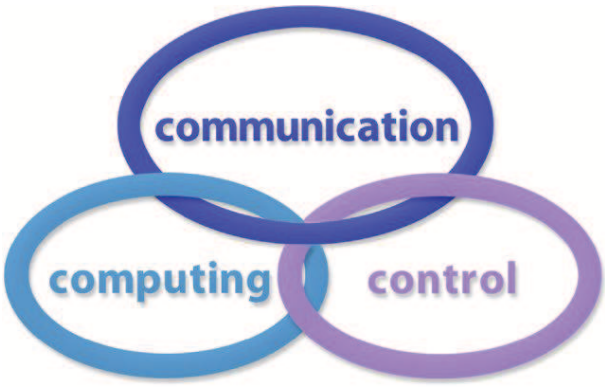
INTERNATIONAL JOURNAL

of

COMPUTERS COMMUNICATIONS & CONTROL

ISSN 1841-9836



A Bimonthly Journal
With Emphasis on the Integration of Three Technologies

Year: 2018 Volume: 13 Issue: 3 Month: June

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 Thomson Reuters/Clarivate Analytics and is indexed in ISI Web of Science/Knowledge: Science Citation Index Expanded.

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

  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)**.
  Impact Factor/5 years in JCR: 0.436(2012), 0.622(2013), 0.739(2014), 0.635(2015), **1.193(2016)**.

- Since 2008 IJCCC is indexed by Scopus: **CiteScore 2016 = 1.06**.
  Subject Category:
  (1) Computational Theory and Mathematics: Q4(2009, 2010, 2012, 2015), Q3(2011, 2013, 2014, 2016);
  (2) Computer Networks and Communications: Q4(2009), Q3(2010, 2012, 2013, 2015), Q2(2011, 2014, 2016);
  (3) Computer Science Applications: Q4(2009), Q3(2010, 2011, 2012, 2013, 2014, 2015, 2016).

  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).

- 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).

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

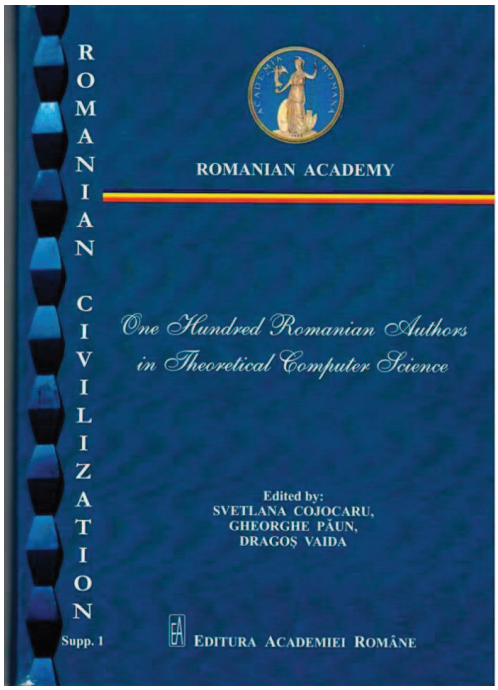
Gheorghe TECUCI
George Mason University, USA
University Drive 4440, Fairfax VA
tecuci@gmu.edu

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

EVENT DEDICATED TO THE 100TH ANNIVERSARY OF GREAT ROMANIA:
A book "One Hundred Romanian Authors in Theoretical Computer Science"



Cover of the Book

In a plenary session of International Conference on Computers Communications and Control (ICCCC2018, May 8-12, Oradea, Romania) has been launched the book "One Hundred Romanian Authors in Theoretical Computer Science" (Eds. Svetlana Cojocaru, Gheorghe Păun, Dragoş Vaida).

Gheorghe Păun: The volume includes short presentations (education, positions, scientific interests, main results, prizes/awards, selected lists of publications) of 100 Romanian researchers with contributions to theoretical computer science. One considers "Romanians" persons born in Romania (the present or the Great one) or speaking the Romanian language. There were included both classic names, such as Grigore C. Moisil and Solomon Marcus, other scientists who passed away, and researchers living in Romania or in Republic of Moldova, or born here and spread all over the world. Of course, several names remained outside the contents of the volume. The book provides a great amount of information, concerning the areas of computer science addressed by Romanian scientists, the places (countries and companies) where they work, the evolution in time of main addressed topics. It is, in fact, a (more technical) supplement of the chapter "From the history of the Romanian theoretical computer science" written by G. Păun for the volume coordinated by F.G. Filip and published in the "Romanian Civilization" series of the Romanian Academy, celebrating the 100th Anniversary of Great Romania.

Sveltana Cojocaru: The presented volume includes also the contribution of researchers from the Republic of Moldova, the area of interest of which is focused on small Turing machines, splicing systems, TVDH systems, insertion and deletion systems, networks of evolutionary processors etc. A special place is occupied by researches in the domain of membrane computing. There were obtained dozens of results regarding the computational completeness and universality for small systems. Membrane computing models with different properties allowing to investigate the relationship between different variants of membrane systems were constructed and compared with other related models like Petri nets or multiset rewriting.

