an iterative process and the simulation process is repeated ten (10) times. Thus we have the results of figure 5.

We see that the result of the proactive approach remains very lower than the reactive approach to each iteration. We note as well that the proactive approach allows to optimize the latency

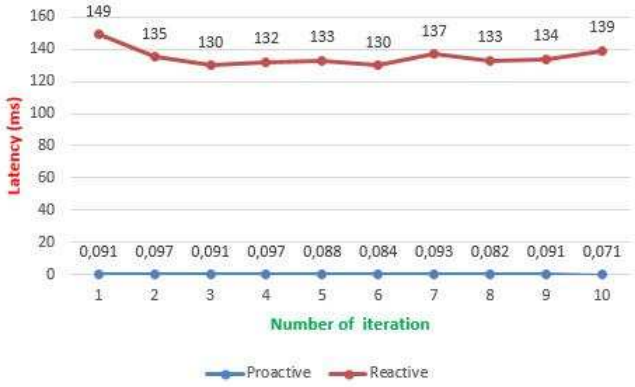


Figure 5: Latency analysis

compared to the reactive approach assuming that no failure occurs in the network. We suppose issues occurred at the level of some network nodes (the nodes have restarted) and we test the scalability of the proactive approach.

Assessment of the latency of the proactive approach with failure

When a node restart or when a break of link on a path, we see that it takes a time μ which varies according to the time of retransmission of the transfer rules of each node, which restarts on the path between the source s and destination d. We suppose that μ is the time-out period new rules of transfer on the part of the controller. Suppose that the controller sends the transfer rules to the nodes each q seconds. So when he is a loss on a transmission path, it becomes obvious that the timeout is less than or equal to the time taken by the controller before sending the new transfer rules. We have :

$$\mu = q - t_e \tag{5}$$

Where t_e is the elapsed time since the controller has transferred the transfer rules. It is obvious that $t_e \leq q$ then $\mu = q \leq t_e < q$.
Suppose a node j decides to contact the controller for new transfer rules. Time to get the rule t_p is estimated by:

$$t_p = t_{jc} + t_{cj} + t_c \tag{6}$$

Where t_{jc} is the time-out for the node j contact the controller c, t_c is the time set by the controller to process the request of the node and t_{cj} the time taken to transmit the transfer rule to the node. Still using the Simulator Mininet, Ryu under OpenFlow1.3, as well as programming Python3 language controller. We have obtained the results in figure 6 and figure 7.

These results show that when there is a failure in the network, the time the latency of the proactive method increases considerably (figures 6 and 7). Because it is not resilient to failures.

If $\mu > t_p$ in the strictly proactive case, one is obliged to wait for a time μ. However in this case, it is interesting to ask the transfer rules in order to receive them within t_p (more quickly). Where hybrid our approach, that we present in the following section.

4 Proposed approach

To realize the potential of programmable networks, we propose in this paper another variant of the algorithm of back - pressure with League of Nations based on a mixed method. Taking into account the set of messages exchanged between the source and destination during the transfer of

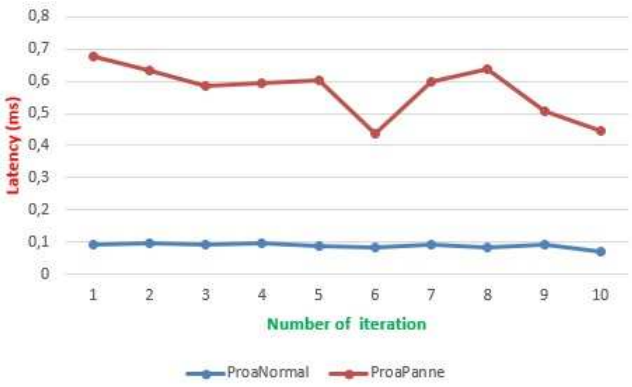


Figure 6: Latency with failure analysis

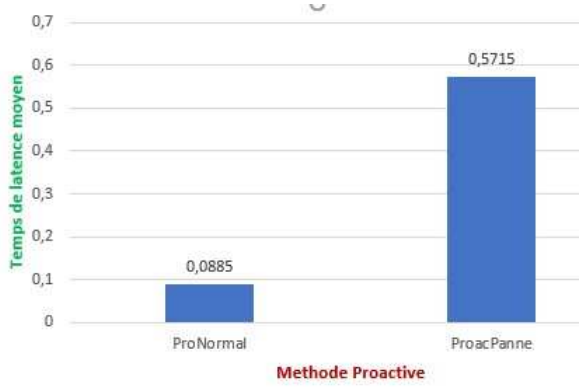


Figure 7: Average latency time with failure analysis

the stream for a better evaluation of the period. This in order to optimize the routing functions by simply placing the traffic here where capacity allows, to avoid congestion in some parts of the network strongly requested and also control failures in the network. The operation of the joint method is shown in figure 8.

4.1 Analysis of approach

When a node receives a packet for a given destination, it checks if there is a transfer for this destination rule since his transfer table and runs it so yes (node 8). Otherwise a message (PACKET_IN) (node 2) is sent to the controller and the controller determines the path satisfiable, then sends the transfer rules (PACKET_OUT) to all nodes in the path selected to avoid situations where the transfer rules are obsolete. If a node on a path restarts, and he contacts the controller as node 2 of the descriptive scheme, it will take time then the set R the path nodes that restart will use a time defined by :

$$\sum_{j \in R} t_{jc} + t_{cj} + t_c \tag{7}$$

To get the new transfer rules. If a set node restart but do not contact the controller, they will use a time defined by:

$$\sum_{j \in R} \mu_j \tag{8}$$

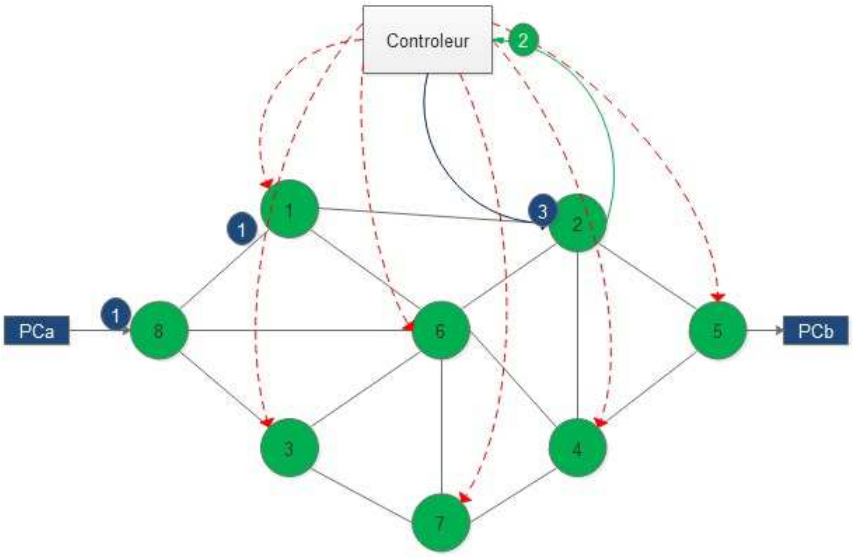


Figure 8: Descriptive diagram approach joint

to get the transfer rules. In these analyses, the transmission time of the information on a path to a source s to destination d is defined by :

$$T(ch(s,d)) = \sum_{j \in R}(t_{jc} + t_{cj} + t_c) + \sum_{j \in R} \mu_j + \sum_{j \in ch(s,d)} (t_j + t'_j) \tag{9}$$

where t_j is the transfer time of the node j, t'_j is the waiting time in the queue of the node j and μ_j is the timeout of the new rules of transfer of the node j with :

$$R = \{j \in ch(s,d) : \mu_j > t_{jc} + t_c + t_{cj}\} \tag{10}$$

$$R' = \{j \in ch(s,d) : \mu_j < t_{jc} + t_c + t_{cj}\} \tag{11}$$

The number of messages in the network nodes in the path for the transmission of information between the source to the destination is defined by :

$$\alpha_{ch(s,d)} = \sum_{i=1}(\alpha_{ic} + \alpha_{ci}) + \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_{ij} \tag{12}$$

$\alpha_{ic} = \{^{1 if the node i \in ch_{(s,d)}}_{0, else}$
$\alpha_{ci} = \{^{1 if the node i \in ch_{(s,d)}}_{0, else}$
$\alpha_{ij} = \{^{k if the link (i,j) \in ch_{(s,d)}}_{0, else}$
k with all the messages destined for the node j

4.2 Proposed algorithm

The algorithm that we offer can be described in the following :
We repeat the actions 1 to 3 to the destination.

Algorithm 1 : ReaPro-CM

Input: Network topology
Output: Transfer rules

Begin

Step 1 : Processing of a node that receives a data transfer to the destination
If a packet arrives at a node j then
  If j is not the recipient node then
    If j has a transfer rule so
      j applies this rule to transfer
    Else
      If $\mu > t_p$ then
        j contact the controller. Go to step 2.
        If j is waiting for the new transfer rule
      Endif
    Endif
  Else
    j receive the packet
  Endif
Endif
Step 2 : Controller Processing
Determine the assessments of the different links on the network
Determine the optimal paths using the dijsktra algorithm of the k shortest paths at time t
Communicate the transfer rule to the node that initiated the request as well as the nodes as a part of the path selected to reduce the tables of transfers of other nodes
Go to step 3
Step 3 : Processing of a node that receives a transfer rule
The node running the transfer rule and send the package to the next to the destination node

End

5 Simulation and results

In order to evaluate our algorithm, we emulated the topology (figure 9) WAN, on a physical machine. The latter is characterized by an Intel Core i5 processor 4 cores running at 2.53 Ghz and an equal to 8 GB RAM. The machine that simulates the virtual network uses Mininet in version 2, a widely used tool in the SDN community and it can simulate topologies classic like bus, ring, hierarchical, or customized, that reflect the exact configuration a topology of company. In order to isolate the operating system experiences, we simulate virtual machines (VMs) through a named VirtualBox virtualization application. Reserve at least two VMs, one for the emulated network and one for the controller. The performance of the proposed algorithm is presented in figure 8.

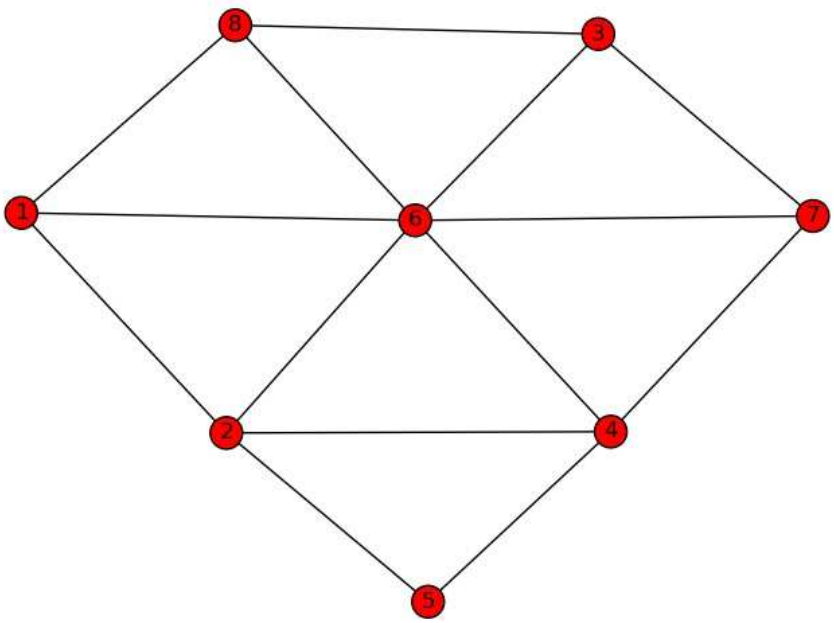


Figure 9: Topology of simulation

The simulation followed an iterative process. The simulation process is repeated ten (10) times and the results are presented in figures 10 and 11. Figure 8 shows the latency time of the routing Back-pressure trouble free approach is lower compared to the routing back - pressure with failure. Indeed, a failure of a node or link requires wait for a node controller of new transfer rules. However, the controller sends the transfer rules according to a period. As a result, a node that is a failure of a link or falling down, must wait for a period of time less than or equal to the delay of the controller transfer rules.

To minimize the latency time found with the results of figure 5, we proposed an alternative approach that improves significantly the lag time when a failure occurs in the network as shown in figure 5. The simulation results show that our approach (ReaPro-CM) latency time, remains much lower than that of Back-pressure with failure (Voting-BP with breakdown). Indeed, when a failure occurs in the network (link broken) during transmission of the stream, our approach provides a backup path allowing to move this flow without however wait until the controller for

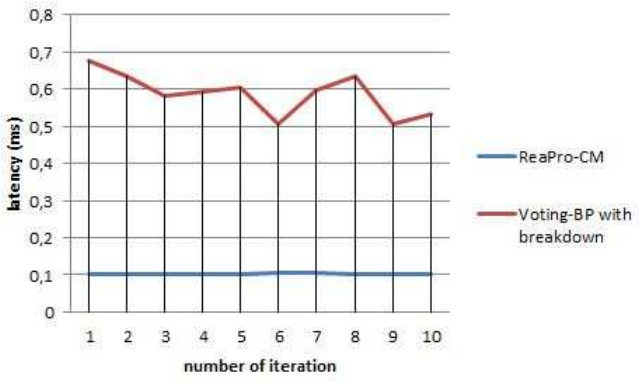


Figure 10: Latency with failure and rescue way

new transfer rules. When a failure occurs at the level of a node (a node reboot), our approach evaluates the time remaining until the node receives new transfer rules. If this time is very high, the node asks the controller new rules of transfer immediately. These principles implemented in our approach to routing can significantly reduce latency when there is a failure in the network (figure 11).

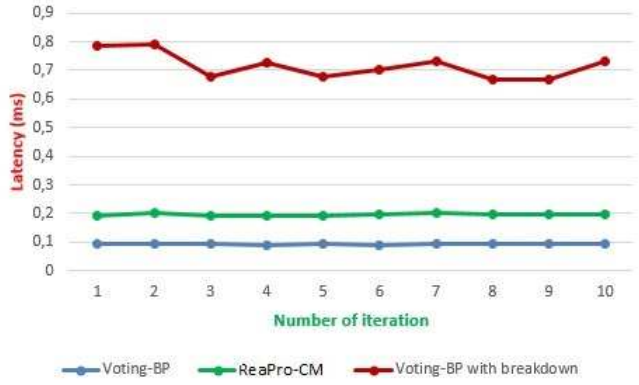


Figure 11: Comparison approach mixed with failure and proactive approaches

6 Conclusion

This article offers an approach of mixed multi-path routing in networks of data with SDN to resolve failures coming into the network. These failures have a significant impact on latency where interest to propose a joint routing based on the proactive and reactive approaches. Mathematical analysis and simulation results show both the algorithm that we offer allows to optimize the transmission time in networks of data with League if all the messages exchanged during the transfer of information are taken into account. Because the main problem in networks with SDN SDN routing tables cannot contain only a very limited number of rules.

Bibliography

[1] Bui, L.; Srikant, R.; Stolyar, A. (2009); Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing, *INFOCOM 2009, IEEE*, 2936–2940, 2009.

[2] Erickson, D. (2013); The beacon openflow controller, *Proc. of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, 13–18, 2013.

[3] Gojmerac, I.; Reichl, P.; Jansen, L. (2008); Towards low-complexity internet traffic engineering: the adaptive multi-path algorithm, *Computer Networks*, 52(15), 2894–2907, 2008.

[4] Goransson, P.; Chuck B. (2014); *Software Defined Networks: A Comprehensive Approach*, British Labrary Cataloguing-in-Publication Data, 2014.

[5] Jacobson, V.; Smetters, D.K.; Thornton, J.D.; Plass, M.F; Briggs, N.H.; Braynard, R.L.(2009); Networking named content, *Proc. of the 5th Intl. Conf. on Emerging networking experiments and technologies*, 1–12, 2009.

[6] Kulkarni, S.S.; Badarla, V.(2014); On multipath routing algorithm for software defined networks, *Advanced Networks and Telecommuncations Systems (ANTS), 2014 IEEE Intl. Conf. on*, 1–6, 2014.

[7] Luca, R.L.; Ciotirnae, P.; Popescu, F. (2016); Influence of the QoS Measures for VoIP Traffic in a Congested Network, *Intl. J. of Computers Communications & Control*, 11(3), 405-413, 2016.

[8] Maglaras, L.A.; Katsaros, D. (2011); Layered backpressure scheduling for delay reduction in ad hoc networks, *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on*, 1–9, 2011.

[9] Maldonado-Lopez, F.A. ; Calle, E.; Donoso, Y. (2016); Checking Multi-domain Policies in SDN, *Intl. J. of Computers Communications & Control*, 11(3), 428-440, 2016.

[10] Moeller, S.; Sridharan, A.; Krishnamachari, B.; Gnawali, O.(2010); Routing without routes: The backpressure collection protocol, *Proc. of the 9th ACM/IEEE Intl. Conf. on Information Processing in Sensor Networks*, 279–290, 2010.

[11] Nithin, M.; Ao, T.; Dahai, X. (2013); Optimal link-state hop-by-hop routing, *Network Protocols (ICNP), 2013 21st IEEE Intl. Conf. on*, 1–10, 2013.

[12] Pinheiro, B.; Cerqueira, E.; Abelem, A. (2016); NVP: A Network Virtualization Proxy for Software Defined Networking, *Intl. J. of Computers Communications & Control*, 11(5), 697-708, 2016.

[13] Tassiulas, L.; Ephremides, A. (1992); Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks, *IEEE transactions on automatic control*, 37(12), 1936–1948, 1992.

[14] Tootoonchian, A.; Gorbunov, S.; Ganjali, Y. (2015); On Controller Performance in Software-Defined Networks, *IEEE/Mediterranean Electrotechnical Conference*, 1–6, 2015.

[15] Ying, L.; Shakkottai, S.; Reddy, A.; Liu, S. (2011); On combining shortest-path and back-pressure routing over multihop wireless networks, *IEEE/ACM Transactions on Networking (TON)*, 19(3), 841–854, 2011.

[16] Foodlight [Online]. http://openflowhub.org, Accessed: March 10, 2016.

[17] OpenFlow Switch Specification [Online]. https://www.opennetworking.org/software-defined-standards/specifications/, Accessed: June 25, 2016.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 837-852, October 2018.

Exploring Analytical Models for Proactive Resource Management in Highly Mobile Environments

Y. Kirsal, V.V. Paranthaman, G. Mapp

Yonal Kirsal*
European University of Lefke
Faculty of Engineering
Department of Electronics and Communication Engineering
Lefke, North Cyprus
TR-10 Mersin, Turkey
*Corresponding author: ykirsal@eul.edu.tr

Vishnu Vardhan Paranthaman, Glenford Mapp
School of Science and Technology
Middlesex University
London, NW4 4BT
v.paranthaman@mdx.ac.uk, g.mapp@mdx.ac.uk

Abstract: In order to provide ubiquitous communication, seamless connectivity is now required in all environments including highly mobile networks. By using vertical handover techniques it is possible to provide uninterrupted communication as connections are dynamically switched between wireless networks as users move around. However, in a highly mobile environment, traditional reactive approaches to handover are inadequate. Therefore, proactive handover techniques, in which mobile nodes attempt to determine the best time and place to handover to local networks, are actively being investigated in the context of next generation mobile networks. Using this approach, it is possible to enhance channel allocation and resource management by using probabilistic mechanisms; because, it is possible to explicitly detect contention for resources. This paper presents a proactive approach for resource allocation in highly mobile networks and analysed the user contention for common resources such as radio channels in highly mobile wireless networks. The proposed approach uses an analytical modelling approach to model the contention and results are obtained showing enhanced system performance. Based on these results an operational space has been explored and are shown to be useful for emerging future networks such as 5G by allowing base stations to calculate the probability of contention based on the demand for network resources. This study indicates that the proactive model enhances handover and resource allocation for highly mobile networks. This paper analysed the effects of β and α, in effect how these parameters affect the proactive resource allocation requests in the contention queue has been modelled for any given scenario from the conference paper "Exploring analytical models to maintain quality-of-service for resource management using a proactive approach in highly mobile environments" [9] (doi:10.1109/ICCCC.2018.8390456).[a]
Keywords: analytical modelling, proactive resource management, contention, time before handover, time to handover, highly mobile environments.

[a]Reprinted (partial) and extended, with permission based on License Number 4386390665938 ©[2018] IEEE 7th International Conference on Computers Communications and Control (ICCCC)

1 Introduction

With the rapid development of mobile communication technologies such as WiFi, femto-cells, Long-Term Evolution Advanced (LTE-A), Vehicular Ad-Hoc Network (VANET), integration

of various other wireless networks known as heterogeneous networking (HetNet) has become necessary to provide the mobile node (MN) with ubiquitous communication. For instance, where, a MN can connect to a nearby LTE, the calls rejected by LTE networks due to lack of radio access can overflow to overlaying WiFi networks, thus reducing the call blocking probability and improving bandwidth utilization in cellular overlay networks. However, this may result in frequent forced handover in those areas covered by small cells and leads to extra signalling overheads [8]. In these environments, traditional handover techniques, which depend on a reactive approach, have been found to be inadequate because of high speeds as resources must be quickly allocated and deallocated as the mobile user moves around. Hence, good resource management must be considered as a key enabling functionality to allow seamless connectivity in highly mobile heterogeneous environments.

The advent of latest wireless technologies, high data rate multimedia services and heterogeneous user equipments, necessitate the development of efficient techniques for resource management in highly mobile heterogeneous environments [15]. In addition, the high mobility between HetNets may lead to a high number of unnecessary handovers as well as handover failures due to the user's velocity [6]. Unnecessary handover occurs when the mobile user's dwell time within the network coverage is less than the handover process from the neighbouring networks to the system. Thus, the mobile user leaves the network coverage area before the handover process is executed [20]. This causes network connection interruption, thus Quality of Service (QoS) of the system degrades. On the other hand, handover failures may lead to data loss, long delays, and even communication interruptions, which are not tolerable for safety life critical applications such as accident prevention, emergency and disaster management as required in Intelligent Transport Systems (ITS) [1].

Proactive behaviour by systems refers to anticipatory, self-initiated and change-oriented behaviour in situations. Proactive behaviour involves acting in advance of a future situation, rather than just reacting. It means taking control and making things happen rather than just adjusting to a situation or waiting for something to happen. Therefore, this proactive approach helps any system to perform better than a reactive approach [17]. It is very important to provide a seamless service to all the users in a network and therefore, these users and their contention for resources have to be looked in detail. In this context, client-based handover can be more efficient, since the metrics for the handover decision mechanism can be monitored by the user equipment (UE) from its wireless interfaces and can be used to decide to trigger the handover. In addition to the need for new handover decision mechanisms, there is also a need for better resource reservation mechanisms due to varying traffic characteristics, QoS application requirements and wireless channel conditions at the access point (AP). Efficient resource management is needed to optimize the performance of a wireless network because only a limited number of simultaneous calls can be hosted by a wireless cell. Therefore, incoming handover calls and new calls should not compromise the quality of the ongoing calls in the cell. Traditionally, user contention has been used to analyse the need for specific resources such as radio channels by mobile users. However, it is possible to use this contention to proactively manage these resources. Implementing the appropriate pre-planned resource allocation when the system state changes are known as proactive resource management. It provides the ability to specify end-to-end QoS quickly and to determine whether the specified QoS can be provided in advance of when these resources will be required [3].

Y-Comm is an architecture that has been designed to build future mobile networks by integrating communications, mobility, QoS and security [13]. The researchers of Y-Comm have explored new parameteres such as time before handover (TBH) and network dwell time (NDT) as shown in Figure 1 [18]. These two parameters was introduced in [16] and are used to provide a new proactive approach for handover and resource allocation in heterogeneous environments.

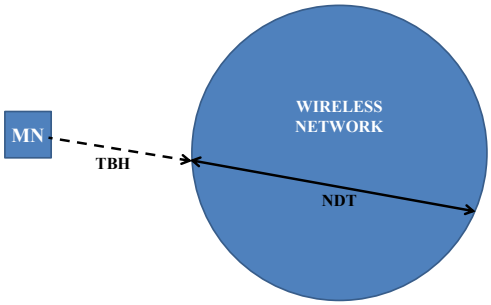


Figure 1: Illustrating time before handover (TBH) and network dwell time (NDT)

Thus, in this paper, possible contention probabilities are considered and operational areas are explored for certain scenarios using an analytical modelling approach. No contention, partial contention and full contention are three possible contentions that are considered based on two key parameters i.e., time to get resource ($\mathbb{T}$) and resource hold time ($\mathbb{N}$). $\mathbb{T}$ is the time when actual resource requested is available for use i.e., even after entering the network's coverage range with a successful soft handover, the resource required by the MN might not be available, for example, other users might be holding the resource. $\mathbb{N}$ is the resource usage time or when actual exchange of data is taking place. Thus, in order to achieve seamless handover in highly mobile environments, it is important to predict the resource availability considering $\mathbb{T}$ and $\mathbb{N}$.

This paper extends the conference paper [9]. The key additions of this journal version are as follows. It adds more on methods to formulate β and α more accurately for any given scenarios. A proactive approach used in [9] is analysed for resource allocation to formulate β and α by investigating all contention probabilities for channel access to a wireless network, results are obtained and verified using analytical models. The proposed model and results presented improve the capacity and services delivered by mobile networks. The rest of the paper is organized as follows: Section 2 includes and describes more related works. Section 3 gives details of resource allocation and contention in highly mobile environments. Section 4 gives a detailed description of our proposed model with the derived Markov queuing model for the new proactive approach and Section 5 shows the results and effects of β and α. Section 6 concludes and gives the future direction of the work.

2 Related work

Bandwidth is an extremely valuable and scarce resource in mobile networks. Therefore, efficient mobility-aware bandwidth reservation is necessary to support multimedia applications (e.g., video streaming) that require QoS. Several research efforts were carried out looking at routing, security and applications for highly mobile environment but very few addressed handover and resource allocation issues.Recent works in [2] and [11] clearly show that researchers are interested in proactive handover or predictive handover mechanisms, however these efforts considered parameters like user preferences, user location and application requirements.

Several studies of the vertical handover procedure, mobility management and common radio resource management schemes in heterogeneous environments have been reported in the literature [5, 21, 24]. These studies show that there are several dynamic factors that must be considered in vertical handover decisions for effective network usage including policies to determine whether or not to handover should occur as well as mechanisms to determine the best network to handover.

The work in [4] has proposed a model for enhancing the modelling of vertical handover in integrated cellular and wireless local area network environments. In addition, in [10] an intelli-

gent handover decision approach is proposed to minimize the handover failures and unnecessary handovers whilst maximizing the usage of resources in highly mobile environments. On the other hand, QoS and Carrier to Interference-and-Noise Ratio (CINR) are some common metrics that must also be considered in this context. Any proposed solution must also be scalable because in the future the MN may have the ability to handover to hundreds of possible target networks. In this context, client-based handover can be more efficient, since the metrics for the handover decision mechanism can be monitored by the UE from its wireless interfaces and can be used to decide to trigger the handover [23]. Knowing the velocity and current position of an MN could help to estimate where the MN is heading, thus the next position of MN where handover might be performed can be predicted.

Proactive handover in which the MN actively attempts to decide when and where to handover has been shown to be an efficient handover policy mechanism to minimize packet loss and service disruption as an impending handover can be signalled to the higher layers of the network protocol stack [12]. A mobility prediction scheme for MNs was proposed in [14]. Probability and Dempster-Shafer processes were applied to predict the likelihood of the next destination based on the user's habits (e.g., frequently visited locations). In addition, second-order Markov chain process was applied at each road junction for predicting the likelihood of the next road segment transition, given the direction to the destination and the path from the trip origin to that specific road junction. A proactive unnecessary handover avoidance scheme was proposed in [22] for the LTE-A small cells. Unnecessary handover was avoided by calculating the probability of the active time during the dwell time in one cell and comparing the same to find if pre-defined threshold exceeds certain value. Here, a model to estimate the dwell time in the small cell was not considered. In [7], Fernandez *et al.* highlighted the need for a decision mechanism for choosing an appropriate point of attachment for high mobility nodes.