This book includes brief resumes of the following authors: 1. Alhazov, Artiom; 2. Aman, Bogdan; 3. Andrei, Neculai; 4. Atanasiu, Adrian Constantin B.; 5. Balcan, Nina-Florina; 6. Bălănescu, Tudor; 7. Băutu, Elena; 8. Boian, Florian Mircea; 9. Bonchiş, Cosmin; 10. Calude, Cristian Sorin; 11. Calude, Elena; 12. Cărăuşu, Alexandru; 13. Căzănescu, Virgil Emil; 14. Câmpeanu, Cezar; 15. Ceterchi, Rodica; 16. Chira, Camelia; 17. Ciobanu, Gabriel; 18. Cojocaru, Svetlana; 19. Czeizler, Eugen; 20. Czibula, Gabriela; 21. Czibula, István-Gergely; 22. Diaconescu, Răzvan; 23. Dinu, Petrişor Liviu; 24. Dobrescu, Radu Nicolae; 25. Dumitrescu, Sorina; 26. Dziţac, Ioan; 27. Enea, Constantin; 29. Farcaş, Dezideriu Dan; 29. Frenţiu, Militon; 30. Gaindric, Constantin; 31. Georgescu, George; 32. Gheorghe, Marian; 33. Gramatovici, Radu Valer; 34. Grigoraş, Gheorghe; 35. Grigorescu, Elena; 36. Iftene, Adrian; 37. Ilie, Lucian; 38. Ionescu, Armand-Mihai; 39. Iorgulescu, Afrodita; 40. Ipate, Florentin Eugen; 41. Istrail, Sorin; 42. Istrate, Gabriel; 43. Ivanov, Sergiu; 44. Jebelean, Tudor; 45. Kari, Lila; 46. Lefticaru, Raluca-Elena; 47. Leuştean, Ioana Gabriela; 48. Leuştean, Laurenţiu; 49. Lucanu, Dorel; 50. Luchian, Henri; 51. Manea, Florin-Silviu; 52. Marcu, Daniel; 53. Marcus, Solomon; 54. Masalagiu, Cristian-Dumitru; 55. Mateescu, Alexandru; 56. Maruşter, Ştefan; 57. Mercaş, Robert; 58. Mihalcea, Rada Flavia; 59. Mitrana, Victor; 60. Moisil, Grigore C.; 61. Moisil, Ioana I.; 62. Moldovan, Grigor; 63. Negru, Viorel; 64. Nicolescu, Radu; 65. Orman, Gabriel V.; 66. Pavel, Ana Brânduşa; 67. Pătraşcu, Mihai; 68. Păun, Paul Andrei; 69. Păun, Gheorghe; 70. Pârv, Bazil; 71. Petcu, Dana; 72. Petre, Ion; 73. Pop, Horia Florin; 74. Popa, Alexandru; 75. Popescu, Andrei; 76. Rezuş, Adrian; 77. Rogojin, Iurie; 78. Rogojin, Vladimir; 79. Rudeanu, Sergiu; 80. Sburlan, Dragoş-Florin; 81. Simovici, Dan; 82. State, Luminiţa-Doina; 83. Stoean, Cătălin-Liviu; 84. Stoean, Ruxandra; 85. Streinu, Ileana; 86. Şerbănuţă, Traian Florin; 87. Ştefănescu, Alin; 88. Ştefănescu, Doru; 89. Ştefănescu, Gheorghe; 90. Tîrnăucă, Cristina; 91. Tomescu, Ioan 92. Trăuşan-Matu, Ştefan 93. Ţâmbulea, Leon; 94. Ţiplea, Ferucio Laurenţiu; 95. Ţuţu, Ionuţ; 96. Vaida, Dragoş Alexandru; 97. Văduva, Ion; 98. Verlan, Sergey; 99. Zaharie, Daniela; 100. Zimand, Marius.

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

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, Volume 13, Issue 3, June 2018.

Contents

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 303-320, June 2018.

On Distributed Solution to SAT by Membrane Computing

H.N. Adorna, L. Pan, B. Song

Henry N. Adorna
Algorithms and Complexity Lab.
Department of Computer Science
University of the Philippines Diliman
Diliman 1101 Quezon City, Philippines
hnadorna@up.edu.ph

Linqiang Pan*
1. Key Laboratory of Image Information Processing and
Intelligent Control of Education Ministry of China
School of Automation
Huazhong University of Science and Technology
Wuhan 430074, Hubei, China
2. School of Electrical and Information Engineering
Zhengzhou University of Light Industry
Zhengzhou 450002, Henan, China
*Corresponding author: lqpan@mail.hust.edu.cn

Bosheng Song
Key Laboratory of Image Information Processing and
Intelligent Control of Education Ministry of China
School of Automation
Huazhong University of Science and Technology
Wuhan 430074, Hubei, China
boshengsong@hust.edu.cn

Abstract: Tissue P systems with evolutional communication rules and cell division (TPec, for short) are a class of bio-inspired parallel computational models, which can solve NP-complete problems in a feasible time. In this work, a variant of TPec, called k-distributed tissue P systems with evolutional communication and cell division (k-$\Delta_{TP_{ec}}$, for short) is proposed. A uniform solution to the SAT problem by k-$\Delta_{TP_{ec}}$ under balanced fixed-partition is presented. The solution provides not only the precise satisfying truth assignments for all Boolean formulas, but also a precise amount of possible such satisfying truth assignments. It is shown that the communication resource for one-way and two-way uniform k-P protocols are increased with respect to k; while a single communication is shown to be possible for bi-directional uniform k-P protocols for any k. We further show that if the number of clauses is at least equal to the square of the number of variables of the given boolean formula, then k-$\Delta_{TP_{ec}}$ for solving the SAT problem are more efficient than TPec as show in [39]; if the number of clauses is equal to the number of variables, then k-$\Delta_{TP_{ec}}$ for solving the SAT problem work no much faster than TPec.
Keywords: Membrane computing, distributed P system, SAT, communication complexity