All studies above suggested resource management as well as handover guidelines in order to achieve seamless communication in high mobility environments. However, in this paper the work in [9] has been extended and analysed the request not being affected and staying in queue considering all possible contention probabilities in highly mobile environments based on the time the mobile user needs to acquire network resources before the handover.

3 Resource allocation and contention in mobile networks

In the era of increasing connectivity, wireless networks are gaining importance in several applications. One of their obvious benefits is the support of mobile users. In complex network infrastructures consisting of many APs, seamless connectivity becomes an important aspect because many applications rely on real-time communication and therefore need seamless handover between APs. If there is a decrease in quality the connection below a minimum threshold then the connection is lost, and the client must scan the wireless channels and look for a new AP in order to establish a new connection. The scanning procedure typically takes a long time and thus prohibits seamless connectivity [6]. Most of the analysis on handover and resource allocation was service oriented and it is important to look into the impact on individual users based on their mobility and dwell times in a network.

In mobile environments it is also necessary to understand why contention should be considered and this can be done by looking at classical approach to analyse the performance of mobile networks. Classical handover occurs when a MN changes its points of attachments (PoA) from the current wireless network to another network using a reactive approach i.e., the handover is initiated only after the MN is within the network coverage of the next wireless network. Classical handover uses a reactive approach i.e., when the MN is moving away from the area covered by one cell and entering the area covered by another cell the call is transferred to the second cell in order

to avoid call termination when the MN gets outside the range of the first cell. In this approach in order to start the handover, the MN should be in the coverage range of the second cell and has to exchange the relative information to start the handover and complete the handover before exiting the first cell.

There are a number of parameters that need to be known to determine whether a handover is required. The signal strength of the base station (BS) with which communication is being made, along with the signal strengths of the surrounding stations. Additionally the availability of channels also needs to be known. The MN is obviously best suited to monitor the strength of the BSs at its location, but only the cellular network knows the status of channel availability and the network makes the decision about when the handover is to take place and to which channel of which cell [24].

Accordingly, the MN continually monitors the signal strengths of the BSs it can hear, including the one it is currently using, and it feeds this information back to the BS. When the strength of the signal from the BS that the MN is using starts to fall to a level where action needs to be taken then the cellular network looks at the reported strength of the signals from other cells reported by the MN. It then checks for channel availability, and if one is available it informs this new cell to reserve a channel for the incoming MN. When ready, the current BS passes the relevant information for the new channel to the MN, which then makes the change. Once there the MN sends a message on the new channel to inform that it is now within the coverage of the new network. If this message is successfully sent and received then the network shuts down communication with the MN on the old channel, freeing it up for other users, and all communication takes place on the new channel.

Under some circumstances such as when one base transceiver station is nearing its capacity, the network may decide to hand some MNs off to another base transceiver station they are receiving that has more capacity, and in this way reduces the load on very busy the base transceiver station. Hence, access can be opened to the maximum number of users. In fact channel usage and capacity are very important factors in the design of a cellular network.

In [10] a service oriented approach for MNs is presented, considering that the services will be resumed as soon as the MN moves to a different network. But the work did not focus on the resources available at the AP or BS and the effects of mobility in acquiring this resource. Understanding this effect is very important as the MNs would be waiting to acquire a channel and might move out of the current network due to mobility without service. The problem with the classical handover approach is that the AP/BS does not know in advance the network requirements of MNs heading towards it. Due to this the MN even after entering the network coverage, it will have to wait until the resource becomes free. In a highly mobile environments, MNs will have less time to spend in a network coverage therefore, the AP/BS has to anticipate the network conditions much before based on the MNs about to reach its network in order to have an effective resource utilization.

3.1 Network coverage parameters for mobile networks

In this work we are further exploring and redefining the communication range segments [15] as shown in Figure 2 which can be put into effective use for achieving both proactive handover and resource allocation for a highly mobile environment.

Figure 2 shows a more advanced scenario in which three consecutive overlapping wireless networks are segmented based on various key time variables which can be used to enhance handover and resource allocation. Time before handover (Υ) is the time after which the handover process should start and Time to handover ($\hbar$) is the time before which the handover to next coverage range has to be completed, if not it will result in a hard handover. Network Dwell

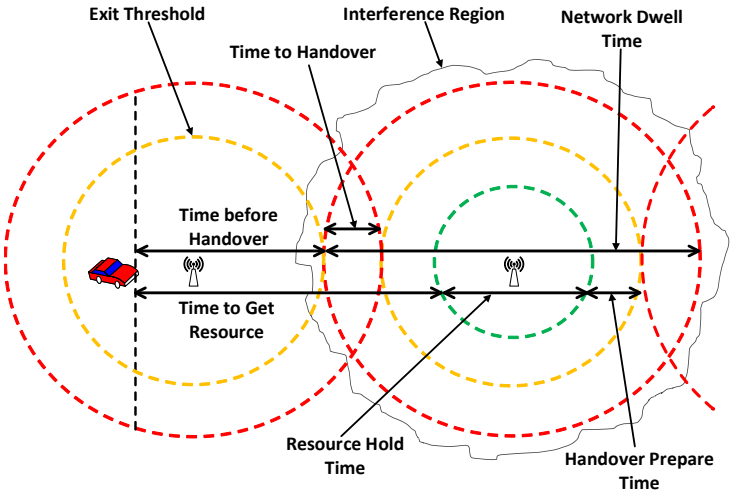


Figure 2: Communication range segmentation

Time ($\aleph$) as defined in Y-Comm Framework is the time MN will spend in the coverage i.e., the Network Dwell Distance (NDD) of new network. Time to get resource, $\mathbb{T}$ is the time when actual resource requested is available to the requested user i.e., even after entering the network's coverage range with a successful soft handover, the resource required by the MN might not be available, for example, other users might be holding the resource. Resource Hold Time, $\mathbb{N}$ is the resource usage time or when actual exchange of data is taking place. Handover Prepare Time (ϱ) is the time taken to prepare for handover during which the resource usage or data transmission will be paused and will be resumed after successful handover to the new network. Usually $\hbar$ and ϱ are very small compare to the values of other segments and therefore, $\mathbb{T}$ can be approximately equal to Υ if there are resources available in the new network, i.e., if there is no contention.

With the knowledge of these coverage parameters, it is possible to enhance the resource management based on the user contention for resources in a mobile network with proactive handover. A new proactive resource allocation based on the user contention to acquire a wireless channel resource is presented in the following section.

4 The proposed analytical model for proactive resource allocation

This section presents the proposed model to maintain QoS for using proactive resource management approach in highly mobile environments. In this paper, all the possibilities of contention interaction i.e., no contention, partial contention and full contention have been explored to show the practicability as well as the operational spaces of the overall system for highly mobile users in cellular system. These interactions may result in a request leaving the contention queue due to full contention with a subsequent request or the request in queue may be rearranged due to partial contention. The contention queue only queues requests before the MNs reach the next network. Once the MN reaches the relevant network, its channel request in the contention queue will be placed in the channel allocation queue. The proposed proactive resource allocation model is shown in Figure 3.

In the proposed approach, the decision algorithm decides whether the MN's request will be admitted to the system based on values of $\mathbb{T}$ and $\mathbb{N}$ of all the MNs requesting a channel. There are three possible contention possibilities that affect mobile users in this scenario. The probability of full contention before entering the queue is given as α. In addition, the probability of full contention occurring in the contention queue can be expressed as β. Finally, partial contention

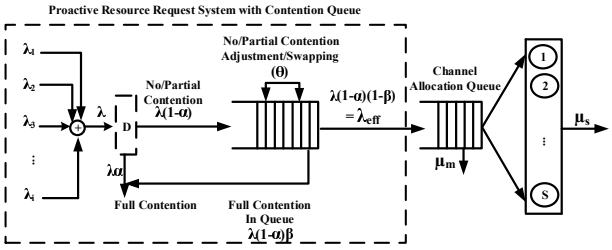


Figure 3: Proactive resource allocation queuing system with β and α

in the contention queue can be expressed as θ and this may result in modification of the request and/or re-ordering of request in the contention queue. Since θ does not involve in leaving the contention queue therefore, it will have no overall effect with respect to the rate of transfer to the channel allocation queue in this paper. The formulation of θ and validation is still in progress and will be considered as a future work.

4.1 The proposed proactive Markov queuing model

The proposed proactive resource allocation queuing model considers S number of channels and can allow i requests at time t. Q is the queueing capacity of the proposed system. The arriving requests may be sent from different MN to the system. It is assumed that the originating calls can join the system with an arrival rate of λ_O. Similarly, the handover calls can join the system with an arrival rate of λ_H. Hence, the total arrival rate is λ=λ_H+λ_O. According to [25], for a two-dimensional fluid model, the arrival rate of handover calls can be obtained as follows:

$$\lambda_H \approx \frac{\mu_m}{\mu_s}\lambda_O \tag{1}$$

Hence, the inter-arrival time of consecutive task follows the Poisson process which can be distributed as an exponential distribution with arrival rate λ=λ_H+λ_O.

$$\lambda = \sum_{i=1}^{S} \lambda_i \tag{2}$$

If there is no full contention in the contention queue, the mobile calls can join the system with an arrival rate of $\lambda(1-\alpha)$. Thus, scheduling and arrangement of requests take place and the total effective arrival rate can be calculated as:

$$\lambda_{eff} = \lambda(1-\alpha) * (1-\beta) \tag{3}$$

Figure 4 shows the state diagram of the proposed model. Let's define the states i (i=0,1,2,$\cdots$,S+Q) as the number of requests in the system at time t.

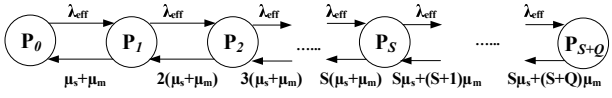


Figure 4: State diagram of the proposed approach

In proposed model, $\mathbb{T}$ and $\mathbb{N}$ are exponentially distributed with a mean rate of μ_s and μ_m, respectively. μ_m can be calculated based on the literature in [9, 10, 25]. Equation 4 is used in

the literature for the dwell time in wireless and mobile systems [10, 19, 25] for handover queuing models. Thus, μ_m can be calculated and described as follows:

$$\mu_m = \frac{E[\nu] \cdot L}{\pi \cdot A} \tag{4}$$

where, $E[\nu]$ is the average velocity (ν) of MN, L is the length of the perimeter of cell (a cell with an arbitrary shape is assumed), and A is the area of the cell. Hence, the total channel holding time of a call is exponentially distributed with mean $1/(\mu_s + \mu_m)$. If there are fewer than S requests in the system, $i < S$, only i of the S channels are busy and the combined service rate for the system is $i(\mu_s + \mu_m)$ or $S\mu_s + i\mu_m$ if $S \leq i \leq S + Q$ as shown in Figure 4.

$$\mu_i = \begin{cases} i(\mu_s + \mu_m) & 0 \leq i < S \\ \\ S\mu_s + i\mu_m & S \leq i \leq S + Q \end{cases} \tag{5}$$

ρ is the traffic intensity in the system, where $\rho = \lambda_{eff}/(\mu_s + \mu_m)$. As the requests are rejected from entering the system (α) and requests in the contention queue can also be removed (β) due to full contention with subsequent requests. Assuming a system in a steady state, the state probabilities, P_i's, can be obtained as in equation 6. P_i is the probability that there are i calls in the system.

$$P_i = \begin{cases} \frac{\rho^i}{i!} \cdot P_0 & 0 \leq i \leq S \\ \\ \frac{\frac{\rho^S}{S!} \cdot \lambda_{eff}^{i-S} \cdot P_0}{\prod_{j=S+1}^{i} [S\mu_s + (j-S)\mu_m]} & S < i \leq S + Q \end{cases} \tag{6}$$

In Equation 6, P_i is the probability that there are i calls in the system. P_0 can be defined as follows:

$$P_0 = \left[\sum_{i=0}^{S} \frac{\rho^i}{i!} + \sum_{i=S+1}^{S+Q} \frac{\frac{\rho^S}{S!} \cdot \lambda_{eff}^{i-S}}{\prod_{j=S+1}^{i} [S\mu_s + (j-S)\mu_m]} \right]^{-1} \tag{7}$$

The mean queue length (MQL) i.e., the average number of requests in the system can then be calculated as $MQL = \sum_{i=0}^{S+Q} i \cdot P_i$.

$$MQL = \left[\sum_{i=0}^{S} \frac{i\rho^i}{i!} + \sum_{i=S+1}^{S+Q} \frac{i \cdot \frac{\rho^S}{S!} \cdot \lambda_{eff}^{i-S}}{\prod_{j=S+1}^{i} [S\mu_s + (j-S)\mu_m]} \right] P_0 \tag{8}$$

Similarly, the blocking probability (P_B), throughput (γ) and mean response time (MRT) of the

system can be calculated as follows:

$$P_B = P(S+Q) = \frac{\frac{\rho^i}{S!} \cdot \lambda_{eff}^Q \cdot P_0}{\prod_{j=S+1}^{S+Q} [S\mu_s + (j-S)\,\mu_m]} \tag{9}$$

$$\gamma = \sum_{i=0}^{S+Q} i \cdot \mu_i P_i \tag{10}$$

$$MRT = \frac{MQL}{\gamma} \tag{11}$$

4.2 Request management in the contention queue

An example of the proactive resource allocation in queue is shown in Figure 5. Let us consider a simple queue and requests with (𝕋,ℕ) arrive to the queue as shown in Figure 5. Req_A arrives with (10,10) in seconds to the contention queue, the decision algorithm checks the queue and Req_A is queued at the front as the queue is empty.

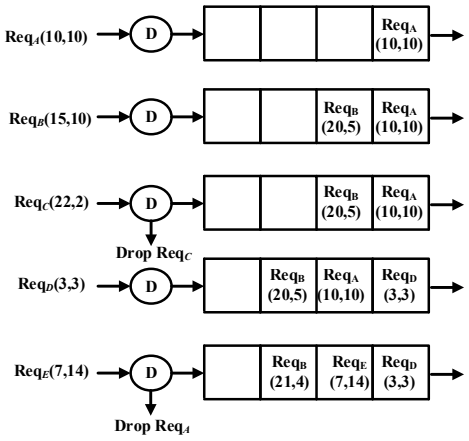


Figure 5: An example of proactive resource allocation management in queue

Now Req_B arrives with (15,10) and the time when it needs the channel and is when Req_A will still be using the channel resulting in partial contention for Req_B. Because, Req_A release the channel at the end of 20. Therefore, Req_B is modified to $\mathbb{T}_\mathbb{B} = 20$ and $\mathbb{N}_\mathbb{B}$ is modified to $(15 + 10) - 20$ hence, the modified request for Req_B is (20,5). Req_C now arrives with (22,2) however Req_B will release the channel at 25 which means that Req_C will never get the channel and therefore it is not admitted to the contention queue due to full contention. A rejection reply is sent to Node C causing it to do an immediate handover to another network. Now Req_D arrives with (3,3), there is no contention with Req_A or Req_B and therefore, it is placed at the head of the queue since $\mathbb{T}_\mathbb{C} + \mathbb{N}_\mathbb{C} < \mathbb{T}_\mathbb{A}$ i.e., $6 < 10$. Req_E arrives with (7,14) this results in no contention with Req_D, full contention with Req_A and hence Req_A is ejected from the contention queue because $\mathbb{T}_\mathbb{E} < \mathbb{T}_\mathbb{A}$ and $\mathbb{T}_\mathbb{E} + \mathbb{N}_\mathbb{E} > \mathbb{T}_\mathbb{A} + \mathbb{N}_\mathbb{A}$. Req_B experiences partial contention and hence the request is modified accordingly.

4.3 Useful service vs mobile service

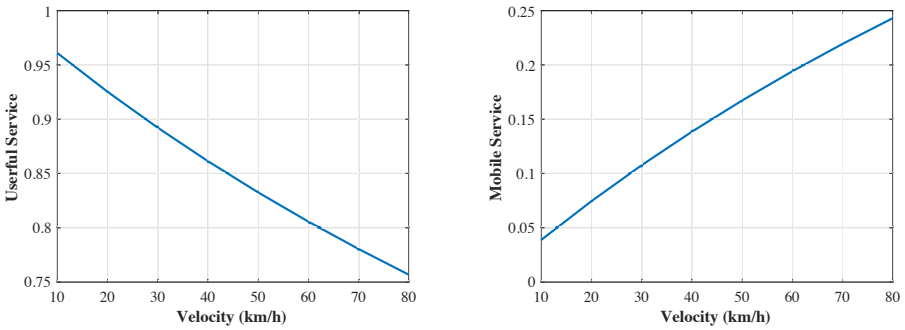
Since the MN can leave the queuing system due to service (μ_s) or due to mobility (μ_m), it is necessary to distinguish these two events to properly reflect the performance of the system. We

therefore define two concepts which are important in a mobile environment. The first is useful service in classical handover, U_{sc}, where the MN leaves the system after using the channel. When the MN leaves the system due to mobility and is not served by the channel, this is called Mobile Departure or Service, U_{mc}. For the classical case, we can represent these parameters as follows:

$$U_{sc} = \frac{S\mu_s}{S\mu_s + Q\mu_m}, \quad U_{mc} = \frac{Q\mu_m}{S\mu_s + Q\mu_m} \tag{12}$$

We say that a system in which $U_{sc} > 0.5$ represents that 50% of the overall service rate is due to the channel being used, while $U_{sc} < 0.5$ represents under utilization because most of the overall service rate is due to MNs leaving the system due to mobility. These concepts are applied to give a better understanding of the effects of mobility.

The system parameters used are taken from [9] for consistency. The system has a fixed number of identical channels: $S = 12$. Q is the queuing capacity, which represents the number of requests waiting for service and is limited to 100. The service rate of the system μ_s is 0.01 requests/sec. The average speed of the MN and the radius of the network are taken as 10km/h to 80km/h and 1000m for all calculations, respectively. The rates are translated into requests per second in order to use consistent values.



(a) Useful service (b) Mobile service

Figure 6: Useful service vs mobile service

We can observe from Figure 6 that as the velocity of the MN increases, the useful service of the system is significantly reduced and the mobile service is increasing. This means that the number of MNs which are moving out of the network due to mobility without being served is increasing. Hence, in mobile networks it is necessary to take into account the mobility aspects to ensure effective resource management, so that we can optimize the useful service of the channels. In these environments most users will leave the system without being served resulting in poor user experience as well as poor system performance. The key observation is that because of the reactive approach in the classical model, MNs can queue for a channel with no hope of getting the channel while they are in the coverage area of the network. As a result this is useless waiting and if this inability to obtain a channel can be signalled to the MN before queueing for the channel then it would allow the user to immediately look for an alternative network and hence would improve the Quality of Experience (QoE) as well as the overall system performance. This means that the analysis of contention between different MNs has to be analysed in detail so that only MNs that have a chance of getting the channel should be queued for service.

5 Results and discussion

This section presents numerical results in order to show the accuracy and effectiveness of the proposed analytical model of the proactive approach to improve resource allocation in highly mobile environments. In addition, the results obtained are analysed and formulated for β and α by investigating all contention probabilities. The system parameters used are mainly taken from [9] and relevant literature [10, 19]. The system has a fixed number of identical channels: S=12 and the queue capacity is 100. It is assumed that the moving direction of the mobile users can be detected by the BS/AP using a control channel. In addition, a mixed traffic pattern is also assumed, as in [9] where on average a minimum of 2 slots are 0.5 ms. Hence, the rates are translated into request per second in order to use consistent values. The service rate of the mobile users μ_m is calculated using Equation 4. The requests are entering the system due to the proposed contention system with arrival rate λ_{eff}.

5.1 Effects of α and β

The focus of the paper is to understand the effects of β and α by investigating all contention probabilities together with the operative area of various parameters to maintain QoS for resource management. In [9] all possible contention probabilities are considered with different scenarios. However, more accurate results of a request not being affected by β and α are analysed and presented. In addition, 3D graphs are also generated in order to see a complete picture for such systems. The parameters taken for the rest of the figures are; S=12, Q=100, μ_s=0,02(reqs/sec), $E[\nu]$=50km/s and R=1000m, unless stated otherwise.

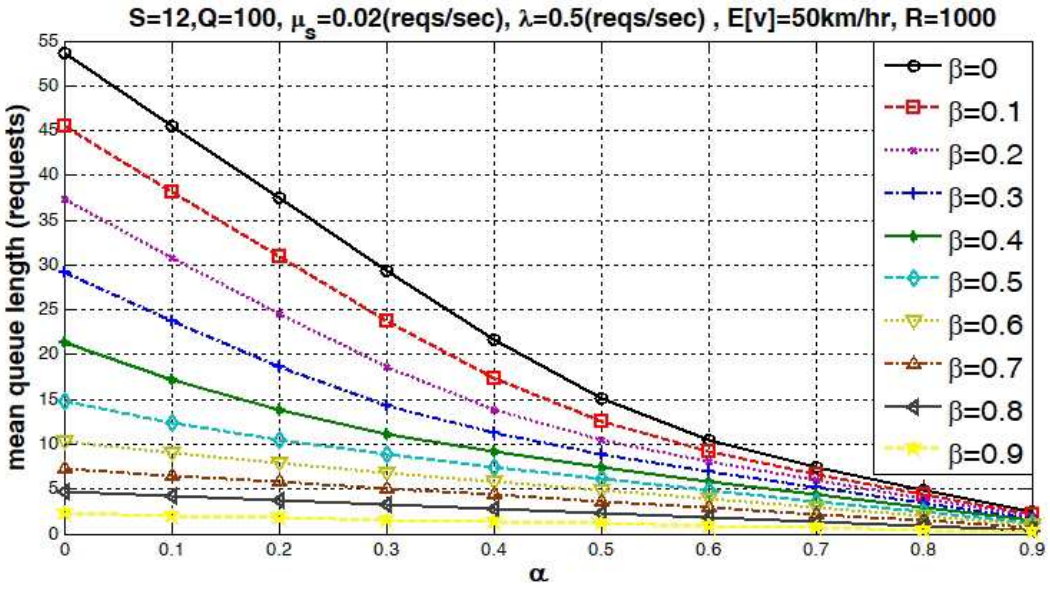
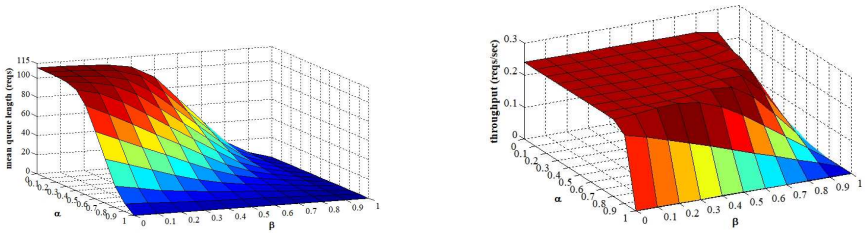


Figure 7: MQL results vs α with different β

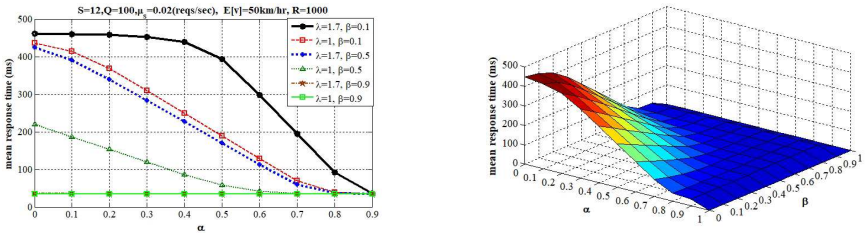
The figure 7 shows MQL results as a function of α considering all full contention probabilities occurring in the queue (β) for a light traffic loads (e.g, λ=0.5). The figure clearly shows the importance of β. Even though, the system has a light traffic loads, the MQL increases without considering β. However, the MQL decreases as β increases. This is due to the full contention occurring in the queue. MNs are not allowed to the system since they will not have enough time to get service. MNs carry on their journey without entering the system.



(a) Mean queue length (b) Throughput

Figure 8: 3D representation of all contention probabilities for λ=1.7

Hence, the operational area of the MNs in acquiring a resource is shown in Figure 8 for high traffic loads (e.g., λ=1.7) in 3D. For instance, the system is almost full (MQL=110.98 reqs/sec) when there is no contention (β=α=0). However, considering both contention probabilities the decision of acquiring a resource for MNs can be efficiently achieved by analysing the results in Figure 8. The Figure 8a shows that, the MNs start to line up when β=0.6 and α=0.7 for high traffic loads (e.g., λ=1.7). However, the MNs can get a channel even when the contention queue is fully loaded. In addition, when the full contention probability in the queue is high in the system, it affects the overall performance significantly. For instance, the MNs can get a channel when β=0.7 regardless of α. On the other hand, in order to obtain optimum QoS, the system performance depends also on α values for low values of β. The throughput results are shown in 3D in Figure 8b considering all possible contention probabilities. The results indicate the drop in the throughput of the given both traffic loads since more requests are removed from the system due to contention. In other words, the throughput decreases as MNs leave the system considering the contention probabilities. However, this improves the overall network performance as these requests can be dealt with by other networks.



(a) MRT vs α (b) 3D representation of MRT for λ=1.7

Figure 9: Mean response time results for all contention probabilities

The Figure 9a shows MRT results as a function of α considering various full contention probabilities occurring in the queue (β) for different traffic loads (e.g., λ=1, 1.7). As clearly seen from the figure, the best MRT results can be obtained for high value of β. When β=0.9, the all MNs can get a channel regardless of α. Even though the system has moderate traffic loads (e.g., λ=1) the full contention probability in the queue is important parameter of getting a resource in such system. This can be clearly seen in Figure 9b.

5.2 Probability of a request staying in the contention queue

In the queueing analysis presented in previous section, α and β values are assumed. In order to utilize the proposed proactive approach, a model has to be developed to obtain the probability of a request not been affected by α and β. It is known from the standard queueing analysis for

a system of k nodes:

$$P_n = \rho^n \frac{1-\rho}{1-\rho^{k+1}} \tag{13}$$

Where, ρ is the traffic intensity (λ/μ), k is the size of queue and n is the number of request in the system.

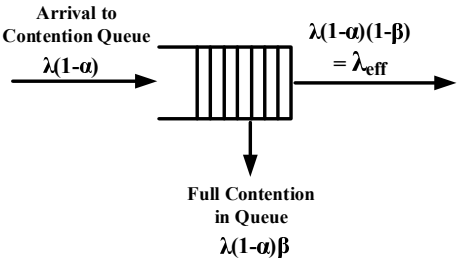


Figure 10: Proactive contention queue

It can be assumed that the proactive contention queue as a simple M/M/1/k queue. Here, we are interested only to find out the probability of a given request staying in the contention queue and not be affected due to β. Here the arrival rate to the contention queue is $\lambda(1-\alpha)$ and service rate is β as we are only interested in the requests affected by β as shown in Figure 10. Therefore, ρ can be represented as:

$$\rho = \frac{\lambda(1-\alpha)}{\beta} \tag{14}$$

Therefore,

$$P_{n-1} = \rho^{n-1} \frac{1-\rho}{1-\rho^{k+1}} \tag{15}$$

Here, the number of requests in the system becomes $n-1$ as there are no entries in the server. In order to find the probability of a given request staying in the contention queue without being affected by β, A request needs to satisfy three conditions:

- The new incoming request occupied the N^{th} position in the queue, i.e., last position of the queue.
- New incoming request is having a full contention with an entry at the N^{th} position and pushing that entry our of the queue.
- New request after occupying a place in the queue is not affected by the subsequent new incoming request.

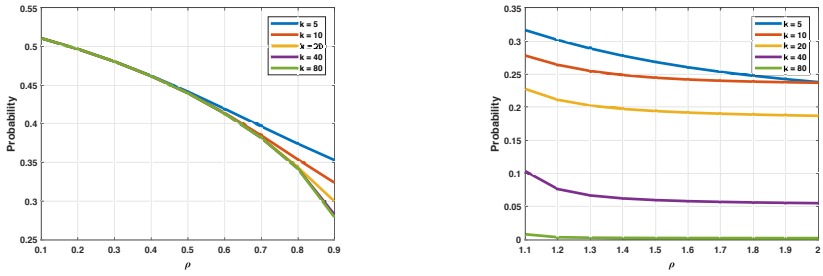
Here, let us assume that P^A_{Full} is the probability of full contention for incoming request A. P^B_{Full} is the probability of full contention for the requests in contention queue and here an average velocity is considered. P^C_{Full} is the probability of full contention for a new subsequent incoming request C after A.

- $(1-P^B_{Full})^{N-1}$ is the probability that request A occupies the N^{th} position in the contention queue and it is not colliding with any other request before it.
- $(1-P^A_{Full})^{N-1}P^A_{Full}$ is the probability that request A do not collide with the request in front of N^{th} position but does collide with the request already occupying the N^{th} position.
- $1-[(1-P^C_{Full})^{N-1}P^C_{Full}]$ is the probability of the request A not getting booted out by a new subsequent request entering the queue i,e C.

Therefore, all these three conditions can be represented as shown in Equation (16) to calculate the probability of a request staying in the queue and not been affected by β.

$$\sum_{n=1}^{n=k} \Big\{ \Big[\Big(\frac{(\rho^{n-1})(1-\rho)}{1-\rho^{k+1}} (1 - P_{Full}^{B})^{n-1} \Big) + \Big(\sum_{m=n}^{m=k} \frac{\rho^{m}(1-\rho)}{1-\rho^{k+1}} (1 - P_{Full}^{A})^{m-1} P_{Full}^{A} \Big) \Big] \times \Big(1 - (1 - P_{Full}^{C})^{n-1} P_{Full}^{C} \Big) \Big\} \times (1 - P_{Full}^{B}) \tag{16}$$

For example, let us consider $\mathbb{T}$ for MN_A is 10s and MN_B is 60s. $\mathbb{N}$ for MN_A is 20s and MN_B is 30s. The resulting probability of full contention based on our two node model: $P_{Full}^{A} = 0.085714$, $P_{Full}^{B} = 0.064286$ and let us assume $P_{Full}^{C} = 0.2$. Probability of a request not affected by β for $k = 1$ is 0.68635, $k = 20$ is 0.45405 and $N = 80$ is 0.75405. Calculating β accurately is very important so that the requests leaving contention queue due to full contention can be served by an alternative network. Since, β is dependent on the values of $\mathbb{N}$ and $\mathbb{T}$ of each request in the queue and new incoming request, therefore, a detailed analysis is required to accurately model β. By using this approach, it is possible to achieve seamless communication.



(a) $\rho = 0.1$ *to* 0.9 (b) $\rho = 1.1$ *to* 2.0

Figure 11: Probability of a request not being affected by β

Let us consider $\mathbb{T}_A$ and $\mathbb{T}_B$ are both 20s. MN_A is moving at a velocity of 30Mph and average velocity of the requests in the contention queue as 10Mph. Therefore, for network the P_{Full}^{A} is 0.094055 and P_{Full}^{B} which is the probability of full contention for the requests with average velocity in the queue is 0.346165. Let us assume P_{Full}^{C} as 0.2. Therefore, we can find the probability of request A staying in the queue using Equation (16) and the resulting graphs for different ρ values from 0.1 to 0.9 and ρ values from 1.1 to 2 are shown in Figure 11a and 11b, respectively. From the result we can observe that as ρ increases the probability of a request being affected by β, resulting in not reaching the channel queue increases. In other words, probability of a request not being affected by β deceases as shown in Figure 11. In addition, it is clearly seen from the Figures 11a and 11b that k affects the system performance significantly especially for highly loaded system.

6 Conclusion and future work

Analytical models for a proactive system where the users contend for resources with a thorough analysis has been presented. These models have yielded two key parameters, β and α, that have to be calculated to effectively use the proposed proactive approach. It has been shown that if these two parameters are identified then we can ensure that each users in the system will be effectively served by at least one of the available network. This can be achieved by using vertical handover to alternative network if an user is about to experience a full contention in the target network. To the best of our knowledge there is no study that has modelled a proactive model in

the context of users contending for resources based on their mobility. With the results showing that the proposed approach is outperforming the classical model. The application of the work and approach in a Middlesex VANET testbed is the future work.

Bibliography

[1] Alhabo, M.; Li, Z.; Naveed, N. (2017); A Trade-off Between Unnecessary Handover and Handover Failure for Heterogeneous Networks. *European Wireless 2017*, 167-172, 2017.

[2] Almulla, M.; Wang, Y.; Boukerche, A.; Zhang, Z. (2014); Design of a fast location-based handoff scheme for IEEE 802.11 vehicular networs, *IEEE Transactions on Vehicular Technology*, 63(8), 3853-3866, 2014.

[3] Bejaoui, T. (2014); QoS-oriented high dynamic resource allocation in vehicular communication networks, *The Scientific World Journal*, Article ID 718698, 9 pages, 2014. http://dx.doi.org/10.1155/2014/718698

[4] Chen, X.; Ni, M. (2016); Seamless handover for high mobility environments, *International Wireless Communications and Mobile Computing Conference (IWCMC)*, 281-286, 2016.

[5] Elhadj, H. B.; Elias, J.; Chaari, L.; Kamoun, L. (2016); Multi-attribute decision making handover algorithm for wireless body area networks. *Comput. Commun.*, 81, 97-108, 2016.

[6] Feirer, S.; Sauter, T. (2017); Seamless handover in industrial WLAN using IEEE 802.11k, *IEEE 26th International Symposium on Industrial Electronics (ISIE)*, 1234-1239, 2017.

[7] Fernandez, P. J.; Santa, J.; PerenÄąguez, F.; Skarmeta, A. F. (2017); Towards seamless inter-technology handovers in vehicular fIPv6g communications, *Computer Standards and Interfaces*, 52, 85-96, 2017.

[8] Huang, Q.; Huang, Y.-C.; Ko, K.T.; Iversen, V.B. (2011); Loss performance modeling for hierarchical heterogeneous wireless networks with speed-sensitive call admission control, *IEEE Transactions on Vehicular Technology*, 60(5), 2209-2223, 2011.

[9] Kirsal, Y.; Paranthaman V.V.; Mapp,G. (2018); Exploring analytical models to maintain quality-of-service for resource management using a proactive approach in highly mobile environments, *2018 7th International Conference on Computers Communications and Control (ICCCC), Proc. of,* Oradea, Romania, May 2018, Publisher: IEEE, 176-182, 2018. DOI: 10.1109/ICCCC.2018.8390456

[10] Kirsal, Y. (2016); Analytical modelling of a new handover algorithm for improve allocation of resources in highly mobile environments, *International Journal of Computers Communications & Control*, 11(6), 789-803, 2016.

[11] Li, J.; Luo, T.; Ding, L. (2014); A location aware based handoff algorithm in v2i system of railway environment. *Global Information Infrastructure and Networking Symposium (GIIS)*, 1-6, 2014.

[12] Mapp, G.; Katsriku, F.; Aiash, M.; Chinnam, N.; Lopes, R.; Moreira, E.; Vanni R.M.P.; Augusto, M. (2012); Exploiting location and contextual information to develop a comprehensive framework for proactive handover in heterogeneous environments. *Journal of Computer Networks and Communications*, 1-17, 2012.

[13] Mapp, G. E.; Cottingham, D.; Shaikh, F.; Vidales, P.; Patanapongpibul, L., Baliosian, J.; Crowcroft, J. (2006); An architectural framework for heterogeneous networking, *International Conference on Wireless Information Networks and Systems*, 2006.

[14] Nadembega, A.; Hafid, A.; Taleb, T. (2015); A destination and mobility path prediction scheme for mobile networks, *IEEE transactions on vehicular technology*, 64(6), 2577-2590, 2015.

[15] Paranthaman, V.V.; Ghosh, A.; Mapp, G.; Iniovosa, V.; Shah, P.; Nguyen, H. X.; Gemikonakli, O.; Rahman, S. (2017); Building a Prototype VANET Testbed to Explore Communication Dynamics in Highly Mobile Environments, *Springer International Publishing*, 81-90, 2017.

[16] Paranthaman, V. V.; Kirsal, Y., Mapp, G.; Shah, P.; Nguyen, H. X. (2017); Exploring a New Proactive Algorithm for Resource Management and Its Application to Wireless Mobile Environments. *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, 539-542, 2017.

[17] Saxena, N.; Roy, A. (2009); Experimental framework of proactive handover with qos over wlans. *Electronics Letters*, 45(25), 1313-1315, 2009.

[18] Shaikh, F.;, Mapp, G. E.; Lasebae, A. (2007); Proactive policy management using tbvh mechanism in heterogeneous networks, *International Conference on Next Generation Mobile Applications, Services and Technologies*, 151-157, 2007.

[19] Trivedi, K. S.; Dharmaraja, S.; Ma, X. (2002); Analytic modeling of handoffs in wireless cellular networks, *Information sciences*, 148(1), 155-166, 2002.

[20] Vidales, P.; Baliosian, J.; Serrat, J.; Mapp, G.; Stajano, F.; Hopper, A. (2005); Autonomic system for mobility support in 4g networks, *IEEE Journal on Selected Areas in Communications*, 23 (12), 2288-2304, 2005.

[21] Vilaplana, J., Solsona, F.; Teixido, I.; Mateo, J.; Abella, F.; Rius, J. (2014); A queuing theory model for cloud computing. *J. Supercomput.*, 69(1), 492-507, 2014.

[22] Wang, W.; Xu, H.; Zhou, H. (2014); Proactive unnecessary handover avoidance scheme in LTE-A small cells, *IEEE Wireless Communications and Networking Conference (WCNC)*, Istanbul, 2214-2218, 2014.

[23] Wu, S.J. (2010); An intelligent handover decision mechanism for heterogeneous wireless networks, *The 6th International Conference on Networked Computing and Advanced Information Management*, Seoul, 688-693, 2010.

[24] Xenakis, D.; Merakos, L.; Passas, N.; Verikoukis, C. (2016); Handover Decision for Small Cells: Algorithms, Lessons Learned and Simulation Study. *Communication Networks*, Elsevier, 100, 64-74, 2016.

[25] Zeng, Q.A.; Agrawal, D.P. (2001); Modeling of handoffs and performance analysis of wireless data networks, *International Conference on Parallel Processing Workshops*, 491-496, 2001.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 853-864, October 2018.

Comparative Analysis of Various Transformation Techniques for Voiceless Consonants Modeling

G. Korvel, B. Kostek, O. Kurasova

Grazina Korvel*, Olga Kurasova
Institute of Data Science and Digital Technologies, Vilnius University
Akademijos str. 4, LT-04812, Vilnius, Lithuania
*Corresponding author: grazina.korvel@mii.vu.lt
olga.kurasova@mii.vu.lt

Bozena Kostek
Audio Acoustics Laboratory, Faculty of Electronics, Telecommunications and Informatics,
Gdansk University of Technology
G. Narutowicza 11/12, 80-233 Gdansk, Poland
bokostek@audioacoustics.org

Abstract: In this paper, a comparison of various transformation techniques, namely Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Walsh Hadamard Transform (DWHT) are performed in the context of their application to voiceless consonant modeling. Speech features based on these transformation techniques are extracted. These features are mean and derivative values of cepstrum coefficients, derived from each transformation. Feature extraction is performed on the speech signal divided into short-time segments. The kNN and Naive Bayes methods are used for phoneme classification. We consider both classfication accuracies and computational time. Experiments show that DFT and DCT give better classification accuracy than DWHT. The result of DFT was not significantly different from DCT, but it was for DWHT. The same tendency was revealed for DCT. It was checked with the usage of the ANOVA test that the difference between results obtained by DCT and DWHT is significant.
Keywords: Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Discrete Walsh Hadamard Transform (DWHT), cepstrum coefficients.

1 Introduction

The state-of-the-art methods applied to speech technology are mostly based on the extraction of features and machine learning. A wide range of speech signal features was conceived and used for classification tasks [19], speech recognition [9], emotional speech recognition [17, 28], phoneme modeling [10], and speech analytics tasks [2]. There are also other approaches employed for processing speech signals, where feature extraction process is discarded. For example, the use of fuzzy logic [29, 30] applied to speech technology, specifically to voice activity detection (VAD), speech segmentation, and coding, cannot be disregarded in this aspect. Moreover, very intense activities connected to the usage of resources for large-scale deep learning analysis applied to speech recognition or emotional speech recognition may be observed in the last few years [15, 24, 32]. However, when we are talking about speech analytics and modeling, speech synthesis or audio-visual speech recognition, the progress in these fields is below expectations. Secondly, this research area requires a different approach, a thorough analysis of individual spoken elements needs to be performed as there is basic knowledge still missing in this context.

The phoneme mathematical models, utilized as tools for describing speech, are of great importance not only for speech synthesis. The need for research on phoneme models of speech is justified by its numerous possible uses. The following can be named: speech recognition, helping

with pronunciation and learning foreign languages (a comparison of phoneme utterances and its model enabling to demonstrate differences of pronunciation), studies in linguistics, medical field (e.g. disturbances in speech present in stroke and neurodegenerative diseases, disorder in one or more prosodic functions, deficits in speech production, etc.). In some of the envisioned applications the obtained results can be a part of a larger multimodal Human-Computer Interaction system consisting of three modalities: vocal, facial and gesture based recognition.

The object of this research is the consonant phoneme signals which are more difficult for analysis, modeling and classification tasks as those of vowels and semivowels. The character of the consonant signals is consonant-dependent and varying. Stop consonants can be considered as quasiperiodic signals in noise, while fricative consonants as aperiodic signals. We can also divide those phonemes into two sets: voiced and voiceless sounds [6]. This means that the vocal folds are apart while saying these sounds. In speech processing, sounds can be represented as a source-filter model [22]. The filter represents the vocal tract, which is excited with a source. A source is a pulse sequence for the voiced sounds and noise for the unvoiced sounds. A commonly used technique for separating source and filter in a speech signal is cepstral analysis [14]. The cepstrum is widely used in speech processing [8, 16].

In all the areas mentioned above automatic feedback for systems and applications is also of importance, thus in a given methodology both feature extraction and machine learning should be applied. To create a mathematical model of a phoneme, it is important to find a suitable parametric description of speech. The speech signal is converted to the appropriate space domain and preprocessing is carried out. The two main domains of analysis are time and frequency. The first of them shows the time varying character of the signal, the second mirrors how the energy of the signal is contained within the frequency range. In the frequency domain, parameters are often based on the Fourier spectrum. In this paper, we perform a comparative experiment based on DFT (Discrete Fourier Transform), DCT (Discrete Cosine Transform) and DWHT (Discrete Walsh Hadamard Transform). The results returned by this study should enable us to verify which transformation method along with feature extraction work better when such a methodology is applied to check phoneme modeling precision.

The relationship between the performance of transformation techniques in the context of feature vectors derivation has been investigated by many researchers, in various speech classification tasks. In the paper of Velican et al. [33], a comparison of DWHT and DCT as feature selection tools in the case of identifying rhotacism is performed. The experiment result shows that classification rate in the case of DWHT is better than the rate obtained with DCT. In the paper of Kekre and Kulkarni [7] a comparison of the performance DCT and DWHT for various feature vector sizes with and without overlap based on speech utterances is given for speaker identification. The results show that DCT performs better than DWHT. The comparison of two fundamentally different approaches the Fast Fourier Transform (FFT) and Hilbert-Huang Transform (HHT) is given in paper of Donnelly [4]. The behavior and flexibility of these two transforms are examined for a number of different time domain signal types.

The targeted consonant phonemes are also more susceptible to noise than vowels mainly due to their lower intensity. This means that in many conditions they may easily be masked by signals interfering with speech. That is why it is important to find optimized feature vectors that will perform in both quiet and noise conditions. This paper deals with a domain-dependent analysis and classification of consonant phonemes utilizing the cepstrum analysis. The feature vectors consist of cepstrum coefficients derived from the Fourier, Cosine and Walsh Hadamard transforms.

2 Transformation techniques

Let $x(k)$ is a signal with length K, where K is an integer power of 2 ($k = 1, \ldots, K$). The frequency domain representation shows how the energy of this signal is contained within the frequency range. The techniques of signal transformation from time to the frequency domain are given in this section.

2.1 Discrete Fourier transform

Fourier analysis is based on decomposing signals into sinusoids [26]. DFT is a family member of this analysis used with digital signals. The transform decomposes the signal $x(k)$ into the sequence of complex numbers $y(1), ..., y(K)$ according to the formula:

$$y(k) = \sum_{n=1}^{K} x(n) e^{\frac{(-2\pi j)}{K}(n-1)(k-1)} \tag{1}$$

where the symbol j denotes the imaginary unit.

To convert signal data from the frequency to the time domain the Inverse Discrete Fourier Transform (IDFT) is applied. The IDFT is defined as follows:

$$x(k) = \frac{1}{K} \sum_{n=1}^{K} y(n) e^{\frac{2\pi j}{K}(n-1)(k-1)} \tag{2}$$

The result of IDFT will be used in the construction of the signal cepstrum.

2.2 Discrete Walsh-Hadamard transform

DWHT is a non-sinusoidal technique that represents a signal as a set of orthogonal rectangular waveforms. The transform is given by the formula:

$$y(k) = \sum_{n=1}^{K} x(n) W_K(k, n) \tag{3}$$

The basis function is described as follows:

$$W_K(k, n) = \prod_{l=1}^{L-1} (-1)^{n_l k_{M-1-l}} \tag{4}$$

where $L = \log_2 K$, n_l is the l^{th} bit in the binary representation of n [27].

As we see from Eq. (4), DWHT takes the binary value 1 or -1. The Inverse Discrete Walsh Hadamard Transform (IDWHT) is defined as follows:

$$y(k) = \frac{1}{K} \sum_{n=1}^{K} x(n) W_K(k, n) \tag{5}$$

The only difference between DWHT and IDWHT (see Eq. (3) and Eq. (5)) is a constant divisor.

2.3 Discrete Cosine transform

DCT decomposes a signal into cosine functions. The transformation has several standard variants. These variants and the mathematical properties of DCT are presented in works of Rao and Yip, Oppenheim *et al.* [18, 23]. In this paper, the Discrete Cosine transform of the signal $x(k)$ is computed according to the formula:

$$y(k) = \sqrt{\frac{2}{K}}\beta(k)\sum_{n=1}^{K} x(n)cos(\frac{\pi(2n-1)(k-1)}{2K}) \tag{6}$$

where coefficient $\beta(k)$ is defined as follows:

$$\beta(k) = \begin{cases} \frac{1}{\sqrt{2}}, & if \quad k = 1 \\ 1, & if \quad k \neq 1 \end{cases} \tag{7}$$

The formula of the Inverse Discrete Cosine Transform (IDCT) is given below:

$$x(k) = \sqrt{\frac{2}{K}}\sum_{n=1}^{K} \beta(k)y(n)cos(\frac{\pi(n-1)(2k-1)}{2K}) \tag{8}$$

DCT and other transformation techniques analyzed in this Section are orthogonal transforms. Therefore, they can be computed using the fast algorithms.

3 Feature extraction

All the N samples of the analyzed phoneme are collected into a vector:

$$x = \left[\begin{array}{cccc} \text{x(1)}, & \text{x(2)}, \ldots, & \text{x(N)} \end{array} \right] \tag{9}$$

The phoneme signal is divided into short-time frames, the length of which is M samples. A process of dividing a signal into frames is typical for the speech signal analysis. To each of these frames, a window function $w(n)$ is used. Due to the window procedure, a part of the signal data is lost. Therefore, an overlap of segments is utilized. How much should the segments overlap can be seen in [5].

Denote by L the number of the overlapped samples. Then the number of intervals can be obtained by the following formula:

$$P = \left[\frac{N-M}{M-L}\right] + 1 \tag{10}$$

where $[\alpha]$ stands for an integer part of the real number.

Then the phoneme signal can be written as the following matrix:

$$X = \begin{bmatrix} w(1) \times x(1) & \ldots & w(M) \times x(M) \\ w(1) \times x(M-L+1) & \ldots & w(M) \times x(2M-L) \\ \ldots & \ldots & \ldots \\ w(1) \times x\Big((P-1) \times (M-L+1)\Big) & \ldots & w(M) \times x\Big((P-1) \times (2M-L)\Big) \end{bmatrix} \tag{11}$$

The calculation procedure of the cepstrum coefficients constitutes a part of the algorithm prepared. The consecutive steps of the algorithm are listed below:

Step 1. The selected transform is applied to each row of the matrix X.

Step 2. The absolute values are taken out.
Step 3. The logarithm is calculated.
Step 4. The inverse transform is applied.
Consequently, we obtain the matrix of the cepstrum coefficients:

$$C = \begin{bmatrix} c_{11} & \dots & c_{1M} \\ c_{21} & \dots & c_{2M} \\ \dots & \dots & \dots \\ c_{k1} & \dots & c_{kM} \end{bmatrix} \tag{12}$$

The mean values of the columns of the matrix C are calculated. All the obtained values are collected into a vector c:

$$c = \begin{bmatrix} \text{c(1)}, & \text{c(2)}, \dots, & \text{c(M)} \end{bmatrix} \tag{13}$$

The mean cepstrum values given in Eq. (13) are used as representative features.

In order to determine whether a function is increasing or decreasing, additionally the first-order delta derivatives are calculated. The first-order dynamic coefficients are calculated from the static cepstrum coefficients using the following regression formula:

$$d_m = \frac{\sum_{n=1}^{N} n(c_{ex}(m+n) - c_{ex}(m-n))}{2\sum_{n=1}^{N} n^2} \tag{14}$$

where

$$c_{ex} = \begin{bmatrix} \underbrace{0, \dots, 0}_{M}, & c(1), & \dots, & c(M), & \underbrace{0, \dots, 0}_{M} \end{bmatrix} \tag{15}$$

$m = 1...M$, N is the regression window size.

4 Classification methods

A vast literature on the application of machine learning to the classification task exist. Researchers developed many approaches to the problem of classification, including methods for inducing rule sets, models in the form of a tree structure, linear discriminants, statistical learning algorithms, and artificial neural networks [24]. In an experiment, we use two classical machine learning algorithms to compare classification rates. First of them is the Naive Bayes classification method, based on Bayes theory [11]. This algorithm is widely used because it often outperforms more sophisticated classification methods. It falls into the statistical learning algorithms and provides the probability of each attribute set belonging to a given class.

The second classification algorithm used in this experiment is k-Nearest Neighbors (kNN), based on Euclidean distances between the elements of the test dataset and elements of the training dataset [13]. The number of nearest neighbors is set by performing preliminary tests.

5 Experimental results

The experiment was performed on Lithuanian speech recordings, created during the project LIEPA (Services controlled by the Lithuanian Speech) [20]. The database consists of 100 hours of words, phrases and sentences recordings, different speakers, both male and female voices and is adapted for scientific research. In the present experiment, we consider the extracted consonant phonemes from this database for our analysis. The phonemes are the following: /t/, /k/, /s/, /ʃ/. The first two (/t/ and /k/) are called stop consonants because the air in the vocal tract is stopped at some period. An example of the phoneme /t/ signal is given in Figure 1.

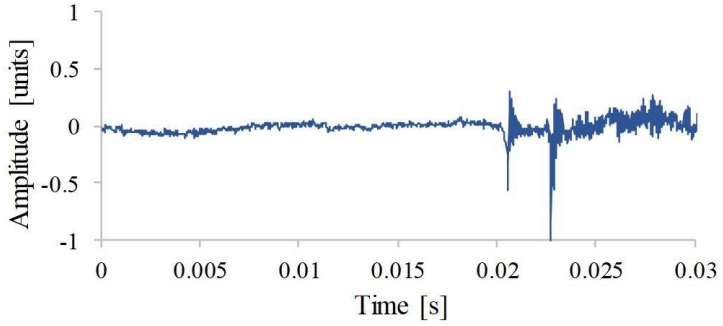


Figure 1: The plot of the consonant phoneme /t/

The next two phonemes (/s/ and /ʃ/) are called fricative consonants, which are produced when the air is squeezed out through a small hole in the mouth. An example of a phoneme /ʃ/ signal is given in Figure 2. The audio data used in the analysis are wav files with the following

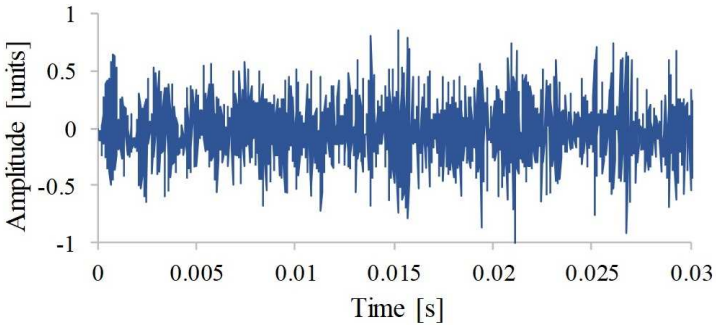


Figure 2: The plot of the consonant phoneme /ʃ/

parameters: sampling frequency: 22 kHz, quantification: 16 bits, the number of channels is 1.

The feature extraction procedure proposed in Section 3 involves several steps. First of all, the signal pre-processing is carrying out. Then the signal is converted to the appropriate space domain and the extraction of features is performed. In this experiment, signal pre-processing is performed using the following parameters: the input signal is divided into frames of 512 samples, and then for each frame, the Hamming window is chosen. The overlap of 50% is used. Therefore, the number of cepstrum coefficients is equal to the number of coefficients of transformation (i.e. 512). An observation reveals that only part of them is useful for separation of the consonant classes. That is why only the first 12 coefficients were selected as representative features. In Figure 3, DFT cepstrum is shown. It can be seen from Figure 3 that the cepstrum coefficients present differences between consonant classes, this is especially visible in the case of the first four coefficients.

It was checked in further analyses that cepstra of both DWHT and DCT followed the same trend. The plots of these cepstra are shown in Figures 4 (DWHT-based) and 5 (DCT-based).

Additional 12 features are derived from computing the first order derivatives.

In the experiment, 480 utterances (120 for each phoneme) were considered. These phonemes were cut out of the recordings of 15 speakers (9 females and 7 males). We extracted parameters for all these phonemes. The data are divided into two segments: one employed to teach a model and another one utilized to test this model. The test set for models is constructed of 10% randomly selected phonemes.

Due to the fact that the set of samples is not very big, and it is important to estimate the true error rate of a given classifier, an experiment was repeated 50 times for each case and the

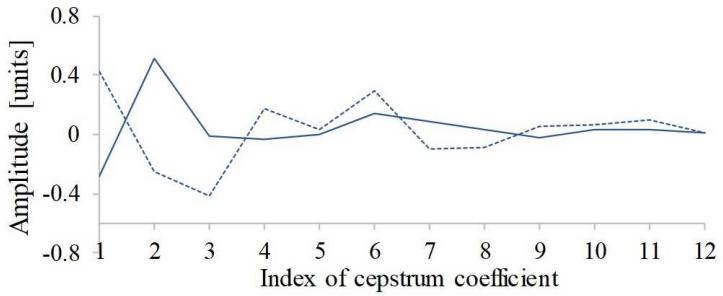


Figure 3: The cepstrum coefficients obtained by DFT (the solid line–consonant phoneme /t/, the dotted line–consonant phoneme /ʃ/)

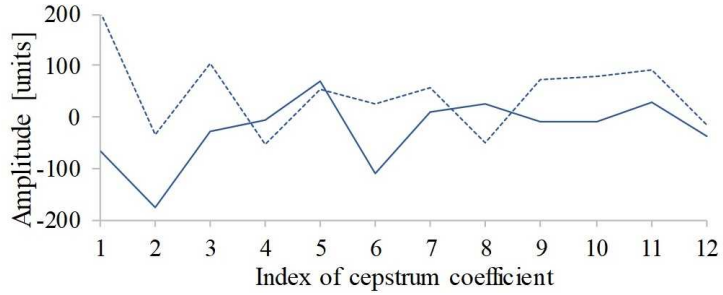


Figure 4: The cepstrum coefficients obtained by DWHT (the solid line–consonant phoneme /t/, the dotted line–consonant phoneme /ʃ/)

arithmetic mean was calculated. A comparison of the performance of two selected classification methods averaged for all speakers, males and females separately is given in Table 1.