1 Introduction

Since the research area of membrane computing was proposed in 1998 [8,19], the research lines about computation power of various variants of P systems [25, 26, 26, 33, 37], their applications [8,24,29,36] and implementation issues [15,16,32] have been investigated widely. Several solution

approaches and techniques have been presented in the literature for solving Satisfiability Problem (SAT) using many variants of P systems [11, 12, 27, 30]. Each of these variants of P systems solving SAT provided better solutions than the conventional model in terms of time efficiency or computational time complexity [18,31,38]. Most of them are benefited from the nondeterministic maximal parallelism of P systems and its ability to produce exponentially many cells or regions (in linear time) in a computation [35].

Evolution and communication are the core operations in the solutions offered by the variants of P systems, where communication allows objects to be transmitted to the other regions/cells for further processing. In [14], Hernandez, et al. provided a solution to 3-SAT, where the amount of communications is measured from a dynamic communication complexity perspective [1]. Several results might have been reported using communication as a measure of complexity [2, 5, 6], but the analyses would be dissimilar to that of [1].

In order to capture the concept of communication complexity as introduced by A.C. Yao in [34], Gh. Păun, et al. [21] introduced and defined a so-called *dP scheme*, where the input of a k-dP scheme is partitioned and these parts are distributed to the participating components. Necessarily, these components need to communicate to solve a problem, and the so-called *inter-component communication rules* are introduced as a new kind of set of communication rules.

Computation done by distributed model in this work depends on the agreed upon partition of the input among the participating P systems. A partition is called *balanced partition* if the number of objects or the length of the part of the input assigned to each participating component P systems is almost equal. Let w be an input for a distributed system with two components. The partition w_1 and w_2 of w, where w_i is assigned to Π_i, $i = 1, 2$ is *balanced* if and only if $||w_1| - |w_2|| \leq 1$. In particular, it is a *balanced fixed-partition,* if $w = w_1 w_2$; otherwise, we have *unbalanced (fixed-)partition.*

The distributed P systems halt if and only if all participating component P systems halt. If a distributed system halts in a specified accepting configuration, then it accepts/decides/solves a problem.

Since k-dP scheme was introduced, some of the results on this variant were reported in [7,10,23]. In [4], Buño, et al. introduced a distributed solution for n-queens problem as presented in Naranjo, et al. [13]. Indeed, in [4], the reduction of the n-queens problem to SAT of a dP scheme were used, where the components are P systems with active membranes.

Recently, Buño, et al. [3] capitalized the power of tissue P systems (*TPec*) with evolutional communication rules and cell division introduced in [39], in proposing a distributed solution to SAT. In particular, they defined a so-called 2-dTP_{ec}. The paper claimed a decent advantage of distributed solution compared to the non-distributed one with respect to the results reported in [39]. Also, only two participating components in systems were considered working on a balanced partition.

In this work, the solution presented in [3] will be revisited and some other insights into the consequences of our results are provided. Some other relevant issues related to communication complexity in distributed tissue P systems with evolutional communication and cell division are proposed.

Contributions of the present work are summarized as follows:

(a) A variant of tissue P systems with evolutional communication and cell division, called k-distributed tissue P systems with evolutional communication and cell division (k-$\Delta_{TP_{ec}}$, for short) and the corresponding recognizer version are proposed.

(b) A uniform solution to the SAT problem by k-$\Delta_{TP_{ec}}$ under balanced fixed-partition is presented. The solution provides not only the precise satisfying truth assignments for all Boolean formulas, but also a precise amount of possible such satisfying truth assignments.

(c) The communication resource for one-way and two-way uniform k-P protocols is increased with respect to k; while a single communication is shown to be possible for bi-directional uniform k-P protocols for any k.

(d) We further show that if the number of clauses is at least equal to the square of the number of variables of the given boolean formula, then k-$\Delta_{TP_{ec}}$ for solving the SAT problem are more efficient than TPec as show in [39]; if the number of clauses is equal to the number of variables, then k-$\Delta_{TP_{ec}}$ for solving the SAT problem works no much faster than TPec.

The rest of this work is organized as follows. Section 2 provides the preliminaries of a dP scheme, introducing the concepts of P protocols and balanced (fixed) partition and the communication resources used in the analyses of the solution to SAT. In Section 3, distributed tissue P systems with evolutional communication and cell division or k-Δ_{TPec} are defined. Solution to SAT using 2-Δ_{TPec} is presented in Section 3, while solution to SAT using 3-Δ_{TPec} is given in Section 5. Remarks on the relative efficiency of distributed solutions are provided in Section 4. Finally, conclusions and discussions are given in Section 5.

2 Preliminaries

In this section, the notions of k-dP scheme and communication complexity of P systems are presented, then tissue P systems with cell division and evolutional communication rules are introduced [39].