In order to determine whether the differences between the means of the three parametrization techniques are statistically significant, the one-way analysis of variance (ANOVA) test is used. The test significance level α equals to 0.05. We state a null hypothesis (H_0) that in each case both samples are from populations with the same means. The decision rule to reject this hypothesis is as follows:

$$reject\ H_0\ if\ F > F_{critical}(1-\alpha) \tag{16}$$

where F is the test statistic calculated as the ratio of the difference between the means over a distribution of their data points, and $F_{critical}$ is the critical value taken from the F distribution table [3]. The results of ANOVA test are given in Table 2.

The obtained F values (see Table 2) are compared with the critical value for F distribution.

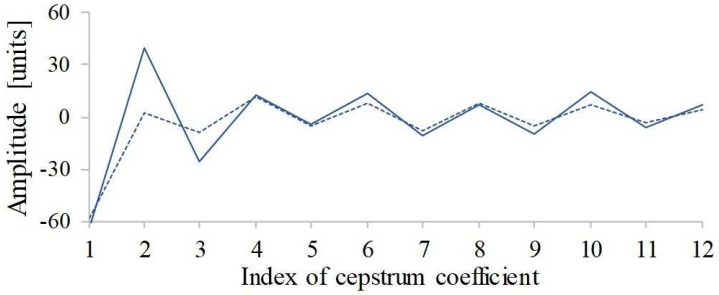


Figure 5: The cepstrum coefficients obtained by DCT (the solid line–consonant phoneme /t/, the dotted line–consonant phoneme /ʃ/)

Table 1: The classification accuracy [%] for 4 consonant classes

		k-Nearest Neighbors			Naive Bayes		
		DWHT	DFT	DCT	DWHT	DFT	DCT
All	Mean	82.67	85.13	**86.04**	87.00	**90.50**	89.42
	Std.Dev.	5.97	4.89	4.44	4.16	4.00	4.41
Male	Mean	86.71	90.71	**90.86**	96.29	98.14	**98.29**
	Std.Dev.	11.55	7.53	8.29	5.25	3.77	3.40
Female	Mean	76.94	79.56	**81.88**	84.38	87.06	**87.81**
	Std.Dev.	6.05	6.35	6.19	4.77	4.93	4.86

Table 2: The result of ANOVA test for kNN and Naive Bayes

		k-Nearest Neighbors			Naive Bayes		
		DWHT/ DFT	DWHT/ DCT	DFT/ DCT	DWHT/ DFT	DWHT/ DCT	DFT/ DCT
All	F-value	**5.07**	**10.28**	0.963	**18.42**	**7.95**	1.66
	p-value	0.026581	0.001815	0.328837	0.000042	0.005826	0.201148
Male	F-value	**4.21**	**4.25**	0.008	**4.13**	**5.11**	0.039
	p-value	0.042865	0.041988	0.928349	0.044831	0.026022	0.842655
Female	F-value	**4.48**	**16.28**	3.40	**7.68**	**12.75**	0.59
	p-value	0.036904	0.000109	0.068158	0.006691	0.000553	0.445528

In the experiments performed, the obtained F is significant at a given level if it is equal to or greater than 4.03 ($F_{critical} = 4.03$). According to these assumptions, the differences between DWHT and DFT as well as differences between DWHT and DCT are statistically significant. Meanwhile, the differences between DFT and DCT are not statistically significant.

The experiments were performed using MATLAB on a Laptop with IntelR CoreTM i5-6200U 2.20 GHz CPU and 8 GB of RAM. The computation time is given in Table 3. From this table we see, that the computational time of kNN is much smaller than Naive Bayes.

We also compare the results obtained on Lithuanian consonants with the classification effectiveness collected from the literature for other languages (see Table 4). Obviously, such a comparison cannot be performed straightforward as the studies recalled here concern different languages and also a variety of features and classification methods as well as they are researched for different purposes (e.g. speech recognition, clean and telephone speech differentiation, speech productionÂ models and mechanisms, pathology disorder, etc.). Thus data contained in Table 4 may serve only to a limited extent when comparing algorithmic performances.

Though different classification methods are employed in the studies recalled, we see that our results are consistent with the results of other researchers, however they are dependent more on the vector feature content than on the type of a classifier.

Table 3: Computational time [s] for the classifiers

	k-Nearest Neighbors			Naive Bayes		
	DWHT	DFT	DCT	DWHT	DFT	DCT
All	0.1464	0.1456	0.1457	2.5672	2.5742	2.5599
Male	0.0270	0.0270	0.0270	1.8016	1.7982	1.8021
Female	0.0908	0.0902	0.0899	2.2227	2.1915	2.1928

Table 4: Consonants classification performance in literature

Reference	**Dataset**	**Parameters**	**Classification technique**	**Overall classification accuracy**
Thasleema and Narayanan, 2018 [31]	Malayalam (India) (unaspirated, aspirated, nasal, approximants, fricatives)	Normalized Wavelet Hybrid Feature (NWHF) vector based on Wavelet Transform	k-Nearest Neighbors (kNN), Artificial Neural Network (ANN), Support Vector Machine (SVM)	From 34.2% to 60.2% for kNN, from 45.9% to 63.7% for ANN and 55.4% to 79.9% for SVM (depending on the mother wavelet)
Korvel and Kostek, 2017 [9]	MODALITY database (English stop consonants)	Descriptors coming from music information retrieval	k-Nearest Neighbors (kNN)	73%
Lee and Choi, 2012 [12]	TIMIT database (American English)	Mel-frequency cepstral coefficients (MFCCs), first and second derivatives plus acoustic parameters such as band-limited RMS energy, amplitude of the first harmonic and peak normalized cross correlation values (PNCC)	Gaussian mixture models (GMMs)	Depending on the type of consonants, i.e.: stops, fricatives, and, affricates classification accuracies are as follows: 82.2%, 80.6%, and 78.4% respectively
Ali et al., 2001 [1]	American English stop consonants	The acoustic-phonetic characteristics	The authors proposed classification system combining the voicing detection and the place of articulation detection	86%
Pruthi and Espy-Wilson, 2003 [21]	TIMIT database (Nasals and semivowels)	The acoustic parameters which include F1 measure, a pick onset/offset measure, an energy ratio, and a formant density measure	Support Vector Machine (SVM) based classifiers	Accuracies of 88.6%, 94.9% and 85.0% were obtained for prevocalic, postvocalic and intervocalic sonorant consonants, respectively

6 Conclusions

In the paper, we have compared the performance of DFT, DWHT and DCT for voiceless consonant (/t/, /k/, /s/, /ʃ/) classification. In order to evaluate the classification accuracy, two methods, namely kNN and Naive Bayes were used. The analyses were performed for the whole group of speakers, and for male and female speakers separately. The highest classification accuracy for all speakers (86.04%) has been achieved for features based on DCT technique, in the case of kNN method. While for Naive Bayes classifier, the highest accuracy for all speakers was equal to 90.50% for DFT. In the cases of the analysis of male and female recordings separately, the highest accuracies have been achieved for features based on DCT technique for both classifiers. These accuracies are as follows: for kNN classifier the highest accuracy for male group was equal to 90.86%, for female group – 98.29%, while for Naive Bayes classifier the highest accuracy for male group was equal to 81.88%, for female group – 87.81%.

The employment of one-way analysis of variance (ANOVA) test for results of both selected classification methods revealed the same tendency for different groups of speakers and different classifiers: the difference between results, obtained by DFT and DCT is not significant, meanwhile difference between results, obtained by DWHT and the other two transformations (DFT and DCT) is significant.

A comparison of the results obtained on Lithuanian consonants with other results in the literature was also performed. A literature review shows, that our results are consistent with those of other researchers.

It is important to mention that our primary intention was not to obtain high classification accuracy, but the goal was to determine which transformation method returns better results when applying a given feature vector and a regular machine learning algorithm. This is important in the context of the feedback needed on phoneme modeling precision to verify the model consistency with the initial phoneme target. As seen from observations the created feature vector is not complete as the accuracy obtained is not fully satisfying. Therefore, in the future research, we will investigate the possibility of extending the created feature vector with additional signal descriptors applicable to short-segmented speech units. In addition, more effective classification algorithms based on the weighted features are to be considered. Finally, some additional tests should be executed on the same feature vectors but taking into account the phoneme neighbors and also presence of noise.

Funding

This research is funded by the European Social Fund under the No 09.3.3-LMT-K-712 "Development of Competences of Scientists, other Researchers and Students through Practical Research Activities" measure.

Bibliography

[1] Ali, A. M. A.; Van der Spiegel, J.; Mueller, P. (2001); Acoustic-Phonetic Features for the Automatic Classification of Stop Consonants, *IEEE Transactions on Speech and Audio Processing*, 9(8), 833–841, 2001.

[2] Czyzewski, A.; Piotrowska, M.; Kostek B. (2017); Analysis of Allophones Based on Audio Signal Recordings and Parameterization, *The Journal of the Acoustical Society of America*, 141 (5), 3521–3521, 2017.

[3] De Muth, J. E. (2014); *Basic Statistics and Pharmaceutical Statistical Applications*, 3rd edn, CRC Press, 2014.

[4] Donnelly, D. (2006); The Fast Fourier and Hilbert-Huang Transforms: A Comparison, *International Journal of Computers Communications & Control*, 1 (4), 45–52, 2006.

[5] Heinzel, G.; Rudiger; A., Schilling, R, (2002); Spectrum and Spectral Density Estimation by the Discrete Fourier Transform (DFT), Including a Comprehensive List of Window Functions and Some New Flat-top Windows, *Internal Report, Max-Planck-Institut fur Gravitationsphysik, Hannover*, 2002.

[6] Kasparaitis, P. (2005); Diphone Databases for Lithuanian Text-to-speech Synthesis. *Informatica*, 193–202, 2005.

[7] Kekre, H. B., Kulkarni, V. (2011); Speaker Identification using Row Mean of DCT and Walsh Hadamard Transform, *International Journal on Computer Science and Engineering*, 3(3), 1295–1301, 2011

[8] Kim C.; Stern R. M. (2016); Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition, *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 24(7), 1315–1329, 2016.

[9] Korvel, G.; Kostek, B. (2017); Examining Feature Vector for Phoneme Recognition, *Proceeding of IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2017*, Bilbao, Spain, 394–398, 2017.

[10] Korvel, G.; Kostek, B. (2017); Voiceless Stop Consonant Modelling and Synthesis Framework Based on MISO Dynamic System, *Archives of Acoustics*, 3, 42, 375–383, 2017.

[11] Kotsiantis, S. B. (2007); Supervised Machine Learning: A Review of Classification Techniques, *Informatica*, 31(3), 249–268, 2007.

[12] Lee, S. M.; Choi J. Y.(2012); Analysis of Acoustic Parameters for Consonant Voicing Classification in Vlean and Telephone Speech, *The Journal of the Acoustical Society of America*, 131, EL197 (2012); doi: 10.1121/1.3678667

[13] Manocha S.; Girolami M. A. (2007); An Empirical Analysis of the Probabilistic K-nearest Neighbour Classifier, *Pattern Recognition Letters*, 28, 1818–1824, 2007.

[14] Milner, B.; Shao X. (2002); Speech Reconstruction from Mel-Frequency Cepstral Coefficients using a Source-Filter Model, *7th International Conference on Spoken Language Processing*, Denver, Colorado, USA, 2421–2424, 2002.

[15] Mitra V.; Sivaraman G.; Nam H.; Espy-Wilson C.; Saltzman E.; Tiede M. (2017); Hybrid Convolutional Neural Networks for Articulatory and Acoustic Information Based Speech Recognition, *Speech Communication*, 89, 103-112, 2017.

[16] Mitra, V.; Franco, H.; Graciarena, M.; Vergyri D. (2014); Medium-Duration Modulation Cepstral Feature for Robust Speech Recognition., *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1749–1753, 2014.

[17] Noroozi, F.; Kaminska, D.; Sapinski, T.; Anbarjafari, G. (2017); Supervised Vocal-Based Emotion Recognition Using Multiclass Support Vector Machine, Random Forests, and Adaboost, *Journal of the Audio Engineering Society*, 65(7/8), 562–572, 2017.

[18] Oppenheim, A. V.; Schafer, R. W.; Buck, J. R. (1999); *Prentice-Hall Signal Processing Series Discrete-Time Signal Processing*, 2nd edn. Prentice Hall, Inc., New Jersey, 1999.

[19] Pravin, S. C.; Anjana, R.; Pandiyan, T. P.; Ranganath, S. K.; Rangarajan P. (2017); ANN Based Disfluent Speech Classification, *Artificial Intelligent Systems and Machine Learning*, 9(4), 77-80, 2017.

[20] Project LIEPA Homepage, https://www.rastija.lt/liepa/about-project-liepa/7596, accessed on 2018/03/02.

[21] Pruthi T.; Espy-Wilson C. (2003); Automatic Classification of Nasals and Semivowels, *ICPhS 2003-15th International Congress of Phonetic Sciences*, 3061–3064, 2003

[22] Pyz, G.; Simonyte, V.; Slivinskas, V. (2014); Developing Models of Lithuanian Speech Vowels and Semivowels, *Informatica*, 25 (1), 55–72, 2014.

[23] Rao, K. R.; Yip, P. (1990); *Discrete Cosine Transform: Algorithms, Advantages, Applications*, 1st edn, Academic Press, 1990.

[24] Ravanelli M.; Brakel P.; Omologo M.; Bengio Y. (2017); A Network of Deep Neural Networks for Distant Speech Recognition, *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4880–4884, 2017.

[25] Sammut C.; Webb G. I. (2011); *Encyclopedia of Machine Learning. Springer Science & Business Media*, Springer New York, 2011.

[26] Smith, S. W. (1999); *The Scientist and Engineer's Guide to Digital Signal Processing*, 2nd edn. California Technical Publishing, San Diego, California, 1999.

[27] Sundararajan, D. (2001); *The Discrete Fourier Transform - Theory, Algorithms and Applications*, World Scientific, 2001.

[28] Tamulevicius, G.; Liogiene, T. (2015); Low-Order Multi-Level Features for Speech Emotion Recognition, *Baltic Journal of Modern Computing*, 4(3), 234–247, 2015.

[29] Teodorescu H.N.L. (2015), A Retrospective Assessment of Fuzzy Logic Applications in Voice Communications and Speech Analytics, *International Journal of Computers Communications & Control*, 10 (6), 105–112, 2015.

[30] Teodorescu H.N.L. (2015); Fuzzy Logic in Speech Technology-Introductory and Overviewing Glimpses, *Fifty Years of Fuzzy Logic and its Applications*, 581–608, 2015.

[31] Thasleema T. M.; Narayanan N. K.: Consonant Classification using Decision Directed Acyclic Graph Support Vector Machine Algorithm, *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 6(1), 59–74, 2013.

[32] Tzinis E.; Potamianos A. (2017); Segment-Based Speech Emotion Recognition using Recurrent Neural Networks, *Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, 190–195, 2017.

[33] Velican, V.; Strungaru, R.; Grigore, O. (2012); Automatic Recognition of Improperly Pronounced Initial 'r' Consonant in Romanian, *Advances in Electrical and Computer Engineering*, 12 (3), 79-84, 2012.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 865-880, October 2018.

Running Cells with Decision-Making Mechanism: Intelligence Decision P System for Evacuation Simulation

Y. Niu, Y. Zhang, J. Zhang, J. Xiao

Yunyun Niu, Yongpeng Zhang, Jieqiong Zhang
School of Information Engineering
China University of Geosciences in Beijing,
Beijing 100083, China
yniu@cugb.edu.cn, zyp940701@163.com, jieqiong1755@163.com

Jianhua Xiao*
The Research Center of Logistics
Nankai University,
Tianjin 300071, China
jhxiao2008@163.com
*Corresponding author: jhxiao2008@163.com

Abstract: Cell migration is a central process which happens along with multicellular organisms' development and maintenance. The process that cells move to specific locations in particular directions has some similarities with pedestrian walking behaviour. In this work, we propose a simulation model called an Intelligence Decision P System (IDPS), which is inspired by the process of cell migration. Each cell has its own decision-making mechanism and moving mechanism. They move towards its goals on a two-dimensional space under the guidance of external signals and its own regulations. Cells also communicate with each other according to specific interaction mechanism. The *environment* is defined as a place for cell movement. It includes signal objects, some of which help start or end the migration and others have great influence on the speed and directions of cells. It also keeps a record of current position for each cell. Comparing with traditional P systems, cells can be considered as intelligent particles with decision-making mechanism and they can move to their destination. A case study is about modeling and simulating a building evacuation problem in a fire emergency by using the IDPS model. To our best knowledge, the topic of evacuation simulation was not under study in the field of membrane computing before. The simulation result shows that the IDPS allows much easier and more precise modelling of pedestrian evacuation problems. So it is supposed to be a good simulation model for pedestrian walking behaviour.
Keywords: membrane computing, P system, modeling and simulation, pedestrian evacuation.

1 Introduction

Membrane computing is a branch of natural computing, which is inspired by the structure and the function of living cells or the organization of cells in tissues and organs [21]. The computational models in membrane computing are called *P systems*. Most P systems were proved to be computationally universal [16]. And many NP-hard problems or PSPACE problems were solved by P systems in a polynomial time or even a linear time [1,36]. Moreover, P systems provided distributed parallel and non-deterministic frameworks for computing and modeling [33], and then applied to various aspects of engineering [7,20]. On one hand, optimization algorithms with membrane structure, such as membrane-inspired evolutionary algorithms (MIEA) [21] and multi-objective membrane algorithm guided by the skin membrane (SMG-MOMA) [34], were proposed and outperformed general optimization algorithms. On the other hand, some kind of

P systems are used as modelling notation for ecological systems [2] and crowd behaviour [26]. Readers can find detailed assessments of various P systems in [22].

In traditional P systems, cells were designed for computing rather than moving. Membrane structure divided the space to several parts at the beginning. Objects or membranes evolved following specific rules. The computational results usually defined according to objects present in the output membrane in the halting condition. Recently, some P systems have involved the idea cell moving. Gemmating P systems introduces a new means of communication between membranes while keeping the definition of p system closer to the true structure of the cell. Through the movement of the membrane, the system sends a signal to the area designated by the target indication. At the same time, cells can selectively receive these signals [5]. Petre et al. got inspiration from mobile ambients and came up with mobile P-systems. By wrapping the object with the membrane and moving it to a predetermined destination, any two "static" cells can establish direct and secure communication [23]. The population P system allows cells to receive an object from other cells connected to it, which has no restriction on the transmission of the object [4]. In addition, cell movement can also be applied to the construction of the environment. Zhang et al. let the cell play the role of an obstacle and achieves dynamic environmental changes through cell movement [37]. However, most of P systems do not involve the concept of position except the spatial P systems [3] and grid-exploring P system [13]. In spatial P systems, objects in membranes are associated with positions. Membranes delimit space to different regions. Evolution rules are associated with the positions of objects. Spatial P systems were proved to be universal and could be used to model the evolution of populations in presence of geographical separations [3]. The grid-exploring P system uses generalized membranes to form the grid elements. The changes of the positions of membranes leaded to different spatial structures of the whole system. The arrangement of the inner membrane of the grid is optimized by artificial evolution to shorten the total time required for all particles to pass through the full channel. It takes the information-carrying particles as the main body of the movement [13]. In this study, we propose a simulation model called intelligent decision P system (IDPS), which is inspired by the cell migration process. Cells in this model are intelligent and movable. They choose their way to move according to their own mechanism and the interaction with their surroundings.

Cell migration is a central process along with the multicellular organisms' development. Tissue formation during wound healing and immune responses all require the accurate movement of cells in particular directions to specific locations. Cells' movement is targeted. For example, when the white blood cells chase and attack the invading bacteria, they ignore the red blood cells and other harmless substances. If they meet the obstructions on their way, they can adjust their actions to bypass them. During the migration process, there are also some kinds of competition behaviours as well as cooperation behaviours between cells. Information can be exchanged between different cells.

Cells often migrate in response to specific external signals. When they receive the signals, they open their internal switches and start the migration process. Some of the signal molecules may even affect the speed and direction of cell migration. The cells receive signaling molecule by using cell-surface receptor, which triggers a series of biochemical reactions and protein-protein interactions in the cell. After that, the signal molecules inside cells pass the information to the cytoskeleton and the molecular motor to complete the migration process. The process of cell migration is illustrated by Figure 1. First, a tip extends out of the polar feet. Second, the cell precursor and extracellular matrix form new cell adhesion. Third, cell shrinks. Finally, the tail is separated from the surrounding matrix, and the cell moves forward. The process includes cytoplasmic displacement at leading edge (front) and laminar removal of dorsally-accumulated debris toward trailing edge (back). The actions are similar to a pedestrian's walking motions. When walking, the forefoot steps on the ground first and the rear foot rises.

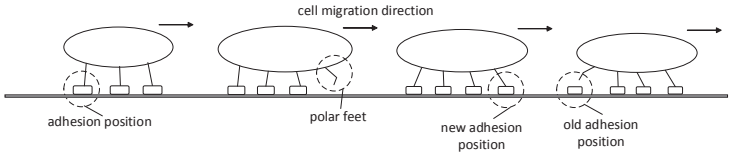


Figure 1: Cell migration

By comparison, we find that cells' migration has inherent similarities with pedestrians' walking behaviour. For example, both cells and pedestrians can move to their destination and choose routes according to their own decision-making mechanisms. They can communicate with each other to update their knowledge during these processes. They can adjust their behaviours according to the environment around them. There are blocking, pushing or competition behaviour in both cell migration process and crowd evacuation process. So a simulation model which is inspired by the cell migration process is expected to be used to model the crowd evacuation behaviour. This is the main motivation of this study.

Evacuation planning has become increasingly important in recent years which has attracted the attention of more and more researchers [10]. The efficient evacuation of pedestrians is very important especially in the presence of a disaster such as a fire. Effective evacuation strategy is the key to improve evacuation efficiency and reduce the number of casualties. The public safety manager can use it to simulate pedestrian evacuation in the corresponding place, and then notice the potential danger in some evacuation process or draw up a reasonable evacuation plan [17].

In order to ensure the safety of evacuees, it is urgent to develop procedures for the evacuation of evacuees in emergency situations. However, the study of evacuation dynamics is very complex [35]. On one hand, a large number of people who belong to different categories involved in the situation. The interaction between them is nonlinear and complex. The psychological factors also have great influence on their behaviours especially in a emergent situation. On the other hand, different evacuation scenarios usually need different strategies, and disasters often change the spatial accessibility of a scene. It is very difficult for current modeling technologies to reproduce realistic situations completely. The basic task of indoor evacuation research is to simulate the building environment [25] and evacuees' behaviours [29]. Some researchers have conducted empirical studies of evacuation behavior, but there are wide variations in the results of various studies [27]. The reason for this may be due to cultural and population differences [11] or different motives for movement [24].

There are several modelling techniques for evacuation problems in the literature. Both continuum models and network-based models are macroscopic models. Both time and space are continuous. However, it is not very good at simulating the details of individual behavior [15]. The network-based models solve the problem of simulation of discrete events [8]. There are three famous microscopic models, that is, cellular automata models, agent-based models and social-force models. Cellular automata models represent the surroundings by a grid of cells [6]. Each cell can be occupied by one pedestrian or several pedestrians. Time is discrete, and at each step the movement of the individual depends on the state of the adjacent individuals and the predefined rules [31]. Agent-based models use different agents to model individual pedestrians and form the macroscopic behaviour based on the interactions between agents [18]. Each agent has unique rules, so these models can model heterogenetic crowd [28]. But these models require a high cost of computing. In the social-force models, the movement of the individual is influenced by the forces from different directions, such as the attraction of the direction of the target and the repulsion from other individuals [12]. Game-theoretic models allow evacuees to predict the behavior of other evacuees, based on the microeconomic concept of maximizing subjective util-

ity [14,19]. It is very difficult for current modeling technologies to describe both the environment and the pedestrians as realistically as possible.

Because of the similarity between cell migration and pedestrian walking behaviours, in this work we propose a simulation model called an intelligence decision P system (IDPS), which is inspired by the cell migration process to model the crowd evacuation behaviour. The main contributions of this paper can be summarized as follows.

(1) Intelligence decision P system is defined including decision-making mechanism, moving mechanism, interaction mechanism. It also involves an accurate description of environment around cells. Each cell moves towards their goals on a two-dimensional space under the guidance of external signals and its own regulations. In each step, cells decide what is the next action to be performed through a deliberation decision-making process until the termination signal occurs. Several characteristics, such as non-deterministic maximally parallel manner, priority rules and communication rules are also inherited by our model from traditional P systems. Unlike traditional P systems, cells in the IDPS are intelligent and movable. They communicate with each other during the moving process. So the IDPS combines individual intelligence and swarm intelligence, where cells can be considered as intelligent particles.

(2) To our best knowledge, the topic of evacuation simulation was not under study in the field of membrane computing before. In this study, the IDPS is used to model the building evacuation process in the presence of a fire disaster as a new modelling technique. Evacuees with different knowledge bases are described by cells with decision-making mechanism. They percept environment and interact with each other to update their knowledge bases during the evacuation process. Each pedestrian is modelled individually. Our model can describe the movement or the interactions of pedestrians as realistically as possible. The result shows that the IDPS model allows much easier and more precise modelling of building evacuation problems.

2 Intelligence decision P system

Formally, an *Intelligence Decision P System* of degree $n \geq 1$ is defined as follows

$$\Pi = (\Gamma, E^{(0)}, C_1, \ldots, C_n, \mathcal{R}, G, s_s, s_t),$$

where:

- $n \geq 1$ (the system contains n cells, labeled with $1, 2, \cdots, n$; all these n cells are placed in the environment and the environment is labeled with 0);
- Γ is the *alphabet* of *objects*;
- The environment E is defined as a set of objects to describe the scene of cell movement. It includes signal objects, some of which help start or end the migration and others have great influence on the speed and directions of cells. It also includes a system clock to keep a record of current time of the system and a counter to calculate the number of cells going out of the exits. The width of environment is denoted by w, and the height of environment is denoted by h. $E^{(0)}$ is the set of objects in the environment at the beginning of the simulation;
- $C_i^{(j)} = \{p_i^{(j)}, v_i^{(j)}, K_i^{(j)}, m_i^{(j)}\}$ represents the state of cell i at step j, where $p_i^{(j)} = (x_i^{(j)}, y_i^{(j)})$ records the real-time location of cell i with $0 < x_i^{(j)} < w, 0 < y_i^{(j)} < h$, $v_i^{(j)}$ is the speed of the cell i at step j, $K_i^{(j)} = \{k_1, \ldots, k_n\}$ describes the knowledge base of cell i at step j, and $m_i^{(j)}$ means the type of cell. $C_i^{(0)}$ denotes the initial state of cell i.

- $\mathcal{R}$ is the set of evolution rules which includes the following types of rules:

1. *Knowledge base update rules*:
   Cells make decision based on their own knowledge bases. The initial knowledge bases are set according to cell types m_i. And then they are updated in two different ways. One is obtaining information from the environment. The other is exchanging information with other cells.
   (1) Interaction with environment:
   Cells obtain information from the environment, such as information about real-time traffic, route guidance signals, obstacles, starting signal and termination signal. Each information has a perception region. If a cell is in this region, it can get the information as shown in the following rules.

$$K_i^{(j)} \to K_i'^{(j+1)}, \forall i \in \{1, \ldots, n\},$$

$$K_i'^{(j+1)} = \begin{cases} K_i^{(j)} \cup \{a\}, & p_i \in R_a \\ K_i^{(j)}, & p_i \notin R_a \end{cases}$$

   where R_a indicates the perception region of information a. The knowledge base of cell i is updated by adding information a, if cell i reaches to R_a.
   (2) Communication with other cells:
   Cells can share information with their neighbors. We suppose the distance of two neighbors is not no more than a threshold d. If $l_{best} \in K_i'^{(j+1)} \cup K_k'^{(j+1)}$, $|p_i - p_k| \leq d$ is the information which leads to the best running plan of cell i or cell j, and l_{best} is not in $K_i'^{(j+1)}$, $K_i'^{(j+1)}$ is updated by adding information l_{best}.

$$K_i'^{(j+1)} \to K_i^{(j+1)}, \forall i \in \{1, \ldots, n\},$$

$$K_i^{(j+1)} = \begin{cases} K_i'^{(j+1)} \cup \{l_{best}\}, & l_{best} \notin K_i'^{(j+1)} \\ K_i'^{(j+1)}, & l_{best} \in K_i'^{(j+1)} \end{cases}$$

2. *Type transition rules*:
   Cell type is defined according to the knowledge base in the cell. The knowledge base change may lead to the cell type transition.

$$K_i^{(j+1)} m_i^{(j)} \to K_i^{(j+1)} m_i^{(j+1)}.$$

3. *Decision-making rules*:
   According to the knowledge base and the current position, cell i can obtain several running schemes at step j.

$$K_i^{(j)} p_i^{(j)} \to \{Scheme_{i,1}^{(j)}, \ldots, Scheme_{i,i_k}^{(j)}\}.$$

   Choose the best scheme $Scheme_{i,best}^{(j)}$ according to specific requirement, for example, minimizing distance from current position to the exit.

$$\{Scheme_{i,1}^{(j)}, \ldots, Scheme_{i,i_k}^{(j)}\} \to Scheme_{i,best}^{(j)}.$$

   It is worth mentioning that priorities can be easily added to these rules for decision-making.

4. *Position-updating rules*:
   The expected velocity and moving direction can be calculated from $Scheme_{i,best}^{(j)}$,

$$Scheme_{i,best}^{(j)} \to (v_i'^{(j)}, d_i'^{(j)}).$$

   In the process of moving, the actual situation may not allow cells to follow the direction of planning, then the cells need to adjust their behaviour. For example, when a cell encounters an obstacle (a wall or other cells in front of the cell), it needs to bypass the obstacle to move on. The actual velocity and direction of cell i at step j are denoted by $v_i^{(j)}$ and $d_i^{(j)}$, respectively.

$$(v_i'^{(j)}, d_i'^{(j)})\{o_1, \ldots, o_q\} \to (v_i^{(j)}, d_i^{(j)}),$$

   where $o_1, \ldots, o_q$ represent obstacles. Cell i updates its position according to the current position and speed.

$$(x_i^{(j)}, y_i^{(j)})(v_i^{(j)}, d_i^{(j)}) \to (x_i^{(j+1)}, y_i^{(j+1)}),$$

   where $(x_i^{(j)}, y_i^{(j)})$ means the position of cell i at step j, $(v_i^{(j)}, d_i^{(j)})$ means the speed and the moving direction of cell i at step j.

- G is the set of destinations or exits of the moving cells.
- s_s is the starting signal for cell movement, while s_t is the termination signal.

The rules of a system as above are used in the non-deterministic maximally parallel manner. When the starting signal s_s appears in the environment, cells move towards their goals under the guidance of rules in $\mathcal{R}$. In each step, cells decide what is the next action to be performed. They usually stop moving when they reach their destinations or they get the termination signal s_t from the environment.

A configuration of IDPS Π is described by the multisets of objects in the cells and the environment. $C_1^{(j)}, \ldots, C_n^{(j)}$ represent the states of all the cells present in the system at step j. They involve four basic characters of cells. That is, position, speed, knowledge base and type. The next system's configuration is determined by rules in $\mathcal{R}$ from the previous one. All computations start from the initial configuration and proceed. Cells stop moving when they reach their destinations or when they get the termination signal s_t from the environment. The system stops evolving when all cells stop moving. The simulation results can be obtained by counting the number of cells going out of the exits.

3 Intelligence decision P System for evacuation simulation

This section describes how intelligence decision P system model the evacuation process of evacuees with different knowledge bases in a fire emergency. A teaching building of China University of Geosciences (Beijing) was used as a sampling evacuation environment in this study, see Figure 2. The classrooms, the walls, the doors, the corridor and the exits are represented by yellow, gray, blue, light blue and green, respectively. There are three exits, from left to right, exit 1, exit 2 and exit 3. The black area represents the safe area outside the building.

For the sake of simplification, cells in our model have the same size and the same moving speed. The deflection angle of cell's moving direction is denoted by θ, $\theta \in [0, 360]$.The viewing angle is set by default to 120° [9]. Evacuees can only see objects within their visual fields, and evacuees cannot see the situation in other side of the wall (see Figure 3).

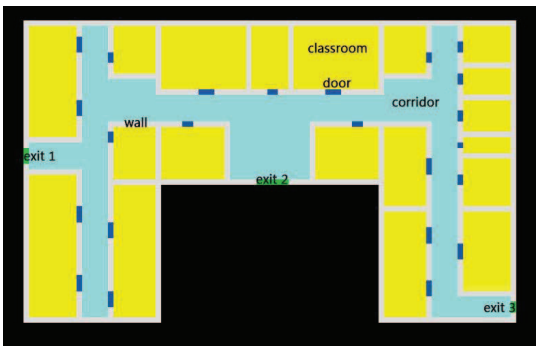


Figure 2: The experimental scene

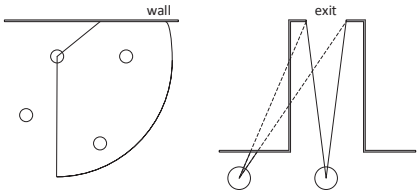


Figure 3: The visual field of a cell

3.1 Knowledge base update

There are eight categories of cells in our model. They are divided according to their familiarity of the exits. The first kind of cells have the knowledge about how to reach all the exits; the second, the third and the fourth kinds of cells have knowledge about two different exits,respectively; the fifth, the sixth and the seventh kinds of cells know one exit only and the eighth kind of cells have no idea about any passable exits. Evacuees knowing all the exits have the whole knowledge of the building, we call them "experts". Others have partial information about the building, we call them "followers". In the initial state, every cell knows at least one exit.

Cells get information to update their knowledge base in three ways. (1) The followers tend to be attracted by the experts. Followers identify experts based on the object m_i in cell i. If the followers see the experts, they will move to the experts and then they get useful information, and update the knowledge base. (2) The followers have a certain probability to interact with each other. They share their own current optimal route information, and update their knowledge base. (3) The information about the nearest exit can be obtained from the guide signs. If cells in it's perception region and find these signs, they update their knowledge base immediately. The knowledge base can be updated by using the following rules.

$$K_i' \to K_i, \forall i \in \{1, \ldots, n\},$$

$$K_i = \begin{cases} K_i' \cup \{l_{best}\}, & l_{best} \notin K_i' \\ K_i', & l_{best} \in K_i' \end{cases}$$

where l_{best} is the best information held by other evacuees who interact with the evacuee i. If the evacuee i interacts with other neighbors(the distance of two evacuees is less than 2) and acquires new information l_{best}, he updates the knowledge base K_i' to $K_i' \cup \{l_{best}\}$, otherwise the original knowledge base K_i' is maintained.

$$K_i' = \begin{cases} K_i \cup \{g\}, & p_i \in R_g \\ K_i, & p_i \notin R_g \end{cases}$$

where R_g indicates the perception region of the information g sent by guide signs. The knowledge base of evacuee i is updated by adding information g, if evacuee i reaches to R_g.

When cells get new information and update the knowledge bases, their categories may be changed at the same time. During the evacuation process, when a fire occurs, the exit may be blocked by fire. In that case, some evacuees may have no useful information to find any exit. They belong to the eighth category, and have to walk randomly until their knowledge bases are updated by interaction with other evacuees or the environment. When they get information about another exit, they will change their category once again. Similarly, cells can become "experts" when they obtain information about all exits during the evacuation process. P_0 represents evacuees who have no idea about any passable exits. P_1, P_2, and P_3 represent evacuees who know one exit, two exits and three exits, respectively.

$$m_i = \begin{cases} P_0, & K_i = \emptyset \\ P_1, & K_i = \{a\} \ or \ \{b\} \ or \ \{c\} \\ P_2, & K_i = \{a, b\} \ or \ \{a, c\} \ or \ \{b, c\} \\ P_3, & K_i = \{a, b, c\} \end{cases}$$

Where a, b, and c respectively represent the information of the three exits.

3.2 Behaviour adjustment

According to the knowledge base, evacuees tend to choose the path with the shortest distance. $Scheme_{i,best}^{(j)} \rightarrow (v_i'^{(j)}, d_i'^{(j)})$ calculates the expected speed and moving direction. However, they cannot move as expected in many cases. To avoid collisions, cells attempt to keep a distance from the surrounding obstacles while moving. These obstacles include walls and other evacuees. So cells have to adjust their behaviour to avoid collisions. If the best way to the exit is blocked, the evacuee will choose another way to move. The feasible speed and moving direction can be calculated by $(v_i'^{(j)}, d_i'^{(j)})\{o_1, \ldots, o_q\} \rightarrow (v_i^{(j)}, d_i^{(j)})$, where $o_1, \ldots, o_q$ represent obstacles.

In the narrow corridor, the flows of cells of two opposite directions are often encountered. To avoid the confusion caused by cross flow of cells, cells attempt to communicate with other cells who move in the opposite direction. They exchange information with each other and update their knowledge base. According to the new knowledge base, they re-plan evacuation routes. Some of them change their direction, and then communicate with other evacuees who are moving in the opposite direction. This process is repeated until all evacuees move in the same direction. This kind of behaviour adjustment is shown in Figure 4.

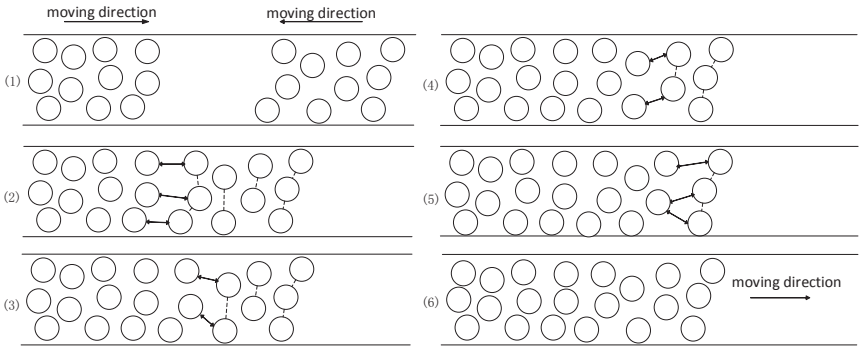


Figure 4: Interactive behaviour in cross flow

3.3 IDPS-based simulation of evacuation

All cells have initial knowledge base $K_i^{(0)}$ at the beginning. When the s_s (initial signal) appears, cells start moving. In detail, the evacuation process can be simulated according to the following steps.

Step 1: The system checks whether there is a termination signal s_t. If there is a termination signal s_t, the simulation is finished. Otherwise, go to step 2.

Step 2: If the cell finds the exit in its view field, go to step 7. Otherwise, the simulation continues.

Step 3: If the cell interacts with the expert before, go to step 6. Otherwise, the simulation continues.

Step 4: If the cell finds no expert in its view field, go to step 6. Otherwise, the simulation continues.

Step 5: The cell moves to the expert and gets information, go to step 8.

Step 6: If the cell finds the guide sign in its view field, it gets the information provided by the guide sign, go to step 8. Otherwise, the simulation continues.

Step 7: The cell has a certain probability to interact with its neighbours and share information with them.

Step 8: The cell updates the knowledge base and translates category.

Step 9: The cell makes path planning according to the decision-making rules.

Step 10: The cell adjusts its behaviour and moves with the actual velocity.

Step 11: If the cells reach the exit, the evacuation is successful. Otherwise, the simulation continues and go to step 2.

In general, the probability of interaction between cells is set to 20%. But when a cell cannot plan the route because there is no information available, its interaction with its neighbours will rise to 100% until it can plan an available evacuation route based on the information obtained. On the other hand, if a cell has interacted with one expert, he will not go looking for other experts because he has got enough information.

4 Experiments and results analysis

The simulation system is implemented using NetLogo. Two scenarios are considered in this experiment. In scenario 1, fire occurred in the classroom and did not block any exit. After the evacuation of the staff in the classroom where the fire broke out, the classroom can be closed to ensure that the fire will not spread for a certain period of time. In scenario 2, the fire happened at the exit 1 and made exit 1 impassable. In a short period of time, the fire will block the exit 1, and evacuees need to consider what strategy should be taken if the planned evacuation route is blocked in the evacuation process.

The total number of evacuees in experiment is set to 500. The proportion of experts can be adjusted. We will discuss it in section 4.1. There are six types of followers at the beginning. Their proportion is given in Table 1. And the initial position of evacuees is distributed randomly.

Table 1: Initial proportion of the number of different types of followers

Type of evacuees	1	2	3	4	5	6
Exit 1	√		√	√		
Exit 2	√	√			√	
Exit 3		√	√			√
Proportion	10%	10%	10%	20%	30%	20%

The results of experiment are evaluated by average evacuation time and the number of casualties. The average evacuation time is used to measure evacuation efficiency. It is calculated by the following equation [30]

$$Avg.E(t) = \frac{(\Sigma_{i=1}^{n_e} t_{e_i}) + t_s n_f}{n_{total}} \tag{1}$$

where n_{total} is the total number of evacuees in the building, t_{e_i} is the time to reach the exit for evacuees i, t_s is the total simulation time, n_e is the number of successful evacuations, and n_f is the total number of evacuees who did not escape in the simulation.

After some investigation, we found that the concentration of smoke generally reached the maximum value that evacuees can bear within 180s in the teaching building. So we set the simulation time to 180s. When the smoke concentration reaches this value, the evacuees in the building will get injured and not move any more. At the same time, the system sends out a termination signal to end the simulation process.

Some of the properties of the experiment are simulated by random control, so individual simulation results with the same environment and parameter specifications may be different. The properties of experiment that are controlled in this way include the initial position of evacuees and evacuees's interactive behavior. Because of this potential difference between simulations, we perform multiple simulations for each scenario and note the average result. For each experiment, the experimental result takes the average of the results of the ten experiments.

4.1 The effect of experts

Experiments were conducted to test experts' impact on evacuation efficiency in scenario 1 and scenario 2 (Figure 5 and Figure 6, respectively). Figure 5 illustrates the evacuation efficiency in scenario 1 with 30 experts and without experts. When there was no experts to guide the crowd, it was not until 174s that the evacuation was completed. This value is very close to the terminal time (180s). Under the guidance of 30 experts, all evacuees completed the evacuation within 140s. Although no one was injured in the two experiments, the efficiency of the evacuation with 30 experts was much higher than the evacuation without experts. Figure 6 illustrates the evacuation efficiency in scenario 2 with 30 experts and without experts. Since exit 1 was blocked, more than 40 evacuees failed to evacuate without experts. When there are 30 experts involved, all evacuees can evacuate within 180s. Although evacuees try to move to the nearest exit they know, sometimes that is not the best choice for them. Without accurate information about the building, evacuees spend a lot of time on redundant paths. When experts appears in the crowd, this situation can be improved greatly. Because they can provide information about the best exit to the followers.

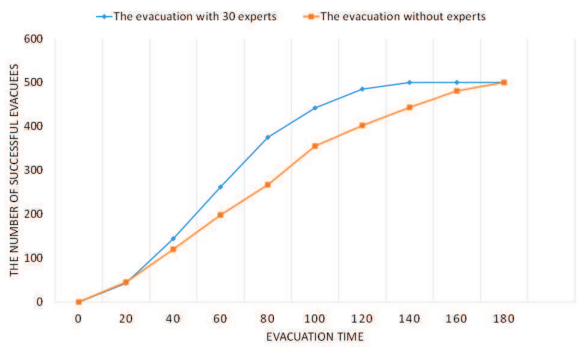


Figure 5: The effect of experts in scenario 1

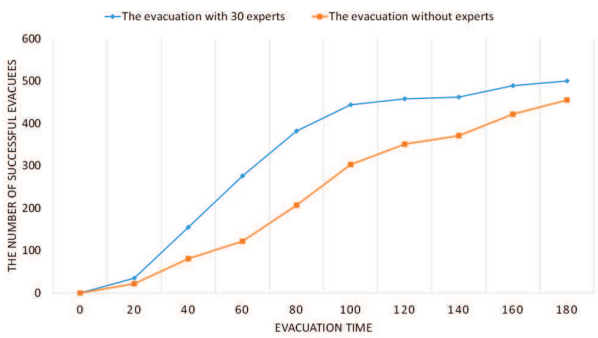


Figure 6: The effect of experts in scenario 2

The impact of experts' quantity on evacuation efficiency was also tested. The number of experts were set to 0, 20, 50 and 100 respectively, see Table 2. In scenario 1, the average evacuation time decreased with the increasing number of experts. But if the number of experts exceeded a threshold, its effect on the evacuation process was not obvious. In scenario 2, the average evacuation time was increased comparing with that in scenario 1 and 42 evacuees got injured without experts. When experts appear in the crowd, the average evacuation time is shortened obviously and the number of injured evacuees reduced greatly. As shown in Table 2, the experts' quantity plays an important role in the evacuation process. The more experts, the fewer injured evacuees. Therefore, we should pay attention to popularize the knowledge of building structure to evacuees at ordinary times.

Table 2: The effect of experts

Experiments	Scenario 1				Scenario 2			
	Exp#1	Exp#2	Exp#3	Exp#4	Exp#1	Exp#2	Exp#3	Exp#4
number of evacuees	500	500	500	500	500	500	500	500
number of experts	0	20	50	100	0	20	50	100
Avg.E(t)	85.3	75.2	50.7	45.2	102.5	86.4	72.7	67.1
Avg.Hurt	0	0	0	0	42	5	0	0

4.2 The effect of guide signs

In this subsection, we analyze the effect of guide signs. Five experiments were conducted with different numbers of guide signs. Figure 7 shows the location distribution of guide signs in the building, where the orange areas represents the guide signs. Guide sign 1 points to exit 1. Both guide sign 2 and guide sign 3 point to exit 2. Both guide sign 4 and guide sign 5 point to exit 3. There was no guide sign in Exp# 1; the number of guide signs was increased sequentially from Exp# 2 to Exp# 6.

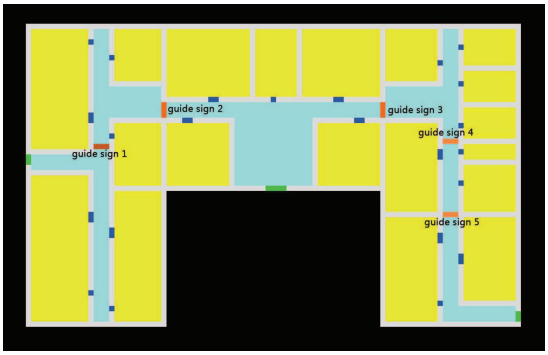


Figure 7: Distribution of guide signs

Experiments were conducted to test the guide signs' impact on evacuation efficiency in scenario 1 and scenario 2 (Figure 8 and Figure 9, respectively). Figure 8 shows that the evacuation efficiency increases with the increase of the number of guide signs in scenario 1. However, the number of evacuees near exit 1 will increase dramatically, and the probability of overcrowding and injury rise.

In scenario 2, exit 1 is congested, the evacuation efficiency is decreased. Attention should be paid to evacuation guidance in this area. Moreover, the invalid road signs will give evacuees the wrong guide. In Exp#2, because of the wrong guidance of guide sign 1, many evacuees took the ineffective path. Therefore, in the early stage of evacuation, the evacuation efficiency in Exp#2 is

lower than that in Exp#1, see Figure 9. But after 80s, the evacuation efficiency in Exp#2 began to increase gradually, and finally exceeded the evacuation efficiency in Exp#1. The reason for this phenomenon may be that guide sign 1 makes a large number of evacuees gathered near the exit 1, and the probability of successful interaction between them is increasing, so the evacuees can quickly find the location of the nearest exit. Another phenomenon that we can observe from Figure 9 is that the emergence of guide sign 2 and guide sign 3 makes the evacuation efficiency improved significantly. So guide sign 2 and guide sign 3 are more important than other signs.

The average evacuation time and the average number of casualties are reported in Table 3. It is shown that guide signs play an important role in shortening evacuation time and reducing the number of injured evacuees in most cases. But in scenario 2, when exit 1 is blocked, there are still a lot of evacuees moving towards exit 1 because of the guide sign 1. In that case, both the average evacuation time and the average number of casualties in Exp#2 are more than those in Exp#1. Therefore, if an exit is blocked by fire, the corresponding guide signs may have a bad effect on the evacuation process.

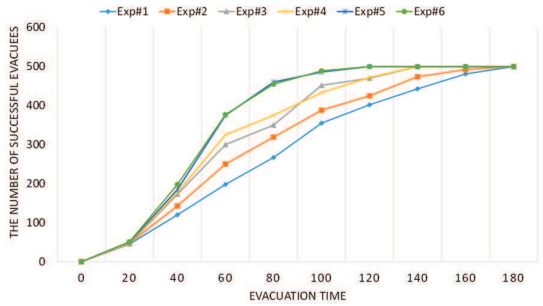


Figure 8: The effect of guide signs in scenario 1

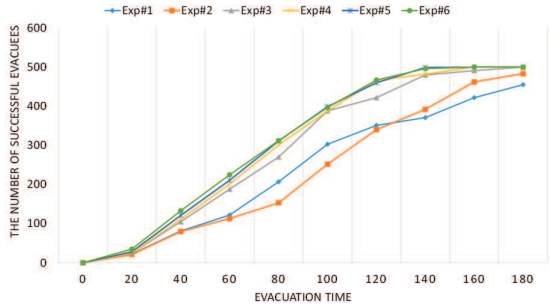


Figure 9: The effect of guide signs in scenario 2

Table 3: The effect of guide signs

Experiments	Scenario 1					
	Exp#1	Exp#2	Exp#3	Exp#4	Exp#5	Exp#6
Avg.E(t)	85.7	82.3	71.5	66.7	60.2	57.2
Avg.Hurt	0	0	0	0	0	0
Experiments	Scenario 2					
	Exp#1	Exp#2	Exp#3	Exp#4	Exp#5	Exp#6
Avg.E(t)	104.8	112.5	97.3	89.5	81.2	78.2
Avg.Hurt	56	77	11	0	0	0

4.3 The effect of composition proportion of the crowd

In this section, three experiments with different proportions of evacuees were conducted. The parameters of three different crowds are listed in Table 4. Table 5 shows the comparison of the experimental results under the influence of three different proportions of evacuees. When the number of evacuees who knew exit 1 and exit 2 was more, the average evacuation time is relatively shorter, but there are injured evacuees. When the number of evacuees who knew exit 3 was higher, although the average evacuation time was longer, there were no injured evacuees. The reason for this phenomenon may be due to a large number of evacuees gathered in the vicinity of exit 1, resulting in a long time congestion phenomenon. This congestion hampered

the movement of many evacuees who wanted to go to other exits, and some evacuees failed to evacuate in time. Therefore, in an emergency, more attention should be paid to handle the crowd efficiently and avoid congestion.

Table 4: The proportion of different types of evacuees

Type of evacuees	1	2	3	4	5	6
Exp# 1	20%	10%	20%	30%	10%	10%
Exp# 2	20%	20%	10%	10%	30%	10%
Exp# 3	10%	20%	20%	10%	10%	30%

Table 5: Results of experiments with different proportions of evacuees

Experiments	Scenario 1		Scenario 2	
	Avg.E(t)	Avg.Hurt	Avg.E(t)	Avg.Hurt
Exp#1	83.7	5	123.5	63
Exp#2	76.3	2	82.6	32
Exp#3	90.4	0	97.2	41

4.4 The effect of interaction probability

In this section, experiments were designed to analyze the effect of interactive probability. The probability of interaction between followers are set to 0%, 20%, 40%, 60%, 80% and 100%, respectively. Experts and guide signs were not involved in these experiments at the beginning. However, when an evacuee got information about all exits, he can convert his role from a follower to an expert.

Figure 10 and Figure 11 show the experimental results in scenario 1 and scenario 2, respectively. As the increase of the probability of interaction, the evacuation time decreases rapidly in the early stage. But as the probability of interaction becomes higher and higher, this reduction trend slows down, and then there is even a rising trend. This phenomenon occurs because the interaction between the evacuees takes a certain amount of time, and frequent interactions may get a lot of repetitive information. This factor has an adverse effect on evacuation efficiency. By the experiments, we can see that 60% is the best interaction probability in both scenario 1 and scenario 2.

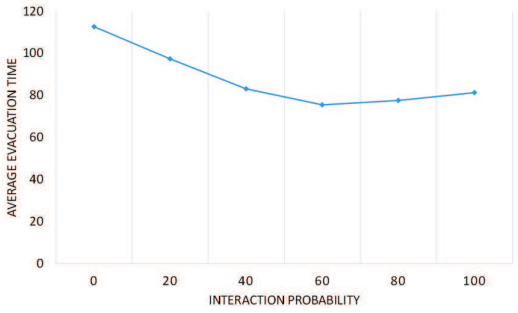


Figure 10: The effect of interaction probability in scenario 1

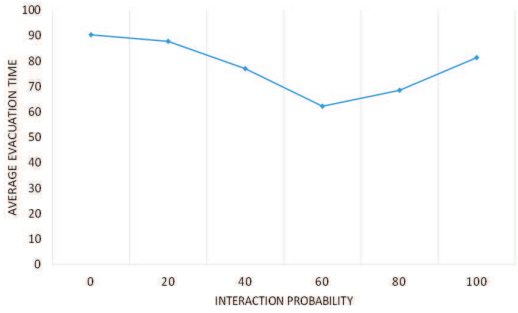


Figure 11: The effect of interaction probability in scenario 2

5 Conclusions

In view of the similarity between cell migration process and the pedestrian walking behaviour, an intelligence decision P system (IDPS), which was inspired by the process of cell migration, was proposed in this paper to simulate the building evacuation. It involves decision-making mechanism, moving mechanism, interaction mechanism of cells, as well as accurate descriptions of external information around them. It can use both continuous and discrete time and space representations. In two-dimensional space, each cell moves towards their goals under the guidance of external signals and its own regulations. Each cell has its own decision-making mechanism and moving mechanism. They can also communicate with each other or interact with external signals in their surroundings during the moving process.

A case study was modeling and simulating the evacuation of pedestrians in buildings by using the IDPS. When a fire occurs, evacuees received signals and started to move toward the exits. They made decisions which were in their best interests. We analyzed some factors that affect evacuation efficiency, including experts, guide signs, the proportions of different types of evacuees, and the interaction probabilities. The results of experiment are evaluated by average evacuation time and the number of casualties. The results showed that the IDPS model allowed much easier and more precise modelling of pedestrian evacuation problems.

This study combined the P system approach with new application scenarios. The crowd evacuation in buildings under emergency situations was modeled and simulated based on a novel P system. So far as we know, the topic was not under study in the field of membrane computing before. Some characters of P systems, such as non-deterministic maximally parallel manner, priority rules and communication rules, were used to help simulate the building environment and evacuees' behaviours. The IDPS model can easily be combined with probabilistic approaches or other technologies to simulate more complex behaviours. Our model can be applied to other types of crowd management problems, such as high-rise building evacuation simulation, with appropriate modifications.