Definition 1. A k-dP scheme ($k \geq 2$) is a tuple

$$k\text{-}\Delta_{\Pi} = (\Gamma, 0, \Pi_1, \Pi_2, \ldots, \Pi_k, R_{\Delta}),$$

where

- Π is a fixed variant of P system;
- Γ is an alphabet of objects in the whole system Δ;
- 0 is the common/shared environment of Π_i;
- Π_i for $i = 1, 2, \ldots, k$ are P systems of the fixd variant Π with Γ as working alphabet, skin membranes or local environments of each P system will be labelled injectively as $s_1, s_2, \ldots, s_k$;
- R_{Δ} is a finite set of rules of the form $(s_i, u/v, s_j)$, where $1 \leq i, j \leq k$, $i \neq j$, and $u, v \in \Gamma^*$, such that $uv \neq \lambda$. We denote by $|uv|$ the *weight* of the rule $(s_i, u/v, s_j)$. This antiport-like communication rule is called inter-component communication rule.

The mechanism by which a k-dP scheme performs its computation could be found in [21]. In particular, an input for a k-dP scheme is partitioned into k parts and distributed one part to each of the k components of the dP scheme. Thus, communication to solve the problem is inevitable. In this paper, definition of *balanced* and *unbalanced* partition of an input is provided.

Definition 2. A partition $\{P_1, P_2, \ldots, P_k\}$ is called balanced partition if and only if for all i, P_i have the same size or at most a difference of 1. Otherwise, we call it an unbalanced partition.

Definition 3. We call a k-balanced (unbalanced) partition $\{P_1, P_2, \ldots, P_k\}$ (resp., a k-balanced (unbalanced) fixed-partition) if and only if the k-partition of input is done from left to right with respect to the (resp., fixed) ordering of the input.

Cooperation between component P systems of a dP scheme is defined in the set R_Δ of inter-component communications. R_Δ specifies the mode of communication protocol of a dP scheme. In what follows, *k-P protocols* for a k-dP scheme are defined.

Definition 4. Let k-Δ_Π be a k-dP scheme.

- k-Δ_Π is called 1-way k-P protocol if and only if R contains only rules of the form $(s_i, u/\lambda, s_j)$.
- k-Δ_Π is called 2-way k-P protocol if and only if R contains rules of the form $(s_i, u/\lambda, s_j)$ and $(s_i, \lambda/v, s_j)$.
- k-Δ_Π is called bi-directional k-P protocol if and only if R contains rules of the form $(s_i, u/v, s_j)$, $(s_i, u/\lambda, s_j)$ and $(s_i, \lambda/v, s_j)$.

A k-dP scheme computes as follows. All component P systems of a k-dP scheme are aware of the problem that they are solving. Each component P system knows only the part of the input assigned to them. We allow each component P system to perform computation to the input part known to them. To solve the problem, component P systems must communicate with respect to a particular protocol.

Definition 5. A configuration δ_j of k-Δ_Π is a vector in $\mathcal{C} = 0 \times \mathcal{M}_{\langle 1,0\rangle} \times \mathcal{M}_{\langle 2,0\rangle} \times \ldots \times \mathcal{M}_{\langle k,0\rangle}$, where 0 is the common environment, and $\mathcal{M}_{\langle i,0\rangle}$ are sets of multisets of objects in $\langle i, 0\rangle$, $i = 1, 2, \ldots, k$. In particular, $\delta_j = (m_{0_j}, m_{1_j}, m_{2_j}, \ldots, m_{k_j}) \in \mathcal{C}$ indicates that at time j, $m_0 \in 0$ and $m_i \in \mathcal{M}_{\langle i,0\rangle}$, $i = 1, 2, \ldots, k$.

Definition 6. A computation of k-Δ_Π is the transition of configurations represented by a sequence $\delta : \delta_0 \Rightarrow \delta_1 \Rightarrow \cdots \Rightarrow \delta_h$, where δ_0 is the initial configuration, and δ_h is the final or halting configuration. The initial configuration δ_0 is a vector of initial multisets contained in 0 (local environment of component P systems). δ is a halting computation if and only if δ_h is a configuration, where one of the objects `yes` or `no` is contained in some (specified) membrane of the system.

k-Δ_Π has an accepting configuration if and only if δ is a halting computation and at configuration δ_h, object `yes` appeared in a specified membrane in the system. δ is a rejecting computation if and only if at δ_h object `no` appeared.

Definition 7. A language L is decided by k-Δ_Π, $L = L(k\text{-}\Delta_\Pi)$ if and only if for every input w over some alphabet, there is a halting computation δ of k-Δ_Π that decides on $w \in L$.

In this work, the existence of at least an object `yes` in 0 implies an affirmative decision on a problem, while the appearance of at least an object `no` in 0 connotes a negative decision.

All component P systems with the same and uniform procedure in processing input part, that is to say, component P systems of a k-dP scheme would be all the same, is called a *uniform k-dP scheme.* If we have a uniform k-Δ_Π, then k-P protocol is called *uniform k-P protocol.*

This work focuses on the amount of communications used by component P systems in deciding the satisfiability of some formula φ in conjunctive normal form (CNF). Thus, the following notions are used [1].