Funding

This work was supported by the Beijing Natural Science Foundation [grant numbers 4164096]; the National Natural Science Foundation of China [grant numbers 61502012, 61373066, 61772290]; the Humanity and Social Science Youth Foundation of Ministry of Education of China [grant number 13YJC630010]; the Science and Technology Development Strategy Research Program of Tianjin [grant number 16ZLZXZF00030]; the Asia Research Center in Nankai University [grant number AS1711]; and the Collaborative Innovation Center for China Economy.

Bibliography

[1] Alhazov, A.; Martin-Vide, C.; Pan, L. (2003); Solving a PSPACE-complete problem by recognizing P systems with restricted active membranes, *Fundamenta Informaticae*, 58(2), 66–77, 2003.

[2] Barbuti, R.; Bove, P.; Milazzo, P.; Pardini, G. (2015), Minimal probabilistic P systems for modelling ecological systems, *Theoretical Computer Science*, 608, 36–56, 2015.

[3] Barbuti, R.; Maggiolo-Schettini, A.; Milazzo, P.; Pardini, G.; Tesei, L. (2011); Spatial P systems, *Nat Comput*, 10, 3–16, 2011.

[4] Bernardini, F.; Gheorghe, M. (2004); Population P Systems, *Journal of Universal Computerence*, 10(5), 509-539, 2004.

[5] Besozzi, D.; Mauri, G.; Păun, G. (2003); Gemmating P systems: collapsing hierarchies, *Theoretical Computer Science*, 296(2), 253–267, 2003.

[6] Blue, V.J.; Adler, J.L. (2001); Cellular automata microsimulation for modeling bidirectional pedestrian walkways, *Transportation Research Part B Methodological*, 35(3), 293–312, 2001.

[7] Ciobanu, G.; Pérez-Jiménez, M.J.; Păun, G. eds. (2006); Applications of membrane computing, *Springer Berlin Heidelberg*, 287(1), 73–100, 2006.

[8] Chalmet, L.G.; Francis, R.L.; Saunders, P.B. (1982); Network Models for Building Evacuation, *Fire Technology*, 18(1), 90-113, 1982.

[9] Fu, Y.W.; Liang, J.H.; Liu, Q.P.; Hu, X.Q. (2015); Crowd Simulation for Evacuation Behaviors Based on Multi-agent System and Cellular Automaton, *International Conference on Virtual Reality and Visualization.*, 103-109, 2015.

[10] Gómez, N.P.; O'Brien, K.; Silverman, B.G.; Badler, N.I. (2005); Crowd Simulation Incorporating Agent Psychological Models, Roles and Communication, *First International Workshop on Crowd Simulation*, 21–30, 2005.

[11] Helbing, D.; Johansson, A.; Al-Abideen, H.Z. (2007); Dynamics of crowd disasters: an empirical study, *Phys. Rev. E - Statist., Nonlin., Soft Matter Phys*, 75(4), 1-7, 2007.

[12] Helbing, D.; Molnar, P. (1995); Social force model for pedestrian dynamics, *Phys. Rev. E*, 51(5), 4282–4286, 1995.

[13] Hinze, T.; Weber, L.L.; Hatnik, U. (2016); Walking membranes: grid-exploring P systems with artificial evolution for multi-purpose topological optimisation of cascaded processes, *International Conference on Membrane Computing*, 251-271, 2016.

[14] Hoogendoorn, S.; Bovy, P.H.L. (2003); Simulation of pedestrian flows by optimal control and differential games, *Optimal Control Applications & Methods*, 24 (3), 153–172, 2003.

[15] Hughes, R.L. (2002); A continuum theory for the flow of pedestrians, *Transportation Research Part B*, 36(6), 507–535, 2002.

[16] Ionescu, M.; Păun, G.; Yokomori, T. (2006); Spiking Neural P Systems, *Fundamenta Informaticae*, 71, 279–308, 2006.

[17] Ibrahim, A.M.; Venkat, I.; Wilde, P.D. (2017); Uncertainty in a spatial evacuation model, *Physica A Statistical Mechanics & Its Applications*, 479, 485-497, 2017.

[18] Junaedi, H.; Hariadi, M.; Purnama, I.K.E. (2013); Multi agent with multi behavior based on particle swarm optimization (PSO) for crowd movement in fire evacuation, *International Conference on Intelligent Control & Information Processing*, 366-372, 2013.

[19] Lovreglio, R.; Ronchi, E.; Nilsson, D. (2016); An Evacuation Decision Model based on perceived risk, social influence and behavioural uncertainty, *Simulation Modelling Practice & Theory*, 66, 226-242, 2016.

[20] Pan, L.; Păun, G.; Pérez-Jiménez, M.J. (2011); Spiking neural P systems with neuron division and budding, *Science China Information Science*, 54(8), 1596–1607, 2011.

[21] Păun, G. (2000); Computing with membranes, *Journal of Computer and System Sciences*, 61(1), 108–143, 2000.

[22] Păun, G.; Rozenberg, G.; Salomaa A. (2009); *The Oxford Handbook of Membrane Computing*, Oxford University Press, 2009.

[23] Petre, I.; Petre, L. (2001); *Mobile Ambients and P-Systems*, Turku Centre for Computer Science, 2001.

[24] Predtechenskii, V.M.; Milinskii, A.I. (1978); *Planning for Foot Traffic Flow in Buildings NBS*, Fire Technology, 1978.

[25] Ronchi, E.; Nilsson, D.; Gwynne, S.M.V. (2012); Modelling the Impact of Emergency Exit Signs in Tunnels, *Fire Technology*, 48(4), 961-988, 2012.

[26] Sakellariou, I.; Stamatopoulou, I.; Kefalas, P. (2012); Using membranes to model a multi-agent system towards underground metro station crowd behaviour simulation, *ECAI 2012 workshop. Montpellier, France, August*, 28, 5–10, 2012.

[27] Schadschneider, A.; Seyfried, A. (2010); Empirical results for pedestrian dynamics and their implications for cellular automata models, *Networks & Heterogeneous Media*, 6(3), 545–560, 2010.

[28] Shi, J.; Ren, A.; Chen, C. (2009); Agent-based evacuation model of large public buildings under fire conditions, *Automation in Construction*, 18(3), 338–347, 2009.

[29] Tan, L.; Hu, M.; Lin, H. (2015); Agent-based simulation of building evacuation: Combining human behavior with predictable spatial accessibility in a fire emergency, *Information Sciences*, 295, 53-66, 2015.

[30] Wagner, N.; Agrawal, V. (2014); An agent-based simulation system for concert venue crowd evacuation modeling in the presence of a fire disaster, *Expert Systems with Applications*, 41(6), 2807-2815, 2014.

[31] Yuan, W.; Tan, K.H. (2007); An evacuation model using cellular automata, *Physica A: Statistical Mechanics and its Applications*, 384(2), 549–566, 2007.

[32] Zhang, G.; Liu, C.; Rong, H. (2010); Analyzing radar emitter signals with membrane algorithms, *Mathematical and Computer Modelling*, 52(11-12), 1997–2010, 2010.

[33] Zhang, G.; Rong, H.; Cheng, J.; Qin, Y. (2014); A population-membrane-system-inspired evolutionary algorithm for distribution network reconfiguration, *J. Electron*, 23(3), 437–441, 2014.

[34] Zhang, X.; Li, J.; Zhang, L. (2016); A multi-objective membrane algorithm guided by the skin membrane, *Nat. Comput*, 15(4), 597–610, 2016.

[35] Zhang, X.; Li, X.; Mehaffey, J., Hadjisophocleous G. (2017); A probability-based Monte Carlo life-risk analysis model for fire emergencies, *Fire Safety Journal*, 89, 51-62, 2017.

[36] Zhang, X.; Wang, S.; Niu, Y.; Pan, L. (2011); Tissue P systems with cell separation: attacking the partition problem, *Science China Information Sciences*, 54(2), 293–304, 2011.

[37] Zhang, G.; Gheorghe, M.; Pérez-Jiménez, M.J. (2017); *Real-life Applications with Membrane Computing*, Springer International Publishing, 2017.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 881-894, October 2018.

A Financial Embedded Vector Model and Its Applications to Time Series Forecasting

Y.F. Sun, M.L. Zhang, S. Chen, X.H. Shi

Yanfeng Sun[1,2], Minglei Zhang[1,2], Si Chen[1,2], Xiaohu Shi[1,2,3]*
1. Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education
Jilin University, China
Changchun, 130012, China
2.College of Computer Science and Technology
Jilin University, China
Changchun, 130012, China
3.Zhuhai Laboratory of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education
Zhuhai College of Jilin University, China
Zhuhai, 519041, China
sunyf@jlu.edu.cn, 632987245@qq.com, 1075083128@qq.com
*Corresponding author: shixh@jlu.edu.cn

Abstract: Inspired by the embedding representation in Natural Language Processing (NLP), we develop a financial embedded vector representation model to abstract the temporal characteristics of financial time series. Original financial features are discretized firstly, and then each set of discretized features is considered as a "word" of NLP, while the whole financial time series corresponds to the "sentence" or "paragraph". Therefore the embedded vector models in NLP could be applied to the financial time series. To test the proposed model, we use Radial Basis Functions (RBF) neural networks as regression model to predict financial series by comparing the financial embedding vectors as input with the original features. Numerical results show that the prediction accuracy of the test data is improved for about 4-6 orders of magnitude, meaning that the financial embedded vector has a strong generalization ability.
Keywords: embedded vector, financial daily vector, financial weekly vector, RBF neural networks.

1 Introduction

Financial time series is nonlinear, non-stationary, and often with high noise [7]. Moreover, the financial market is a dynamic complex system which is vulnerable to various external factors [23]. Therefore, it is a challenging job to analyze the financial time series, especially for its forecasting problem. Unlike the traditional predictive methods such as the Autoregressive Integrated Moving Average (ARIMA) model, the computational intelligence-based approaches are data-driven models that do not require any specific assumptions about the distribution of the data. They are widely used in the financial market, the exchange rate market and corporate financial forecasting and other fields [6].The intelligent computing methods, such as Multilayer Perceptron(MLP) [12, 29], Radial Basis Function (RBF) neural network [1, 24, 32], Adaptive Neural-Fuzzy Inference System [8], Support Vector Regression (SVR) [5, 31], Deep Belief Network [14, 25], simulation model [9], lifecycle model [22], and Particle Swarm Optimization (PSO) [21] are often used for forecasting jobs.

An important work in the field of natural language processing (NLP) is how to represent words or documents. With the applications of the deep learning algorithms [2, 16] and the machine learning algorithms [13, 27] in the field of NLP, a number of word representation methods have

been proposed. One-hot Representation [30] is simple and one of the most widely used methods, but the weaknesses are involved in high dimension of the word vector, easily leading to dimension disaster and easily resulting in the phenomenon of lexical division.

The weaknesses have been solved to a great extent of the advent of the distributed representation methods, or word embedding, which are vectors whose relative similarities correlate with semantic similarity. One of the most influential early models is Latent Semantic Analysis (LSA), developed in the context of information retrieval [10]. Latent Dirichlet Allocation (LDA) is the refinement of LSA, which is the most well-known topic model [4]. Both LSA and LDA use documents as contexts, which become computationally very expensive on large data sets. Another type of distributed representation method is neural networks based. Bengio et. al. proposed a neural network statistical language model by learning a distributed representation of words which allows each training sentence to inform the model about an exponential number of semantically neighboring sentences [3]. While until Mikolov et. al. proposed the word2vec model, and provided open source toolkit [17, 18], word embedded vector had been widely used in different kinds of NLP applications. Word2vec models are two-layer neural networks that are trained to reconstruct linguistic contexts of words. Unlike the earlier language model, word2vec represents all the words in the corpus with a real dense vector, namely the word embedding. During the learning process, the embedding vectors are trained together with language model parameters. The most advantage of word2vec is that it could be better represented by latent semantics. Two topology frameworks, namely Continuous Bag of Words (CBOW) and Skip-gram, could be adopted in word2vec model, as well as two training techniques, Hierarchical Softmax and Negative Sampling, could be selected, respectively. Compared with other neural network models, the hidden layer is removed from word2vec, and the mapping layer no longer emphasizes the order of words. Paragraph vector is an extension of word2vec, which adds the ID of a paragraph (might be a phrase, a sentence or a document) into the inputs and outputs paragraph representation vectors together with word embedded vectors [15]. Paragraph vector includes two models: PV-DM (Distributed Memory Model of Paragraph Vectors) and PV-DBOW (Distributed Bag of Words version of Paragraph Vector).

Word embedded vector is also widely applied to other fields. In machine translation [26], word embedded vector can be used to mine the relationship between words in the two languages. After training word vectors of the two languages, it can be translated by a matrix transformation. From used sentences in image understanding [11]. Images and sentence knowledge are integrated by the neural network. Task extraction is built on semantic or knowledge relations. In the social network [19], network embedding method is widely used. DeepWalk [20] model is proposed based on word2vec. It is a new method of learning the potential representation of vertices in complex network. These potential representations encode social relations in continuous vector space and can be easily utilized by other machine learning models. Tang [28] proposed a novel embedding method-LINE, which embedded the large complex network into the low dimensional vector space. The classical stochastic gradient descent restriction is solved by using a sampling algorithm for edges. LINE is suitable for many type of information network.

In the paper, the idea of word embedded vector is introduced to abstract the temporal characteristics of financial time series. After discretizing original financial features, we consider each set of discretized features as a "word" of NLP, and the financial time series as the "paragraph". Then we apply the embedded vector models to financial series data and obtain financial embedded vectors. Next we adopt the financial embedded vectors as the input of the RBF neural network for single-step financial time series prediction.

To test the proposed financial embedded vector model, we use the daily data of S&P500 index for experiments. Compared with the RBF neural network without financial embedded vector, the numerical results have been improved greatly in the prediction accuracy.

2 Financial embedded vector model

Considering each set of discretized features of financial time series as a 'word' of NLP, we apply the embedded vector models in NLP to the financial time series. Appling word2vec and Paragraph vector, we can construct financial daily vector and financial weekly vector, respectively. In section 2.1, word2vec and Paragraph vector will be introduced briefly. And then our proposed financial embedded vector model will be stated in section 2.2.

2.1 Embedded vector model in NLP

1) Word2vec

Word2vec has two topology frameworks, namely Continuous Bag of Words (CBOW) and Skip-gram. The task of the former is to predict the word given its context, while that of latter is to predict the context given a word. On the other hand, there are two training methods to train the parameters for both of the topology frameworks. One is Hierarchical Softmax, the other is Negative Sampling. In this paper, CBOW framework and Hierarchical Softmax training method are used in our proposed method. Therefore word2vec with CBOW and Hierarchical Softmax will be described in the following. CBOW framework is a neural network model with three layers: input layer, projection layer and output layer (Fig.1). Unlike Neural Network Language Model(NNLM) [3], there are no nonlinear hidden layers in CBOW. The representation vectors of the context of a word are the inputs of the model. They are mapped to the projection layer by simply summation. The output layer represents the central word, which is expressed by a leaf node of the Huffman tree.

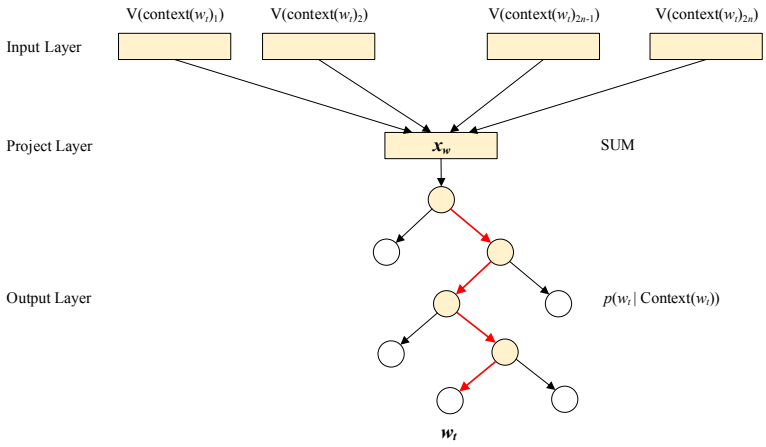


Figure 1: Diagram of CBOW model structure

Suppose the tth sample is the pair of ($Context(w_t)$, w_t),where $Context(w_t)$ is consisted of n words before and after the word w_t, $V(w)$ is the representation vector or embedded vector of word w. CBOW predict the probability of occurrence of the target word w_t by context words $Context(w_t)$. The target function is the following logarithmic likelihood function

$$L = \sum\nolimits_{w \in c} \log p(w|\text{context}(w)) \tag{1}$$

where p($w_t|Context(w_t)$) is the probability of w_t on condition of $Context(w_t)$. The goal is to maximize the logarithmic likelihood function, which could be solved by Stochastic Gradient Ascent (SGA) [17].

2) Paragraph Vector

Based on word2vec model, Le and Mikolov added paragraph ID into the inputs and proposed Paragraph Vector model, which could learn the representation vectors for both word set and paragraph set [15]. Similar with word2vec, Paragraph vector has two frameworks, namely Distributed Memory Model of Paragraph Vectors (PV-DM) and Distributed Bag of Words (PV-DBOW). Fig.2 gives the network structure of PV-DM, which could be considered as the CBOW framework of word2vec adding "Paragraph ID" in the input layer. The learning algorithm is the same with that of CBOW model. After being trained, the paragraph vectors can be used as features of the paragraph. There is one thing should be noticed that, it should be performed a short inference stage to get a representation vector for a new paragraph, by making gradient descended on the paragraph vector while holding the learned word embedded vectors fixed.

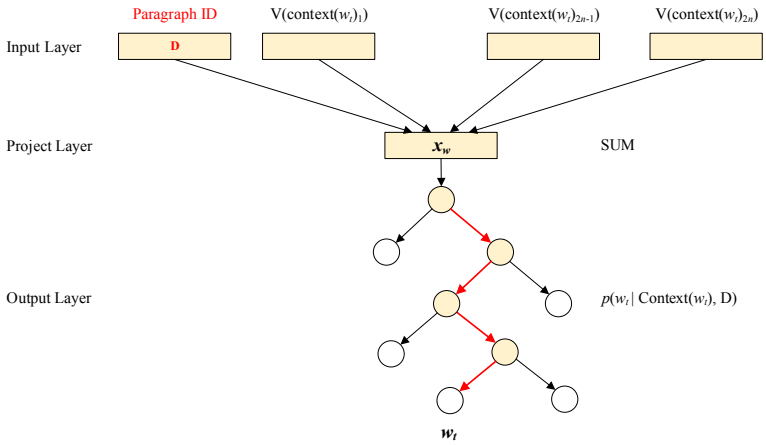


Figure 2: Diagram of PV-DM model structure

2.2 Financial embedded vector model

The key point of word2vec (or Paragraph vector) is to abstract the context relationships among different words by a neural network model and then obtain the embedding representation vectors of all the words in the dictionary. Similar with natural language, financial time series has obviously temporal characters, therefore based the embedding model in NLP, we propose a financial embedded vector model to represent the "context" relationships between different discretized feature sets of financial time series. According to word2vec and Paragraph vector models, "daily embedding vector" and "weekly embedding vector" are proposed, respectively. In the following of this section, the former will be described in detail, the latter is similar.

Assume the financial time series has m features, all of which are discretized. For example, it has 5 features and each of which is discretized into 3 sets. Then we could get the discretized feature sets from '11111' to '33333', $3^5 = 243$ sets in total. That is the "dictionary" size of financial embedding models. Firstly, we train the word2vec model to get the financial daily embedded vector by using CBOW model and Hierarchical Softmax framework. Assume the ith discretized feature set be '23213', that is to say w_i='23213', and denote the "context" set of w_i as $Context(i)$, then we will show the training procedure for the sample $(w_i, Context(i))$, which is shown in Fig.3.

The structure of Fig.3 is the same as Fig.1, each leaf node of Huffman tree represents a discretized feature set, the position of which is predetermined by its frequency in the training corpus. For example, '23213' is a leaf node of Huffman tree in Fig.3. There is a single path p^i from root node to the leaf node '23213', which includes 5 nodes and 4 branches. Each branch could be considered as a binary classification problem. Let the left branch be a positive class

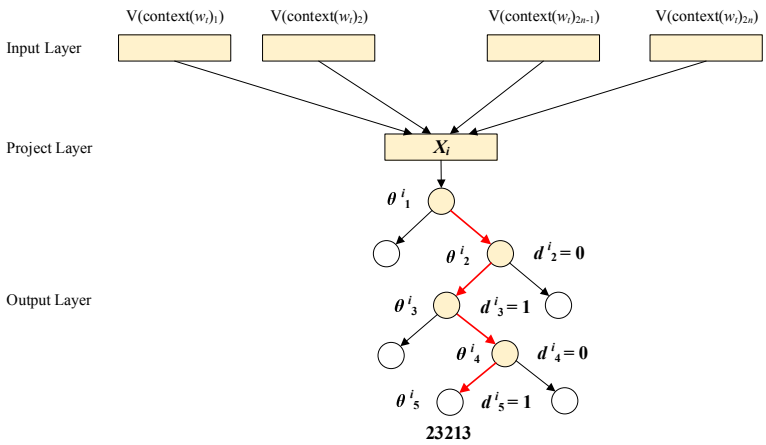


Figure 3: Diagram of CBOW model structure for the training of '23213'.

and the Huffman code be 1, while the right branch be a negative class and the Huffman code be 0. The Huffman code of w_i is the binary code of the nodes in p^i (except root node), namely that ${d_2}^i{d_3}^i{d_4}^i{d_5}^i$='0101'. It is easy to know the probability of the positive class is

$$\sigma(X_i^T\theta) = \frac{1}{1+e^{(-X_i^T\theta)}} \tag{2}$$

where θ is the undetermined non-leaf node vector, and X_i is the summation of input context vectors. The probability of the negative class is $1-\sigma\left({X_i}^T\theta\right)$. Therefore, the probability of the binary classification on the jth node of path p^i is

$$p(d_j^i|X_i,\theta_{j-1}^i) = \begin{cases} \sigma(X_i^T\theta_{j-1}^i), d_j^i = 1 \\ 1-\sigma(X_i^T\theta_{j-1}^i), d_j^i = 0 \end{cases} \tag{3}$$

where j=2,3,4,5. It can be equivalently written as

$$p(d_j^i|X_i,\theta_{j-1}^i) = [\sigma(X_i^T\theta_{j-1}^i)^{1-d_j^i}]\cdot[1-\sigma(X_i^T\theta_{j-1}^i)^{d_j^i}] \tag{4}$$

Then we could use the probability of the path p^i to represent how likely to generate w_i on condition of its context discretized feature set is $Context(i)$, which could be computed by

$$p(w_i|Context\,(i)) = \prod_{j=2}^{l^{w_i}} p(d_j^i|X_i,\theta_{j-1}^i) \tag{5}$$

where l^{w_i} is the node number of p^i , being 5 in this example. Of course, the probability of each sample pair is expected as large as possible. Then the objective functions of the whole model could be set as

$$\begin{aligned} L &= \sum_{w\in C}\log\left(p(w|Context(w))\right) \\ &= \sum_{w\in C}\log\left(\prod\nolimits_{j=2}^{l^w} p\left(d_j^w|X_i,\theta_{j-1}^i\right)\right) \\ &= \sum_{w\in C}\log\left(\prod\nolimits_{j=2}^{l^w} \sigma\left(X_i^T\theta_{j-1}^i\right)^{1-d_j^i}\left[1-\sigma^{1-d_j^i}\left(X_i^T\theta_{j-1}^i\right)\right]^{d_j^i}\right) \\ &= \sum_{w\in C}\sum\nolimits_{j=2}^{l^w}\left\{(1-d_j^i)\log\sigma\left(X_i^T\theta_{j-1}^i\right)+d_j^i\log\left[1-\sigma^{1-d_j^i}\left(X_i^T\theta_{j-1}^i\right)\right]\right\} \end{aligned} \tag{6}$$

The goal is to maximize the objective function by training parameters $V(X)$ and θ. It could be solved by Stochastic Gradient Ascent (SGA), and finally the well trained V is the "daily financial embedded vector" set. For the detailed algorithm of SGA, one can refer to [17]. The pseudo codes of the method are shown in Fig.4.

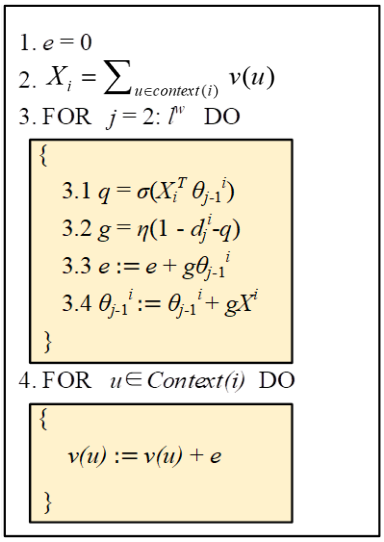


Figure 4: Pseudo codes for SGA

Furthermore, we divide the financial series into weekly segments, and each weekly segment could be considered as a "paragraph" in NLP which consists of 5 discretized feature sets as the daily financial vector. Then by introducing the Paragraph vector model, we could get "weekly financial vector" for each week. The method is similar with that of "daily financial vector".

3 The algorithm framework

In Section 2, a financial embedded vector model is proposed, from which we could get "daily financial vector" and "weekly financial vector", respectively. And then the trained financial vectors could be set as the features for financial series analysis. Compared with the original features of financial series, the financial vectors abstract more temporal characters and are expected to get better results. In this section, we develop a financial series prediction framework based on financial embedded vectors and RBF neural network. There are three main processes, namely data discretization, financial embedded vector construction, and single step prediction based on RBF neural network, which will be described in the following of this section.

3.1 Data discretization

A financial time series is usually a set of time-dependent consecutive real numbers. In our financial embedded vector model, the financial vector at each time point is a "financial word", therefore, the real features should be discretized firstly and the feature set could be divided to a "financial vocabulary". Assume there are m real features of a financial time series, the ith feature is discretized into n_i segments, then the size of "financial vocabulary" is:

$$S = \prod_{i=1}^{m} n_i \tag{7}$$

In our method, K-means algorithm is adopted for the discretization. For all the values of the ith feature, they are clustered into n_i categories, which are labelled with the set $\{1, 2, ..., n_i\}$. Then

the ith feature of series is discretized into n_i segments according to clustering results. K-means is an unsupervised clustering algorithm and often be used as discretization tool. Compared with other discretization algorithms, K-means algorithm has lower computational complexity and is advantageous for dealing with large-scale data.

3.2 Financial embedded vector construction

After data discretization, each time point of the financial series could be mapped to a "financial word" in the "financial vocabulary". And the whole financial series is a long "financial paragraph" consists of these "financial words". Take the context of a "financial word" as inputs, and the target "financial word" as output, the financial embedded vector model could be trained on CBOW framework and by SGA algorithm. Then we could obtain so called "daily financial vectors". Part of them are shown in Table1.

Table 1: Part of daily financial vectors

Discretization Results	Daily Financial Vectors
21232	[-0.0078 -0.0112 0.0211 0.0196 -0.0059 0.0211 0.0241 0.0091 0.0249 -0.0156 -0.0077 -0.0105 -0.0145 0.0141 0.0079 0.0021 -0.0110 0.0076 0.0153 0.0010]
33112	[-0.0182 -0.0200 0.0228 0.0003 -0.0129 -0.0152 -0.0154 0.0085 0.0010 0.0067 -0.0211 -0.0190 -0.0007 0.0057 0.0127 -0.0159 0.0204 0.0046 0.0056 -0.0030]
33113	[0.0077 -0.0190 -0.0205 0.0248 0.0190 0.0155 -0.0030 0.0201 -0.0025 0.0037 -0.0007 -0.0149 0.0066 -0.0067 -0.0191 0.0238 0.0219 -0.0055 -0.0011 0.0025]

We consider five days' "financial words" of a week as a weekly "financial sentence". Adding the sentence ID into the financial embedded vector model, it could be trained by stochastic gradient descent algorithm on PV-DM framework, and get "weekly financial vectors". Part of them are shown in Table2.