Definition 8. [21] Let Δ be a dP scheme, $\delta : \delta_0 \Rightarrow \delta_1 \Rightarrow \cdots \Rightarrow \delta_h$ is a halting computation in Δ, where δ_0 is the initial configuration. Then for each $i = 0, 1, \cdots, h-1$, we have the following parameters:

- $ComN(\delta_i \Rightarrow \delta_{i+1}) = \begin{cases} 1, & \text{if a communication rule is used in this transition,} \\ 0, & \text{otherwise,} \end{cases}$

- $ComR(\delta_i \Rightarrow \delta_{i+1})$ denotes the number of communication rules used in this transition,
- $ComW(\delta_i \Rightarrow \delta_{i+1})$ denotes the total weight of the communication rules used in this transition.

The above mentioned parameters can also be used to measure computations, results of computations, systems and languages (problems).

Definition 9. Let $L(\Delta)$ be the set of strings accepted by Δ. For $X \in \{N, R, W\}$, we define:
$ComX(\delta) = \sum_{i=0}^{h-1} ComX(\delta_i \Rightarrow \delta_{i+1})$, δ is a halting computation,
$ComX(w, \Delta) = min\{ComX(\delta) \mid \delta$ is an accepting computation of Δ for $w\}$,
$ComX(\Delta) = max\{ComX(w, \Delta) \mid w \in L(\Delta)\}$.

3 Distributed recognizer tissue P systems

In this section, we introduce the notions of (recognizer) tissue P systems with evolutional symport/antiport rules and cell division and k-distributed tissue P systems with evolutional symport/antiport rules and cell division.

Definition 10. A tissue P system (of degree $q \geq 1$) with evolutional symport/antiport rules and cell division (TPec) is a tuple

$$\Pi = (\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{out}),$$

where

1. Γ is a working alphabet of objects;
2. $\mathcal{E} \subseteq \Gamma$ is the set of objects initially located in the environment;
3. $\mathcal{M}_i$, $1 \leq i \leq q$, are finite multisets over Γ;
4. R is a finite set of rules of the following forms:
   (a) Evolutional communication rules:
      i. Evolutional symport rules: $[u]_i[\ ]_j \rightarrow [\ ]_i[u']_j$, for $1 < i \leq q, 0 < j \leq q, i \neq j; u \in \Gamma^+, u' \in \Gamma^*$ or $i = 0, 1 < j \leq q; u \in \Gamma^+, u' \in \Gamma^*$, and there exists at least one object $a \in \Gamma \setminus \mathcal{E}$, which is in cell $i = 0$. The length of an evolutional symport rule is defined as $|u| + |u'|$.
      ii. Evolutional antiport rule: $[u]_i[v]_j \rightarrow [v']_i[u']_j$, where $0 \leq i \neq j \leq q$, $u, v \in \Gamma^+, u', v' \in \Gamma^*$. The length of an evolutional antiport rule is defined as $|u| + |u'| + |v| + |v'|$.
   (b) Division rules: $[a]_i \rightarrow [b]_i[c]_i$, where $i \in \{1, 2, \ldots, q\}$, $i \neq i_{out}, a, b, c \in \Gamma$.
5. $i_{out} \in \{1, 2, , \ldots, q\}$.

The details of the mechanism on how TPec works can be found in [39]. In what follows, we introduce the notion of recognizer TPec systems.

Definition 11. A recognizer tissue P system with evolutional symport/antiport rules and cell division of degree $q \geq 1$ is a tuple

$$\Pi = (\Gamma, \Sigma, \mathcal{E}, \mathcal{M}_1, \cdots, \mathcal{M}_q, R, i_{in}, i_{out}),$$

where

- Γ has two distinguished objects `yes` and `no`;
- $\Sigma \subseteq \Gamma$ is the input alphabet;
- $\mathcal{M}_1, \cdots, \mathcal{M}_q$ are finite multisets over $\Gamma \setminus \Sigma$;
- $i_{in} \in \{1, \cdots, q\}$ is the label of the input cell, and $i_{out} = 0$;
- all computations halt;
- either an object `yes` or an object `no` is released into the environment at the last step of any computation.

Theorem 12. *[39] Let $PMC_{TDEC(k)}$ be the set of all decision problems solvable in a uniform way and polynomial time by means of recognizer tissue P systems with cell division and evolutional communication rules of length at most k. Then, $SAT \in PMC_{TDEC(4)}$.*

In this work, at least two recognizer tissue P systems with evolutional symport/antiport rules and cell division of degree $q \geq 1$ are used to solve the SAT problem, where the input multiset is partitioned with respect to the number of component recognizer tissue P systems. Consequently, a so-called *k-distributed tissue P system with evolutional symport/antiport rules and cell division rules* (k-dTPec system) is defined.

Definition 13. A k-distributed tissue P system with evolutional symport/antiport rules and cell division or k-dTPec system is defined as follows:

$$k\text{-}\Delta_{TPec} = (\Gamma, 0, \Pi_1, \Pi_2, \ldots, \Pi_k, R_\Delta, i_{out}),$$

where

1. Γ is a set of alphabet of objects in the whole system Δ;
2. 0 is the common or shared environment of all Π_i, $i = 1, 2, \ldots, k$;
3. Π_i, $i = 1, 2, \ldots, k$ are recognizer tissue P systems with evolutional symport/antiport rules and cell division. Each Π_i has the alphabet of objects in Γ. Each cell of Π_i will be labelled $\langle i, j\rangle$, where $j = 1, 2, \ldots, d_i$, and d_i denotes the number of cells of Π_i. The external region or the environment is different for each component. We will refer to the environment of component i as the local environment of i and is denoted by the label $\langle i, 0\rangle$;
4. R_Δ is a set of finite inter-component communication rules of the form: $(\langle i, 0\rangle, u/v, \langle j, 0\rangle)$, where $\langle i, 0\rangle$ and $\langle j, 0\rangle$ are the local environments of components i and j, respectively, $u, v \in \Gamma^*$, and $|uv| \geq 1$;
5. i_{out} is the component of the dTPec designated as the output component. Only the objects produced in the output region of the system are considered as output in a halting computation of the dTPec.

Note that the k-dP tissue P system used in the next section is actually a uniform k-distributed recognizer tissue P system with evolutional symport/antiport rules.

4 Solving SAT by 2-Δ_{TPec}

In this section, a 2-Δ_{TPec} is presented that solves the satisfiability of any instance φ of the SAT problem. In particular, a uniform 2-P protocols is constructed which is based on the construction in [39].

Theorem 14. *Let φ be any instance of the SAT problem in CNF with m clauses and n variables. Then there exists 2-$\Delta_{TP_{ec}}$ deciding on satisfiability of φ under a balanced fixed-partition in $3n + 3\lceil \frac{m}{2} \rceil + 4$ steps using antiport-like inter-component communication rules.*

Proof: A 2-$\Delta_{TP_{ec}}$ for the SAT problem will be constructed such that component P systems are quite the same with that presented in [39] but with some additional rules.

Let 2-Δ_{TPec} be defined as follows:

$$2\text{-}\Delta_{TPec} = (\Gamma, 0, \Pi_1, \Pi_2, R_\Delta),$$

where

- $\Gamma = \Gamma_1 \cup \Gamma_2$,
- 0 is the common shared environment of the Π_1 and Π_2,
- Π_k, $k = 1, 2$ are recognizer tissue P systems defined as:

$$\Pi_k(\langle n, m \rangle) = (\Gamma_k, \Sigma_k, \mathcal{E}_k, \mathcal{M}_{k,1}, \mathcal{M}_{k,2}, \mathcal{M}_{k,3}, \mathcal{M}_{k,x}, R_k, i_{k_{in}} = 2, i_{k_{out}} = 0),$$

  where:

  - $\Gamma_k = \Gamma'_k \cup V$,
    $\Gamma'_k = \Sigma_h \cup \{a_i, t_i, f_i, t'_i, f'_i, \beta_i, \beta'_i \mid 1 \le i \le n\}$
    $\cup \{e_{i,j}, e'_{i,j}, \overline{e}_{i,j}, \overline{e'}_{i,j}, E_{i,j}, \overline{E}_{i,j} \mid 1 \le i \le n, 1 \le j \le m\}$
    $\cup \{d_{i,j,h}, d'_{i,j,h}, \overline{d}_{i,j,h}, \overline{d'}_{i,j,h} \mid 1 \le i \le n, 1 \le j \le m, 1 \le h \le n-1\}$
    $\cup \{b_j c_j, \overline{c}_j, c'_j, E_j \mid 1 \le j \le m\}$
    $\cup \{b_{j,h}, b'_{j,h} \mid 1 \le j \le m, 1 \le h \le n-1\}$
    $\cup \{\alpha, \alpha'_i \mid 0 \le i \le 3n + 3m_k\}$
    $\cup \{d, E_{m_k+1}, \alpha_{3n+3m_k+1}, \texttt{yes}, \texttt{no}, y, y'\}$,
    $V = \{v_l \mid 1 \le l \le 2^n,$ where n is the number of variables$\}$.
  - $\Sigma_k = \{x_{i,j}, \overline{x}_{i,j} \mid 1 \le i \le n, 1 \le j \le m\}$,
  - $\mathcal{E}_k = \{\alpha'_i \mid 0 \le i \le 3n + 3m_k\}$,
  - $\mathcal{M}_{k,1} = \{a_1, \ldots, a_n, E_1\}$, $\mathcal{M}_{k,2} = \{b_1, \ldots, b_n, d, \alpha_0\}$, $\mathcal{M}_{k,3} = \{a_1, \ldots, a_n\}$, and $\mathcal{M}_{k,x} = \emptyset$,
  - R_k is the set of rules of each $k = 1, 2$ component.
    The set of rules we will use are those set of rules $r_{1,i}$ until $r_{26,i}$, from [39] only with the following modifications and addition:

    (a) we split $r_{1,i}$ into $r_{1,i;c}$, where c indicates in which cell $r_{1,i}$ will be applied on variable i :
    $r_{1,i;3} \equiv [\, a_1 \,]_3 \to [\, \beta_1 \,]_3 [\, \beta'_1 \,]_3, \quad 1 \le i \le n,$
    $r_{1,i;1} \equiv [\, a_i \,]_1 \to [\, t'_i \,]_1 [\, f'_i \,]_1, \quad 1 \le i \le n;$