Table 2: Financial weekly vectors

Discretization Results (Weekly)	Weekly Financial Vectors
[33112 31232 21232 23113 31112]	[0.0429 -0.0725 -0.0783 0.2201 0.0254 -0.1170 0.0982 -0.0089 0.0813 -0.0245 -0.1699 -0.1943 -0.0463 -0.1262 0.0201 0.1168 -0.1370 0.0347 -0.1303 -0.1324]
[23113 33113 31212 33113 33113]	[0.0318 -0.0232 -0.1067 0.1555 0.0652 -0.1573 -0.0302 -0.1005 0.0855 0.0142 -0.0757 -0.1824 -0.0594 -0.0842 -0.0969 0.1414 -0.1923 0.0295 -0.0568 -0.1683]
[12313 31233 21233 23323 12132]	[0.0407 -0.0812 -0.1055 0.2603 -0.0162 -0.1914 0.1718 -0.0282 0.0654 0.0374 -0.2742 -0.1971 -0.0573 -0.1199 0.0504 0.1890 -0.1782 0.0816 -0.1934 -0.1976]

3.3 Single step prediction based on RBF neural network

To predict the financial series data, we select radial basis function (RBF) neural network as the forecasting model. RBF neural network is a particular type of Multilayer Perceptron (MLP) neural network, with input, hidden and output three layers (Fig.5) [26]. The hidden layer consists of RBF neurons, which use radial basis function as their active functions.

After getting the financial embedded vectors, we use RBF neural network to build the predicting models for one-step forecasting of the daily and weekly closing price, respectively. The original percentage data, the daily financial vector and the weekly financial vector are used as input data for comparison. Therefore we build six models for two goals with three types of inputs, which will be described in detail in section 4.2. In the hidden layer, the mostly used Gaussian function is selected as the active function. There is only one output node in all six models for our goal is one-step daily or weekly closing price predicting.

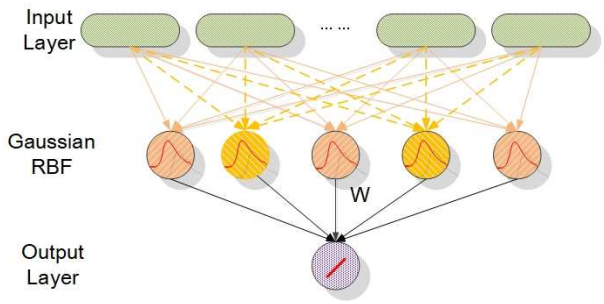


Figure 5: RBF network structure

4 Experiments

The S&P500 index has a longer trading history and enough sample which can reflect the laws of a mature securities market. The data are more stable and truthful. We collect the S&P500 index data from Yahoo Finance [33]. Date is from January 4, 1950 to November 14, 2016, a total of 16826 trading days. Every daily trading data includes five-dimensional data, namely the opening price, the highest price, the lowest price, the closing price and the volume. To eliminate the dimension influence of the five-dimensional data and improve the stability of the data, we will firstly perform the data pretreatment. Next, using RBF neural network and selecting different feature combinations, daily closing price and weekly closing price are predicted, respectively.

4.1 Data pretreatment

Firstly, Eq.8 is performed for percentage processing of the 5-dimensional data, respectively.

$$x'_t = (x_t - x_{t-1})/x_{t-1} \tag{8}$$

where x_t is the data for the day t,x_t-1 is the data for the day t-1, x'_t is the percentage of the data for the day t.

Secondly, the K-means algorithm will be adopted for the discretization of the original data. We take three clusters in the K-means algorithm. Finally, all the five features are clustered into three classes independently. That is to say, the discretized feature space size is $3^5 = 243$. Set the cluster labels as 1, 2, 3, a five-dimensional data of a time point will be discretized into $\{1, 2, 3\}^5$. Take for example, the clustering results of the five dimensions for the date 1950-01-04 are 2, 1, 2, 3, and 2, respectively. Therefore, the discretization result for the date is '21232', which is "financial word" of that day. After discretizing 5-dimensional time series data, we apply the financial embedded vector models to obtain "daily financial vectors" and "weekly financial vectors", which is described in section 3.2. For example, the daily financial vector of '21232' is shown in the first line of Table 1.

4.2 Daily closing price forecasting based on financial embedded vector

To test the performance of the obtained financial embedded vectors, they are set as the input features to predict daily closing price in this section. RBF network is applied as the forecasting model, and three types of feature combinations are arranged for comparison. Denote them as Model1, Model2, and Model3, the detail of which are described as follows:

Model1-Comparison daily model. The model output is the closing price for day t, that is, the closing price for day t will be forecasted, which is 1-dimensional data. The inputs are the

data three days before, the percentage data for day t-1, t-2, and t-3. The data for each day includes 5 percentage data of original features, namely that, the opening price, the highest price, the lowest price, the closing price and the volume. Therefore the dimension of the inputs is 15. The previous ten thousand data are used for training, the rest for testing.

Model2-Financial daily vector daily model. The model output is the same as Model1. The inputs are the financial daily vectors of three days before, namely day t-1, t-2, and t-3. In this model, we will take the dimension of each financial daily vector as 20. Then the dimension of input is 60. The division of training data and testing data is the same with Model1.

Model3-Financial daily and weekly vector daily model. The model output is the same as above two models. The inputs are the financial daily vectors for day t-1 and t-2, and the financial weekly vector for the last week of the date. In this model, we will take the dimension of each financial daily (and weekly) vector as 20, and then the dimension of input is also 60. The division of training data and testing data is the same with above two models.

RBF neural network training function is the built-in function in MATLAB. An important parameter is distribution density ('*Spread*' in the built-in function). A trade-off between over fitting and under fitting will be considered by different value of *Spread*. The value of *Spread* is set as 10 values in our experiments, which are 0.01, 0.05, 0.1, 0.3, 0.4, 0.7, 1, 5, 10, 20 and 50, respectively. Mean Square Error (MSE) is used as the evaluation index of forecasting effect, which is defined by

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(z_i - y_i)^2 \tag{9}$$

where z_i is the real value of the ith data, y_i is the forecasting value of the ith data, n is the sample number. The smaller the MSE value is, the more accurate the prediction is. The results of MSE for both training set and testing set are shown in Table 3.

From the results in Table 3, the MSE values of the three models for the training set are very similar. Model1 even gets the least average MSE among the three models, though all of them reach 10-4 magnitude. While for the testing set, the MSE values differ greatly in magnitude for different Spreads and different models. Model2 and Model3 are far better than Model1, and Model3 is superior to Model2. For example, the average MSE value of Model2 is 99.8% improved from Model1, and that of Model3 is 92.0% decreased to Model2. According to the big differences among the three models results of the testing data, it could be concluded that the financial embedded vector can improve the generalization ability of the models, especially for the combination of daily financial vector and weekly financial vector.

Fig.6 shows the comparing results of the predicted daily closing price curves of the three models. The curves in the top figure are the local enlarged of the ones in bottom figure. From the bottom figure, it is easy to find that both curves of Model2 and Model3 are obvious closer to actual data curve than that of Model1. When we focus on the enlarged curves, we could find that the performance of Model3 is better than that of Model2.

4.3 Weekly closing price forecasting based on financial embedded vector

To further examine the effect of financial embedded vectors, longer periods of data are adopted. In this section, experiments are performed to predict the weekly closing price. Similar with the above section, RBF is chosen as the forecasting model, and three types of feature combinations are arranged for comparison. They are denoted as Model4, Model5 and Model6, which are described below.

Model4-Comparison weekly model. Assume the closing day (Friday) of the target week is day t, the closing price of day t is the output of the model, which is a 1-dimensional value. The

Table 3: Prediction results of daily close price

	MSE for training				MSE for testing		
Spread	**Model1**	**Model2**	**Model3**	**Spread**	**Model1**	**Model2**	**Model3**
0.01	0	0.000029	0.000053	0.01	0.00410	0.000169	0.000198
0.05	0	0.000029	0.000054	0.05	0.10830	0.000168	0.000194
0.1	0.000007	0.000029	0.000054	0.1	287.808	0.000470	0.000191
0.3	0.000014	0.000029	0.000054	0.3	1225.83	0.000334	0.000266
0.4	0.000013	0.000029	0.000054	0.4	48.7838	0.005158	0.000216
0.7	0.000031	0.000029	0.000054	0.7	35.3886	1.243979	0.029582
1	0.000039	0.000029	0.000054	1	88.7457	2.229143	0.214848
5	0.000059	0.000030	0.000054	5	12.8574	0.003093	0.033577
10	0.000063	0.000042	0.000057	10	2.10287	0.000398	0.000277
20	0.000065	0.000053	0.000061	20	2.71368	0.000214	0.000193
50	0.000066	0.000065	0.000066	50	1.30023	0.000166	0.000161
average	**0.000032**	**0.000036**	**0.000056**	**average**	**155.058**	**0.316663**	**0.025428**

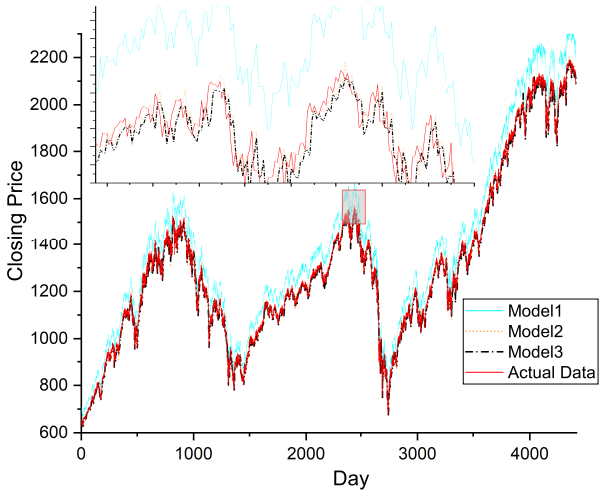


Figure 6: Forecasting and actual daily closing price in the training set

inputs are the percentage data of original features of 5 days before, namely from day t-1 to t-5. Therefore the dimension of the input is 25. Also, the previous ten thousand data are used for training, the rest for testing.

Model5-Financial daily vector weekly model. The output is the same as Model4. The inputs are the financial daily vectors of 5 days before, namely from day t-1 to t-5. For the financial daily vector is 20-dimensonal, the input dimension is 100. The division of training set and testing set is the same with Model4.

Model6-Financial weekly vector weekly model. The output is the same as Model4. The input is the weekly vector of the week before, which is 20 dimensions. The division of training set and testing set is the same with Model4.

The results of MSE for both training set and testing set are shown in Table 4. The results are very similar with those of above section. The MSE values of the three models for the training set are similar and very small. For the testing set, the MSE values differ greatly from different models. Model5 and Model6 are far better than Model4, and Model6 is superior to Model5. The average MSE value of Model5 is only 1.16 x 10-5% of that of Model4, and that of Model6 is 13.52% of that of Model5.

Table 4: Prediction results of weekly close price

	MSE for training				**MSE for testing**		
Spread	**Model4**	**Model5**	**Model6**	**Spread**	**Model4**	**Model5**	**Model6**
0.01	0	0.000003	0.000004	0.01	0.00018	0.000149	0.000148
0.05	0	0.000003	0.000004	0.05	0.00140	0.000134	0.000211
0.1	0	0.000003	0.000004	0.1	0.00984	0.000126	0.000185
0.3	0	0.000003	0.000004	0.3	54.2725	0.002544	0.000455
0.4	0	0.000003	0.000004	0.4	30.7013	0.000280	0.000526
0.7	0	0.000003	0.000004	0.7	2779.44	0.001075	0.000735
1	0	0.000003	0.000004	1	211338	0.016169	0.000740
5	0.000021	0.000003	0.000045	5	6941.79	0.002522	0.000148
10	0.000034	0.000003	0.000051	10	2492.17	0.001514	0.000133
20	0.000043	0.000003	0.000052	20	756.955	0.001221	0.000133
50	0.000049	0.000010	0.000058	50	0.68581	0.000465	0.000124
Average	**0.000013**	**0.000004**	**0.000021**	**Average**	**20399.5**	**0.002382**	**0.000322**

Fig.7 shows comparing results of the predicted weekly closing price curves of the three models. The graph in the top of Fig.7 is the enlarged one of the local area of the bottom figure. From the figure, it is easy to find that both curves of Model5 and Model6 are obviously closer to the actual data curve than that of Model4, and Model6 is the best one. Also, the financial embedded vector can greatly improve the generalization ability of the models, especially for the weekly financial vector.

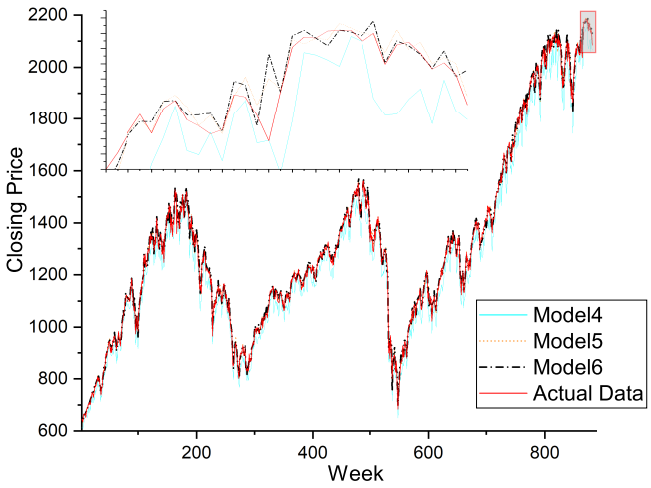


Figure 7: Forecasting and actual weekly closing price in the testing set

5 Conclusion

In the paper, inspired by the idea of embedded vector idea in NLP, we propose the financial embedded vector model to represent the financial time series, which could abstract the temporal features very well. Focused on the financial time series prediction, a framework is developed

based on the financial embedded vector model and RBF neural network. To test the effectiveness of our method, it is applied to S&P500 index historical data for the daily and weekly closing price prediction. Comparing different kinds of input features, it could be concluded that the financial embedded vectors could improve the precision greatly to the original features.

Funding

The authors are grateful to the support of the National Natural Science Foundation of China (61373050), and Jilin Provincial Key Laboratory of Big Date Intelligent Computing (20180622002JC).

Bibliography

[1] Akbilgic, O.; Bozdoganm, H.; Balaban, M.E. (2014); A novel Hybrid RBF Neural Networks model as a forecaster, *Statistics and Computing*, 24(3), 365-375, 2014.

[2] Al-Ayyoub, M.; Nuseir, A.; Alsmearat, K.; Jararweh, Y.; Gupta, B. (2018); Deep learning for Arabic NLP: A survey, *Journal of Computational Science*, 26522-531, 2018.

[3] Bengio, Y.; Ducharme, J.; Vincent, P.; Janvin, C. (2003); A neural probabilistic language model, *The Journal of Machine Learning Research*, 3(2), 1137-1155, 2003.

[4] Blei, D.M.; Ng, A.Y.; Jordan, M.I. (2003); Latent dirichlet allocation, *Journal of Machine Learning Research*, 3(1), 993-1022, 2003.

[5] Cao, L. (2003); Support vector machines experts for time series forecasting, *Neurocomputing*, 51321-339, 2003.

[6] Cavalcante, R.C.; Brasileiro, R.C.; Souza, V.L.F.; Nobrega, J.P.; Oliveira, A.L.I. (2016); Computational Intelligence and Financial Markets: A Survey and Future Directions, *Expert Systems with Applications*, 55194-211, 2016.

[7] Chang, D.; Ma, Y.F.; Ding, X.L. (2016); Time Series Clustering Based on Singularity, *International Journal of Computers Communications & Control*, 12(6), 790-802, 2017.

[8] Dai, Y.; Wu, W.; Zhou, H.B.; Zhang, J.; Ma, F.Y. (2018); Numerical Simulation and Optimization of Oil Jet Lubrication for Rotorcraft Meshing Gears, *International Journal of Simulation Modelling*, 17(2), 318-326, 2018.

[9] Dai, Y.; Zhu, X.; Zhou, H.; Mao, Z.; Wu, W. (2018); Trajectory Tracking Control for Seafloor Tracked Vehicle By Adaptive Neural-Fuzzy Inference System Algorithm, *International Journal of Computers Communications & Control*, 13(4), 465-476, 2018.

[10] Deerwester, S.; Dumais, S.T.; Furnas, G.W.; Landauer, T.K.; Harshman, R. (1990); Indexing by latent semantic analysis, *Journal of the American Society for Information Science*, 41(6), 391-407, 1990.

[11] Frome, A.; Corrado, G.; Shlens, J.; Bengio, S.; J. Dean; Ranzato, M.A.; Mikolov, T. (2013); DeViSE: A Deep Visual-Semantic Embedding Model, *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 262121-2129, 2013.

[12] Jasemi, M.; Kimiagari, A.M.; Memariani, A. (2011); A modern neural network model to do stock market timing on the basis of the ancient investment technique of Japanese Candlestick, *Expert Systems with Applications*, 38(4), 3884-3890, 2011.

[13] Khan, W.; Daud, A.; Nasir, A.; Amjad, T. (2016); A survey on the state-of-the-art machine learning models in the context of NLP, *Kuwait Journal of Science*, 43(4), 95-113, 2016.

[14] Kuremoto, T.; Kimura, S.; Kobayashi, K.; Obayashi, M. (2014); Time series forecasting using a deep belief network with restricted Boltzmann machines, *Neurocomputing*, 13747-56, 2014.

[15] Le, Q.V.; Mikolov, T. (2014); Distributed Representations of Sentences and Documents, *the 31st International Conference on Machine Learning*, 32(2), 1188-1196, 2014.

[16] LeCun, Y.; Bengio, Y.; Hinton, G. (2015); Deep learning, *Nature*, 521(7553), 436-444, 2015.

[17] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. (2013); Efficient Estimation of Word Representations in Vector Space, *1st International Conference on Learning Representations (ICLR2013)*, 2013.

[18] Mikolov, T.; I. Sutskever, I.; Chen, K.; Corrado, G.; Dean, J. (2013); Distributed Representations of Words and Phrases and their Compositionality, *the 26th International Conference on Neural Information Processing Systems*, 263111-3119, 2013.

[19] Moyano, L.G. (2017); Learning network representations, *The European Physical Journal Special Topics*, 226(3), 499-518, 2017.

[20] Perozzi, B.; Al-Rfou, R.; Skiena, S. (2014); DeepWalk: online learning of social representations, *the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701-710, 2014.

[21] Pulido, M.; Melin, P.; Castillo, O. (2014); Particle swarm optimization of ensemble neural networks with fuzzy aggregation for time series prediction of the Mexican Stock Exchange, *Information Sciences*, 280188-204, 2014.

[22] Rehar, T.; Ogrizek, B.; Leber, M.; Pisnic, A.; Buchmeister, B. (2017); Product Lifecycle Forecasting Using System's Indicators, *International Journal of Simulation Modelling*, 16(1), 45-57, 2017.

[23] Scheffer, M.; Carpenter, S.R., ; Lenton, T.M.; Bascompte, J.; Brock, W.; Dakos, V.; van de Koppel, J.; van de Leemput, I.A.; Levin, S.A.; van Nes, E.H.; Pascual, M.; Vandermeer, J. (2012); Anticipating Critical Transitions, *Science*, 338(6105), 344-348, 2012.

[24] Shen, W.; X. Guo, X.; Wu, C.; Wu, D. (2011); Forecasting stock indices using radial basis function neural networks optimized by artificial fish swarm algorithm, *Knowledge-Based Systems*, 24(3), 378-385, 2011.

[25] Shen, F.; Chao, J.; Zhao, J. (2015); Forecasting exchange rate using deep belief networks and conjugate gradient method, *Neurocomputing*, 167243-253, 2015.

[26] Singh, S.P.; Kumar, A.; Darbari, H.; Singh, L.; Rastogi, A.; Jain, S. (2017); Machine translation using deep learning: An overview, *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, 162-167, 2017.

[27] Sun, S.; Luo, C.; Chen, J. (2017); A review of natural language processing techniques for opinion mining systems, *Information Fusion*, 36(Supplement C), 10-25, 2017.

[28] Tang, J.; M. Qu, M.; Wang, M.; Zhang, M.; Yan, J.; Mei, Q. (2015); LINE: Large-scale Information Network Embedding, *the 24th International Conference on World Wide Web*, 1067-1077, 2015.

[29] Tsai, C.-F.; Hsiao, Y.-C. (2010); Combining multiple feature selection methods for stock prediction: Union, intersection, and multi-intersection approaches, *Decision Support Systems*, 50(1), 258-269, 2010.

[30] Turian, J.; Ratinov, L.; Bengio, Y. (2010); Word representations: a simple and general method for semi-supervised learning, *the 48th Annual Meeting of the Association for Computational Linguistics*, 384-394, 2010.

[31] Wang, J.; Hou, R.; Wang, C.; Shen, L. (2016); Improved v -Support vector regression model based on variable selection and brain storm optimization for stock price forecasting, *Applied Soft Computing*, 49164-178, 2016.

[32] Xiong, T.; Y. Bao, Y.; Hu, Z.; Chiong, R. (2015); Forecasting interval time series using a fully complex-valued RBF neural network with DPSO and PSO algorithms, *Information Sciences*, 30577-92, 2015.

[33] [Online]. http://finance.yahoo.com, Accesed on 1 Jan 2017.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(5), 895-907, October 2018.

Service Innovation Decision Analysis Based on Influence Diagrams

Liang Tan
School of Information Management
Jiangxi University of Finance and Economics
Nanchang 33013, China
ghosticy@foxmail.com

Abstract: The influence diagram is a probabilistic model for presenting decision problems as a directed graph. In this study, the dynamic influence diagram and the interactive dynamic influence diagram are used to model the three parties to service innovation: customers, suppliers, and service enterprises. The models analyze the decisions of these different parties and describe the process by which service enterprises should consider their own innovation conditions as well as those of the other parties, that is, customers and suppliers. Moreover, during the process of service innovation, service enterprises should be in constant communication with customers and suppliers. After the customers and suppliers respond, service enterprises can modify their innovation decision-making, and improve service innovation quality and income.
Keywords: influence diagram, service innovation, decision-making.

1 Introduction

Traditional decision-making tools such as the decision tree and decision table are limited and cannot describe the conditional independence and time sequence relations between uncertain quantities. By contrast, the influence diagram (ID)- a probabilistic model for presenting decision problems as a directed graph-uses these relations to simplify the solution, thereby quickly becoming a research hotspot and the key point of decision analysis in uncertain environments. Koller in Stanford University et al. propose a multi-agent ID (MAID), which uses a structure that affects the characteristics of the game, combined with game theory and the graph model [12]. MAID can effectively represent the static structure between the multi-agent relationship, and deal with complex multi-agent countermeasures. Compared with game theory, MAID increases the conditional independence of uncertain variables, thereby simplifying the model and making it easier to solve. In order to express the temporal relation, this paper applies dynamic ID (DID) or interactive DID (I-DID) to solve dynamic decision-making problems with single or multiple agents [14].

Service innovation refers to new ideas, new technical methods, and new or improved service, to ensure that potential users have a service experience that is different from past experiences [4]. Service innovation is not restricted to service industry innovation alone; all service-related innovations can be classified under the concept of service innovation [17]. Bilderbeek, Hertog et al. developed the four-dimension model of service innovation, consisting of new service concepts, service delivery systems, customer interfaces, and non-technical factors [3]. Hertog classified the innovation of service enterprises into five modes-internal innovation, service process innovation, comprehensive innovation, customer-driven innovation, and supplier-dominated innovation and discussed the positive effect of knowledge-intensive services on the role of innovation. Aranda, Molina–Fernández, who created the service innovation degree model, emphasized that organization members' knowledge and their ability to exchange such knowledge effectively, is the

foundation of successful innovation. Therefore, knowledge management has become a decisive factor in service innovation [2]. Galluj believes that service innovation has two main interaction features: (1) external interaction, which refers to employees interacting with external customers and (2) internal interaction, which refers to the interaction between internal staff and the senior leadership. The two parties will consciously or subconsciously participate in driving innovation consciousness [9].

2 Participants in service innovation

Service innovation mainly consists of three participants-service enterprises, customers, and suppliers. Enterprise leaders, top-level staff members, and innovation teams together implement service innovation through the internal structure of the enterprise. The participation of customers and suppliers in service innovation is an important channel for innovation and supports service innovation through the external structure of the service enterprise. As is shown in Figure 1, service innovation is a process in which managers, employees, innovation teams, customers, and suppliers jointly participate.

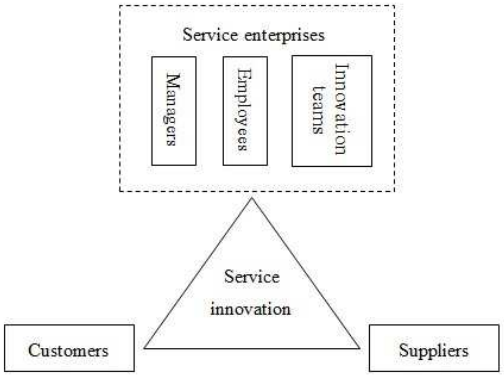


Figure 1: Participants in Service Innovation

Service innovation is often created through the cooperation of service enterprises, customers, and suppliers. The overall process of service innovation refers to participation in the transformation of ability from an internal point of view. When defining innovation, service enterprises, customers (existing customers, planned customers, potential customers), and suppliers often convey their opinions to the senior staff, leaders, and innovation groups of service companies. They cooperate with the company to implement innovative ideas more effectively and facilitate the emergence of new service definitions. Service enterprises need customers and suppliers to participate in the research and development (R&D) and planning of new services. This cooperative process allows each participant to shares their ability. After new service plans are developed, it is necessary to implement these plans. All staff, right from the top to the bottom, need to learn the new service concepts to facilitate new service delivery, marketing management, sales service, and so on. In the process of following reviews and innovation safeguards, the content transformation between companies and their customers is critical. The service company may obtain market evaluations of the new service from customers and implement the relevant innovation guarantee scheme in the new service, which cannot spread the knowledge between the customer and the company. Therefore, service innovation is a comprehensive process that involves many important participants. There are three main aspects to service innovation: (1) knowledge transfer

from customers; (2) knowledge integration of service enterprises; and (3) knowledge sharing of suppliers.

In general, service innovation is an activity that many subjects participate in together. The service enterprise designs and develops the innovation. The customer is the motivation behind the innovation and accepts the results. The supplier is a partner in service innovation [20,21]. In this process, customers, suppliers, and service enterprises are all agents. The three-party decision-making system is, thus, a multi-agent decision making model. There is considerable literature that uses game and evolution game theory to analyze the decision-making relationship. So, I-DID is used to analyze the evolutionary relationship among the three parties [13], and the simplified Discriminative Model Updates (DMU) model is used to solve the decision problems of service innovation [18].

3 ID of single agent

First, the customer ID is modeled as shown in Figure 2. Suppose the customer is agent j; the customer's possible actions are participation and non-participation; the possible observation values are that the innovation conditions are mature or that they are immature.

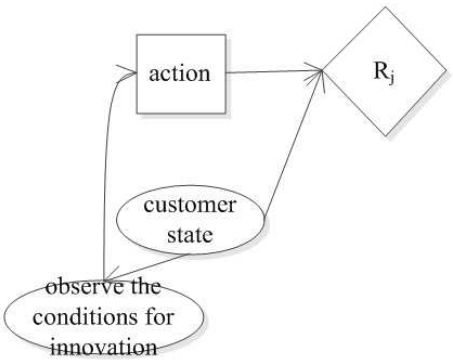


Figure 2: A Customer ID Model

In the figure, the rectangle is the decision node; the state and observation are random nodes, represented by an ellipse and associated with a conditional probability table; R_j is the value node, represented by a rhombus and associated with a value function. Suppose that the probability of customers' interest is x_1. The conditional probability of the state transfer function of the customer is shown in Table 1, the conditional probability of the observation function is shown in Table 2, and the conditional probability of the value function is shown in Table 3.