    (b) we replace r_{27} with the following:
    $r_{27} \equiv [\, v_l \, E_{m_k+1} \,]_1 [\, v_l \,]_3 \to [\, y \,]_1 [\, v_l \, \texttt{yes} \,]_3;$

(c) we replace r_{28} with the following:
$r_{28} \equiv [\, v_l \, \texttt{yes} \,]_3 [\ ]_0 \to [\ ]_3 [\, v_l \, \texttt{yes} \,]_0$;
(d) we replace r_{29} with the following:
$r_{29} \equiv [\, \alpha_{3n+3m_k+1} \, d \,]_2 [\ ]_0 \to [\ ]_2 [\, \texttt{no} \,]_0$;
(e) cell-labeling rules: $r_{p;c}$ denotes r_p is used to label cell c,
$r_{30;1} \equiv [\, \beta_1 \,]_3 [\, t'_1 \,]_1 \to [\, v_1 \,]_3 [\, v_1 \, t_1 \,]_1$,
$r_{31;1} \equiv [\, \beta'_1 \,]_3 [\, f'_1 \,]_1 \to [\, v_2 \,]_3 [\, v_2 \, f_1 \,]_1$.
$r_{32;1} \equiv [\, \beta_i \,]_3 [\, t'_i v_l \,]_1 \to [\, v_{2l} \,]_3 [\, t_i \, v_{2l} \,]_1, \quad l \in \{1, \ldots, 2^n\},\ 2 \leq i \leq n$,
$r_{33;1} \equiv [\, \beta'_i \,]_3 [\, f'_i v_l \,]_1 \to [\, v_{2l-1} \,]_3 [\, f_i \, v_{2l-1} \,]_1, \quad l \in \{1, \ldots, 2^n\},\ 2 \leq i \leq n$;
(f) cleaning rules, unused objects (during computation) are dump to cell x
$r_{34} \equiv [\, \texttt{yes} \, v_l \,]_0 [\ ]_x \to [\ ]_0 [\, \texttt{yes} \, v_l \,]_x$,
$r_{35} \equiv [\ ]_3 [\, y \,]_1 \to [\, y' \,]_3 [\ ]_1$,
$r_{36} \equiv [\, y' \,]_3 [\ ] \to [\ ]_3 [\, y \,]_0$,
$r_{37} \equiv [\, y \,]_0 [\ ]_x \to [\ ]_0 [\, y \,]_x$.

- R_Δ is the set of inter-component communication rules:

$$\begin{aligned} R_{\text{bi-}\Delta^2} \quad = \quad & \{r'_1 \equiv (\langle 1,0\rangle, \texttt{yes}\, v_l/\texttt{yes}\, v_l, \langle 2,0\rangle), r'_2 \equiv (\langle 1,0\rangle, \texttt{no}/\texttt{no}, \langle 2,0\rangle), \\ & r'_3 \equiv (\langle 1,0\rangle, \lambda/\texttt{no}, \langle 2,0\rangle), r'_4 \equiv (\langle 1,0\rangle, \texttt{no}/\lambda, \langle 2,0\rangle), \\ & r'_5 \equiv (\langle 1,0\rangle, \texttt{yes}\,/\lambda, 0), r'_6 \equiv (\langle 1,0\rangle, \texttt{no}/\lambda, 0),\ r'_7 \equiv (\langle 2,0\rangle, \texttt{yes}\,/\lambda, 0), \\ & r'_8 \equiv (\langle 2,0\rangle, \texttt{no}/\lambda, 0)\}. \end{aligned}$$

Note that each (uniform) recognizer tissue P system $\Pi_k(\langle n, m\rangle)$ in a dP scheme will process all Boolean formulas φ, which are in conjunctive normal form (CNF) with n variables and m clauses, where $\langle n, m\rangle = \frac{(n+m)(n+m+1)}{2} + n$, as long as appropriate input multiset $cod(\varphi)$ is supplied to each component system [39]. Furthermore, we will use non-deterministic maximal parallelism in the application of rules of the system. Thus, the correctness of the computations made by the component P systems of 2-$\Delta_{TP_{ec}}$ is done [39].

In the construction of 2-$\Delta_{TP_{ec}}$, labelling rules are introduced. Initially, both cells 1 and 3 contain $a_1, a_2, \ldots, a_n$, which represent the variables in φ. After applying rules $r_{1,i;1}$ and $r_{1,i;3}$, we would have 2^n copies of cell 1 and 2^n copies of cell 3 in both Π_1 and Π_2 in 2-$\Delta_{TP_{ec}}$. Each cell 1 contains a unique truth assignments of the n variables to be evaluated. Each cell 3 contains the corresponding sequence of β_i and β'_i, $1 \leq i \leq n$. Note that the number of t'_i and f'_i equals to the number of β_i and β'_i, respectively.

We need to show that our labelling of all cells 1 is unique to guarantee a consistent truth assignments by both component P systems in 2-$\Delta_{TP_{ec}}$ for each variable in φ before the inter-component communication is done.

The set of labelling rules is composed of $r_{30;1}$ to $r_{33;1}$ of 2-$\Delta_{TP_{ec}}$. The existence of β_i and β'_i in each of cell 3 is assured after applying $r_{1,i;3}$, $1 \leq i \leq n$. Also t'_is and f'_is are in each cell 1 after applying $r_{1,i;1}$.