Table 1: Conditional Probability of Customers' State Transfer Function

Interested	Disinterested
x_1	1-x_1

Secondly, the supplier's ID is modeled as shown in Figure 3. Suppose the supplier is agent k; the supplier's possible actions are cooperation and non-cooperation; the possible observations are that the conditions of innovation are mature or that they are immature.

Suppose that the probability of the supplier's interest is x_2. The conditional probability of the state transfer function of the supplier is shown in Table 4, the conditional probability of the

Table 2: Conditional Probability of Customers' Observation Function

State	Mature Innovation Conditions	Immature Innovation Conditions
Interested	y_1	$1\text{-}y_1$
Disinterested	$1\text{-}y_1$	y_1

Table 3: Conditional Probability of Customers' Value Function

Action	Interested	Disinterested
Participation	z_{11}	z_{12}
Non-Participation	z_{13}	z_{14}

observation function is shown in Table 5, and the conditional probability of the value function is shown in Table 6.

Finally, the ID model of the service enterprise is as shown in Figure 4. Suppose that the service enterprise is the deciding agent i. There are two possible actions-innovation, non-innovation; the possible observations are that the conditions of innovation are mature or that they are immature; moreover, there are two states of service enterprises-interested and disinterested.

Suppose that the probability of service enterprise interest is x_3. The conditional probability of the state transfer function of the supplier is shown in Table 7, the conditional probability of the observation function is shown in Table 8, and the conditional probability of the value function is shown in Table 9.

4 ID of multiple agents

In order to research the multi-agent decision-making problem in a dynamic uncertain environment, MAID, interactive ID, and I-DID are proposed. I-DID is a research method to solve the dynamic decision process. It is based on the interactive partially observable Markov decision process (I-POMDP), which is an extension of the partially observable Markov decision process (POMDP) [11]. In I-POMDP, the deciding agent considers other agents as single decision-making subjects, rather than simply treating them as noise from the environment, as game theory does. This better matches reality. But I-POMDP also cannot express the decision process directly, and cannot solve the complex computation problem in the state space. Therefore, we choose I-DID

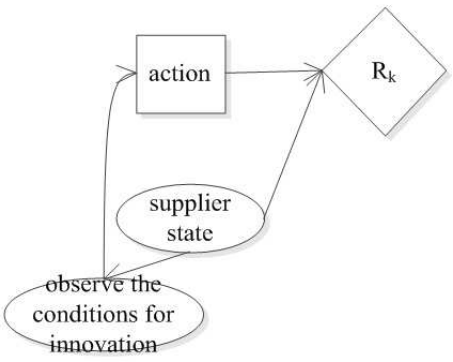


Figure 3: A Supplier ID Model

Table 4: Conditional Probability of Supplier's State Transfer Function

Interested	Disinterested
x_2	1-x_2

Table 5: Conditional Probability of Supplier's Observation Function

State	Mature Innovation Conditions	Immature Innovation Conditions
Interested	y_2	1-y_2
Disinterested	1-y_2	y_2

Table 6: Conditional Probability of Supplier's Value Function

Action	Interested	Disinterested
Cooperation	z_{21}	z_{22}
Non-Cooperation	z_{23}	z_{24}

Table 7: Conditional Probability of Service Enterprise's State Transfer Function

Interested	Disinterested
x_3	1-x_3

Table 8: Conditional Probability of Service Enterprise's Observation Function

State	Mature Innovation Conditions	Immature Innovation Conditions
Interested	y_3	1-y_3
Disinterested	1-y_3	y_3

Table 9: Conditional Probability of Service Enterprise's Value Function

Action	Interested	Disinterested
Innovation	z_{31}	z_{32}
Non-Innovation	z_{33}	z_{34}

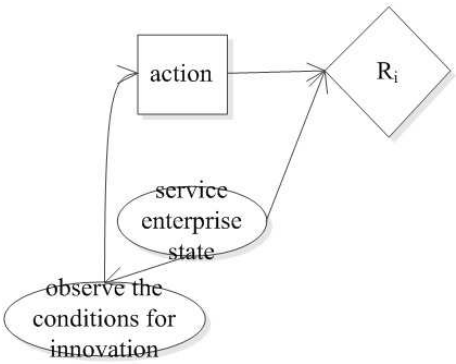


Figure 4: A Service Enterprise ID Model

to express the multi-agent decision-making process.

4.1 Interactive ID

Considering the service enterprise as the deciding agent, we generate an interactive ID as shown in Figure 5 [8]. Panel (a) shows the interactive ID of the l level, in which the model nodes $M_{j,l-1}$ and $M_{k,l-1}$ are represented by hexagons. In addition to the model nodes, there are also random nodes such as "*Action j*", "*Action k*" and dashed lines between them (called policy chains). The nodes "*Action j*", "*Action k*" are used to denote the probability distribution of the actions of j and k. The future state is the three-way interaction state, in which the observation i node represents the observation value of agent i, and the R_i node represents the value function of agent i.

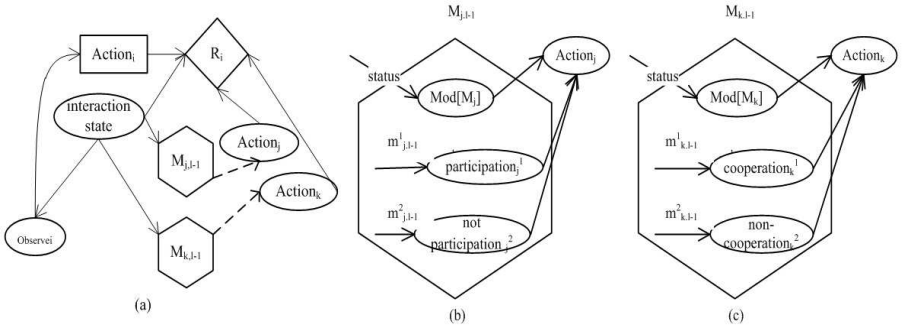


Figure 5: Interactive ID

Panel (b) shows the internal structure of the model nodes $M_{j,l-1}$. The model node $M_{j,l-1}$ contains the model of j on the *l-1* level, denoted as $m^1_{j,l-1}$, $m^2_{j,l-1}$. To solve each model, we obtain the corresponding optimal action set and its probability distribution, represented as "participation$_j^1$" and "non-participation$_j^2$" in Panel (b). Assume that the optimal action set obtained by solving $m^1_{j,l-1}$ is OPT, then the probability distribution of the action $Pr(a_j \in Action_j^1) = 1/|OPT|\,(a_j \in OPT)$, otherwise 0. Therefore, the action nodes of *l-1* level interactive ID are transformed into the random nodes ("participation$_j^1$" and "non-participation$_j^2$" in the figure), and the probability distribution of the action a_j corresponding to the optimal policy is generated as $1/|OPT|\,(a_j \in OPT)$. In Figure 4, the corresponding action nodes ("participation$_j^1$" and "non-participation$_j^2$") and the node $Mod[M_j]$ are the parent nodes of the node "*Action j*". Since each action node corresponds to a model, the number of action nodes in the model nodes $M_{j,l-1}$ is the same as the number of mod-

els in the model node. The conditional probability distribution of the random node "*Action j*" uses the probability distribution of the action node "participation$_j^1$" or "non-participation$_j^2$". Furthermore, the value of *Mod[M_j]* selects the distribution of which node. Use the value of *Mod[M_j]* to distinguish between the different models of *j*. For example, when the value of *Mod[M_j]* is $m_{j,l-1}^1$, the random node "*Action j*" adopts the distribution of node "participation$_j^1$"; when the value of *Mod[M_j]* is $m_{j,l-1}^2$,the node "*Action j*" adopts "non-participation$_j^2$". The probability distribution of the node *Mod[M_j]* is the belief of the agent *i* to the *j* model. Panel (b) clarifies the concept of the strategy chain and we find that the strategy chain can be represented by the traditional ID arc (or edge). Panel (c) is similar to (b).

4.2 I-DID

The interactive ID is expanded in time to obtain I-DID, as shown in Figure 5 [8]. The dashed dotted arrow between model nodes is called the model update chain. The conditional probability distribution on random nodes "Interaction state^{t+1}" and "Observe^{t+1}" correspond to the state transfer function Ti and the observation function O_i in I-POMDP [10].

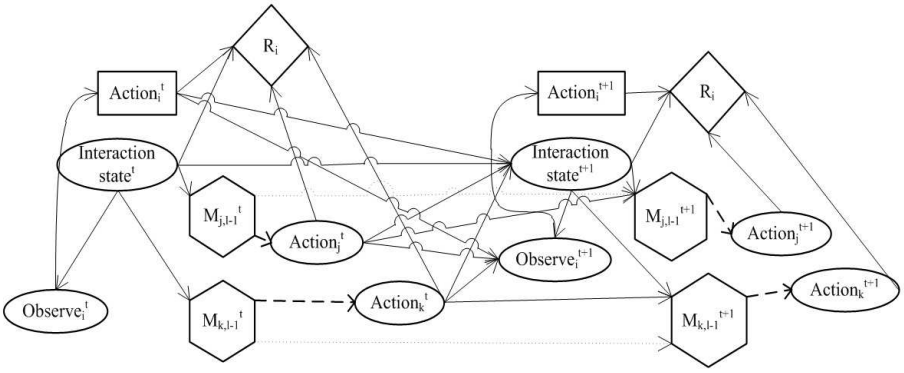


Figure 6: An I-DID of Service Innovation

Figure 6 [8] shows the implementation of the model update chain in I-DID with customer agent *agent j* as an example. Assuming that both models $m_{j,l-1}^1$ and $m_{j,l-1}^2$ of *agent j* at time *t* in level *l* produce only one optimal action, and assuming that the optimal actions are "participation" and "non-participation" respectively, then *j* may obtain two different observations: "innovation condition mature$_j^1$" and "innovation condition immature$_j^2$," then the model node contains four new models at time *t+1* ($m_{j,l-1}^{t+1,1}$, $m_{j,l-1}^{t+1,2}$, $m_{j,l-1}^{t+1,3}$, $m_{j,l-1}^{t+1,4}$ in the figure).Since *j* updates its belief through its actions and observations, the belief of these models is different. These four models represent the I-DIDs or DIDs on the *l-1* level, whose action nodes correspond to the random nodes "Action$_j^1$", "Action$_j^2$", "Action$_j^3$" and "Action$_j^4$" respectively. The supplier agent *k* is similar to the customer agent *j*.

4.3 DMU method

The solution of I-DID is very complicated. This is mainly because the number of candidate models in I-DID increases exponentially with time (as can be seen from Figure 7). The state space of S is the interactive state between agents. It is very large, and increases rapidly with the increase in the modeling nested layer number among agents. The existing solution algorithm is based on the principle of equivalent behavior (BE). The basic idea is that the behaviors of the two candidate models on agent *j* predicted by agent *i* are exactly the same, so we consider that the behaviors of the two models are equivalent. We can delete one model and assign the

belief of the deleted model to the other model. The policy tree is used to represent the solution model, as seen in Figure 8. The left is the DID of a single agent j in two time slices, and the right is the corresponding policy tree. The policy tree of T time slices is represented by $\pi^T_{m_{j,l-1}}$; therefore, $OPT(m_{j,l-1}) \triangleq \pi^T_{m_{j,l-1}}$, in which $OPT(.)$ is the solution of the model. In the policy tree, each branch extends from the root node to the leaf node and is an action-observation sequence, represented by $h_j^{T-1}=\{a_j^t, o_j^{t+1}\}_{t=0}^{T-1}$. For example, there are two branches in Figure 7 [8]: "participation→ innovation condition mature→ participation" and "participation→ innovation condition immature→ non-participation".

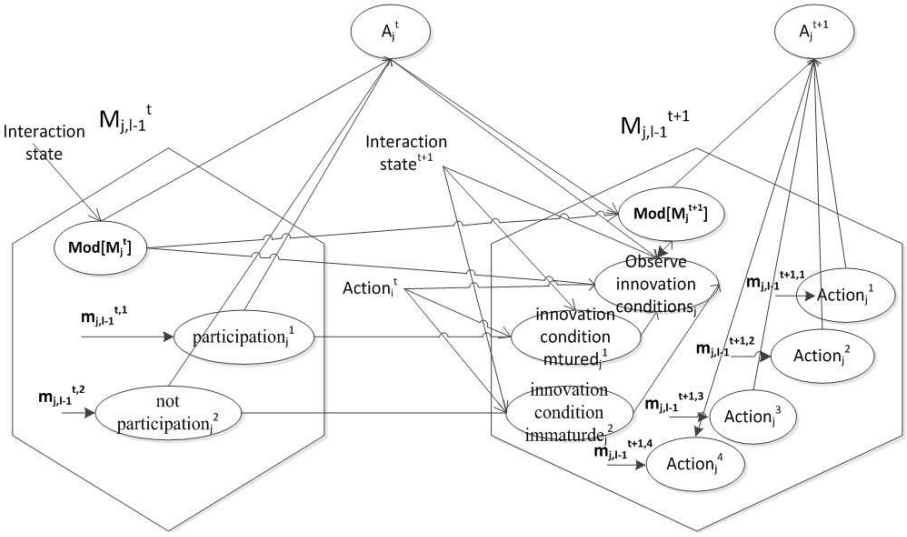


Figure 7: Realization of Model Update Chain in I-DID

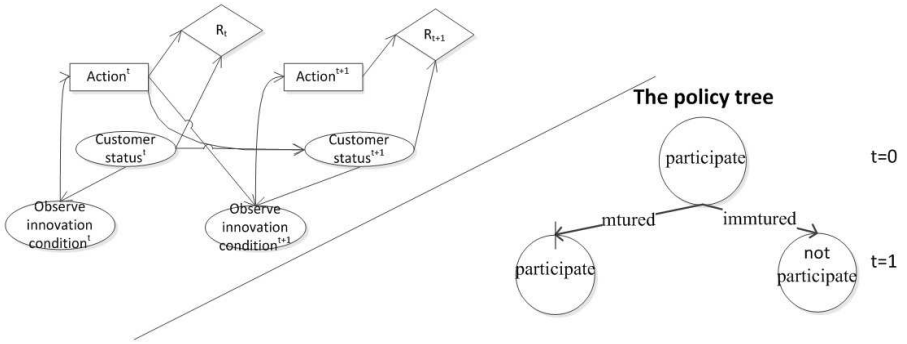


Figure 8: The Policy Tree and The Branch of ID

The bottom-up approach is used to solve I-DID, and the interactive ID to obtain the policy tree. Then we contrast the policy tree from top to bottom to check whether the models are behaviorally equivalent. The solution of the level 0 model is used by the traditional DID method; it provides a probability distribution for the corresponding action node of the level 1 I-DID.

One of the most effective methods based on the behavioral equivalence principle is the DMU method, which was proposed by Doshi and Zeng (2009) [7]. It obtains the minimum model set by merging the policy tree.

The minimum model set: $\hat{M}_{j,l-1}$ is the minimum model set of $M_{j,l-1}$, that is, for any $m_{j,l-1} \in \hat{M}_{j,l-1}$, there is no other model $m'_{j,l-1} \in \hat{M}_{j,l-1}/m_{j,l-1}$ and $OPT(m'_{j,l-1})=OPT(m_{j,l-1})$. The key of the DMU method is to get the minimum model set $\hat{M}_{j,l-1}$ and $\hat{M}_{k,l-1}$, that is, merge

the policy tree from top to bottom, and obtain the policy graph [19].

Formally, let $\hat{m}_{j,l-1} \in \hat{M}_{j,l-1}$, then:

$$\hat{b}_i(\hat{m}_{j,l-1}|s) = \sum_{m_{j,l-1} \in \mathbb{M}_{j,l-1}} b_i(m_{j,l-1}|s) \tag{1}$$

where $b_{j,l-1}$ is the level l-1 belief, and $\mathbb{M}_{j,l-1} \subseteq M_{j,l-1}$ is the Build Extended model set in which the representative $\hat{m}_{j,l-1}$ is included.

Correspondingly, let $\hat{m}_{k,l-1} \in \hat{M}_{k,l-1}$, then:

$$\hat{b}_i(\hat{m}_{k,l-1}|s) = \sum_{m_{k,l-1} \in \mathbb{M}_{k,l-1}} b_i(m_{k,l-1}|s) \tag{2}$$

According to the DMU method solution, the multi-agent I-DID is modeled and solved in this paper. Assume that the 0-level models of agent j and k are both the DID, and x_1=0.5, x_2=-0.5, x_3=0.5, y_1=0.7, y_2=-0.7, y_3=0.7, z_{11}=8, z_{12}=-5, z_{13}=2, z_{14}=0, z_{21}=8, z_{22}=-5, z_{23}=2, z_{24}=0, z_{31}=8, z_{32}=-5, z_{33}=2, z_{34}=0, r_{11}=8, r_{12}=5, r_{13}=2, r_{14}=5, r_{15}=3, r_{16}=2, r_{17}=2, r_{18}=1, r_{21}=-1, r_{22}=-3, r_{23}=-5, r_{24}=-3, r_{25}=1, r_{26}=0, r_{27}=0, r_{28}=-1, m=0.85, n=0.85, and x=0.5, in which the interactive state number is 2, the action number of agent i, j and k are all 2, the observation number of agent i is 8, and the observation number of agent j and k are both 2. The initial model numbers N=25 and N=50 are used to solve the model respectively [16].

Table 10: DMU Method's Speed

Level 1	N	Agent Number	T	DMU Times
Solve The 0-Level Model	25	2	3	124
	50	2	3	215
	25	2	5	652
	50	2	5	1247
	25	3	3	96
	50	3	3	160
	25	3	5	552
	50	3	5	987
BE Model	25	2	3	24
	50	2	3	24
	25	2	5	52
	50	2	5	52
	25	3	3	132
	50	3	3	132
	25	3	5	9087
	50	3	5	9087

In Table 10, N represents the initial model number, T represents the number of time slices, and the DMU time unit is seconds. It can be seen that the time required to solve and extend the multi-agent I-DID model of three agents is much larger than that of two agents. In other words, if we only build a model only between the service enterprise and customer or between the service enterprise and supplier, the process is much faster and the length of time slice is much longer. However, the service enterprise must consider the combined effect of both the supplier

and customer; it must establish three agent models. In the 0-level model, the number of agents has no relationship with the speed; this is because we assume that the three agents I-DID of the 0-level model is the DID. So it is the same as the two agent model; only, we need to consider another agent, such as customer or supplier.

The solution of the model is Figure 9, in which (a) is the result of three time slices; (b) is the result of five time slices; DMU 25 is the result of the DMU method, where the initial model is 25; DMU 50 is also considered. The abscissa represents the value of , and the ordinate represents the average value obtained by the service enterprise. When T=3 in the figure, each average is the average of 10 one thousand game income values. When T=5, each average is the average of 10 one hundred game income values. It can be seen from the figure that the average value increases with decreasing, until it approaches the exact algorithm. When the initial model N=50, the average value is generally higher than N=25, that is to say, the more models there are, the greater the gain. Furthermore, the average value increases as the time slice increases, and the accuracy of decisions improves. But with the increase of the time slice and the initial model, the solution speed becomes progressively slower until it cannot be solved. Therefore, we need to choose the appropriate time slice and initial model number.

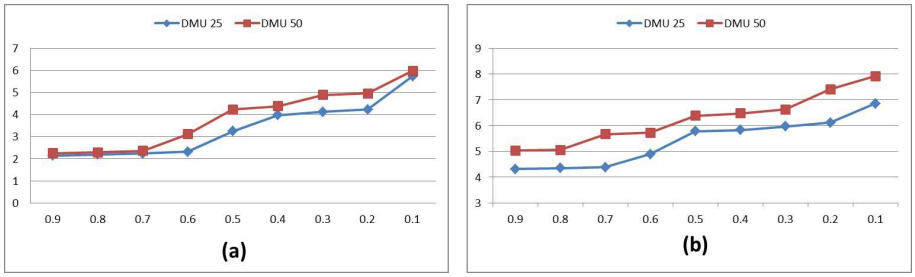


Figure 9: Comparison of Model Solving Results

It can be seen from the results of the above model that the quality of decisions made by service enterprises, customers, and suppliers in service innovation is related to the number of time slices and the initial model. Because I-DID is an individual decision-making method, it considers a problem from a single agent perspective, unlike the Nash equilibrium as a whole, such as game theory, since the service enterprise is the main agent in this case. Of course, the customer or supplier can also be considered the main agent. In the process of model solving, suppose the environment state is "interested", the customer decision action is "participation" and the supplier decision action is "cooperation", the service enterprise's decision-making action should be "innovation;" in such a case, making the decision will obtain the maximum return value of 8. Suppose the environment state is "disinterested", the customers' decision action is "non-participation" and the supplier's decision action is "non-cooperation", then the service enterprise's decision-making action should be "non-innovation"; "innovation" will be the worst action as the probability of innovation failure is great, and we obtain the maximum penalty value of -5. The setting of the data is the hope of the enterprise to innovate; this is mainly because there is no way out without innovation [13]. There could be mistakes in the innovation process, but it also allows enterprises to accumulate experience. Consequently, the reward of innovation is greater than the punishment of innovation failure. Setting x=0.5 means that customer, service enterprise, and supplier initially have no preference for the maturity of the innovation condition. This is an a priori Bayesian probability that becomes gradually more ineffective as the time slice increases. When m=0.85, it means that the customer wants to participate with an accuracy of 0.85, and the corresponding value of n indicates the accuracy of the observation of supplier cooperation. At each time slice, the service enterprise will make a decision about whether to

innovate or not, and will then obtain a corresponding reward or penalty value. With a longer time slice, the communication between the service enterprise and the customer or supplier increases the gradual understanding of the environment and the opponent increases the probability of making the right decision, and the return value increases. The more models for each agent, the bigger the return value and the ability to solve the problem is greatly tested. So the number of initial candidate models is the key to the quality of the decision-making.

5 Conclusion

First, service innovation is not an end in itself, but a means to an end, because the enterprise needs the benefit. The enterprise should decide whether or not to innovate for its own benefit.

Second, the initial degree of interest from the customer and supplier is only an a priori probability. After many decisions, it tends to a stable value, which means there is no direct relationship between the initial interest of the customer and supplier and the final innovation results. Different initial values only affect the speed of the solution.

Third, if the service enterprise takes more time to make the decision, the effect of the innovation will be more ideal. As time progresses, multiple decisions will make the effect of the innovation more desirable.

Lastly, the more customer and supplier factors that the service enterprise considers, the more reasonable its decision will be. At the same time, we should not blindly consider too many factors for the sake of rationality, since this may slow down the decision-making speed of the enterprise, thus affecting the efficiency of service innovation.

In summary, a service enterprise should consider its own innovation conditions and then continue communicating with the customer and supplier, predicting possible reactions between them, correcting the innovation decision, and improving the quality and benefit of service innovation.

Funding

This research was supported by the National Natural Science Foundation of China (No. 61562033); the Natural Science Foundation of Jiangxi Province (No. 20142BBE50038, No. 20161BBF60090, No. 20171BAB202022); and the Social Science Planning Project (No. 16GL20) of Jiangxi Province.

Bibliography

[1] Andres, B.; Poler, R.; CamarinhaMatos, L. M.; Afsarmanesh, H.(2017); A Simulation Approach to Assess Partners Selected for a Collaborative Network;*International Journal of Simulation Modelling*, 16(3), 399–411, 2017.

[2] Aranda, D.A.; Molina–Fernández, L.M. (2002); Determinants of innovation through a knowledge-based theory lens, *Industrial Management & Data Systems*, 102(5), 289–296, 2002.

[3] Bilderbeek, R.; Hertog, P.D.; Marklund, G.; Miles I. (1998); Services in innovation - Knowledge intensive business services (KIBS) as co-producers of innovation, *Synthesis report, SI4S project*, 1998.

[4] Burns, T.E.; Stalker, G.M. (2009); The Management of Innovation, *Quarterly Journal of Economics*, 109(4), 1185–1209, 2009.

[5] Dai, Y.; Wu, W.; Zhou, H.B.; Zhang, J.; Ma, F.Y. (2018); Numerical Simulation and Optimization of Oil Jet Lubrication for Rotorcraft Meshing Gears, *International Journal of Simulation Modelling*, 17(2), 318–326, 2018.

[6] Dai, Y.; Zhu, X.; Zhou, H.; Mao, Z.; Wu, W. (2018); Trajectory Tracking Control for Seafloor Tracked Vehicle By Adaptive Neural-Fuzzy Inference System Algorithm, *International Journal of Computers Communications & Control*, 13(4), 465–476, 2018.

[7] Doshi, P.; Zeng, Y.F. (2009); Improved Approximation of Interactive Dynamic Influence Diagrams Using Discriminative Model Updates, *International Conference on Autonomous Agents and Multiagent Systems*, 907–914, 2009.

[8] Doshi, P.; Zeng, Y.F.; Chen, Q. (2009); Graphical models for interactive POMDPs: representations and solutions, *Autonomous Agents and Multi-Agent Systems*, 18(3), 376–416, 2009.

[9] Gallouj, F. (2015); Innovation in the Service Economy-The New Wealth of Nations, *Post-Print*, 2015.

[10] Gmytrasiewicz, P.J.; Doshi, P. (2005); A Framework for Sequential Planning in Multi-Agent Settings, *Journal of Artificial Intelligence Research*, 24(1), 49–79, 2005.

[11] Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. (1998); Planning and Acting in Partially Observable Stochastic Domains, *Artificial Intelligence*, 101(1-2), 99–134, 1998.

[12] Koller, D.; Milch, B.(2001) ; Multi-agent influence diagrams for representing and solving games, *International Joint Conference on Artificial Intelligence*, 1027–1034, 2001.

[13] Ojasalo, K.; Koskelo, M.; Nousiainen, A.K. (2015), Foresight and Service Design Boosting Dynamic Capabilities in Service Innovation, *The Handbook of Service Innovation*, Springer London, 193–212, 2015.

[14] Pan, Y.H.; Luo, J.; Zeng. Y.F. (2012); Modeling Method of Multi - Agent Interactive Dynamic Impact Diagram, *Journal of Xiamen University (Natural Science)*, 51(6), 985–990, 2012.

[15] Pan, Y.H.; Zeng, Y.F. (2018); Interactive Dynamic Influence Diagram Research Summary and New Solutions on Top-K Model Selection, *Chinese Journal of Computer*, 41(1), 28–46, 2018.

[16] Pan, Y.H.; Zeng, Y.F.; Xiang, Y.P.; Sun, L.; Chen, X.F. (2015); Time-Critical Interactive Dynamic Influence Diagram, *International Journal of Approximate Reasonin*, 57, 44–63, 2015.

[17] Ryu, H.S.; Lee, J.N.; Choi, B. (2015); Alignment Between Service Innovation Strategy and Business Strategy and Its Effect on Firm Performance: An Empirical Investigation, *IEEE Transactions on Engineering Management*, 62(1), 100–113, 2015.

[18] Zeng, Y.F.; Doshi, P. (2012); Exploiting Model Equivalences for Solving Interactive Dynamic Influence Diagrams, *Journal of Artificial Intelligence Research*, 43(1), 211–255, 2012.

[19] Zeng, Y.F.; Doshi, P.; Chen Y.K.; Pan Y.H.; Mao H.; Chandrasekaran M. (2016) ; Approximating behavioral equivalence for scaling solutions of I-DIDs, *Knowledge and Information Systems*, 49(2), 511–552, 2016.

[20] Zhang, H.Q.; Lu, R.Y. (2013); The impact of customer participation on employee's innovation behavior in the service industry, *Science Research Management*, 34(3), 99–105, 2013.

[21] Zhang, R.Y.; Liu, X.M.; Wang, H.Z.; Nie, K. (2010); Customer-Firm Interaction and Service Innovation Performance: A Perspective of Organizational Learning from Customers, *Chinese Journal of management*, 7(2), 218-224, 2010.

Author index