Labelling rules can be expressed as follows. Given initial labels obtained by using $r_{30;1}$ and $r_{31;1}$. Let $r_{32;1}$ be the mapping $g_{t'} : l \mapsto 2l$, and $r_{33;1}$ be the mapping $g_{f'} : l \mapsto 2l - 1$. These mapping are bijections. Thus the unique labelling v_l of each cell 1 is obtained. Furthermore, each cell 1 labelled v_l contains distinct thruth assignments that makes true the formula φ.

The labelling procedure is done in $O(n)$ steps. Each component P system of $\Delta_{TP_{ec}}$ performs its evaluation individually in $3n + 3m_k + 2$ steps, where $m_k = \lceil \frac{m}{2} \rceil$. In particular, after $3n + 3m_k$ steps, E_{m_k+1} and v_l are found in each cell 1, which means the truth assignment for φ is satisfied. Rule r_{27} collects all pairs v_l and yes in cell 3 at step $3n+3m_k+1$, then r_{28} releases pairs of v_l and yes to the local environment of each component P system in the dP scheme. The communication rule r'_1 can be applied at the same time unused pairs of v_l and yes during the communication will be dumped to cell x.

Finally, the object yes will be at the common environment after step $3n + 3m_k + 4$ or the object no will be at the common environment after step $3n+3m_k+3$. Note that at δ_h the object yes is in 0. □

The succeeding results will measure the amount of communications in each component P system.

Theorem 15. *There exists a bi-directional P protocol $\Delta_{TP_{ec}}$ for solving the SAT problem under a balanced fixed-partition such that $ComN(\Delta_{TP_{ec}}) = 1$, $ComR(\Delta_{TP_{ec}}) = S$, $ComW(\Delta_{TP_{ec}}) = 4S$, where S is the number of satisfying truth assignment to the SAT problem.*

Proof: The dP scheme 2-$\Delta_{TP_{ec}}$ from Theorem 14 decides φ using bi-directional P protocol. After the component P systems of $\Delta_{TP_{ec}}$ individually decide on their parts of the input, they would need to communicate their decisions to the other components for consistency of truth assignments. Since 2-$\Delta_{TP_{ec}}$ is using an antiport-like inter-component communication rules, this requires only one communication. Each of the component P systems, if φ is satisfiable, then the system will certainly have some v_l. In particular, if $(\langle 1,0\rangle, v_l\texttt{yes}/v_l\texttt{yes}, \langle 2,0\rangle)$ is used by both component P systems Π_1 and Π_2, which implies both of them obtained at least a satisfying truth assignment for φ.

Let T_k be the set of satisfying truth assignments obtained by Π_k, $(k = 1, 2)$, then $T_1 \cap T_2$ is the set of satisfying truth assignments for φ. Let $|T_1 \cap T_2| = S$, then $ComN(\Delta_{TP_{ec}}) = 1$, $ComR(\Delta_{TP_{ec}}) = S$, $ComW(\Delta_{TP_{ec}}) = 4S$. □

Theorem 16. *Let φ be any instance of SAT in CNF with m clauses and n variables. Then under a balanced fixed-partition, there is a two-way 2-P protocol $\Delta_{TP_{ec}}$ for solving the SAT problem such that $ComN(\Delta_{TP_{ec}}) = 2$, $ComR(\Delta_{TP_{ec}}) = S + T$, $ComW(\Delta_{TP_{ec}}) = 2(S + T)$, where S is the number of satisfying truth assignment to φ_1, and T is the number of satisfying truth assignments of the SAT problem.*

Proof: The same 2-$\Delta_{TP_{ec}}$ in Theorem 15 is used but rules in R_Δ will be as follows.

$$\begin{aligned} R_{2\text{-}\Delta^2} \quad = \quad & \{r'_1 \equiv (\langle 1,0\rangle, v_l\,\texttt{yes}\,/\lambda, \langle 2,0\rangle), r'_2 \equiv (\langle 1,0\rangle, \lambda/\,v'_l\,\texttt{yes}\,\langle 2,0\rangle), \\ & r'_3 \equiv (\langle 1,0\rangle, \texttt{no}/\lambda, 0), r'_4 \equiv (\langle 2,0\rangle, y'\,\texttt{yes}\,/\lambda, 0), r'_5 \equiv (\langle 2,0\rangle, \texttt{no}/\lambda, 0)\}. \end{aligned}$$

Furthermore, we add cell 4 to each component P system of Δ_{TPec}. Then we have

$$\Delta_{TPec} = (\Gamma, 0, \Pi_1, \Pi_2, R_\Delta),$$

where $\Pi_k(\langle n, m\rangle) = (\Gamma_k, \Sigma_k, \mathcal{E}_k, \mathcal{M}_{k,1}, \mathcal{M}_{k,2}, \mathcal{M}_{k,3}, \mathcal{M}_{k,4}, \mathcal{M}_{k,x}, R_k, i_{k_{in}} = 2, i_{k_{out}} = 0)$, such that $\mathcal{M}_{k,4} = \{a_i \mid 1 \le i \le n\}$, and $\Gamma_k = \Gamma' \cup V \cup V' \cup \{\epsilon_i, \epsilon'_i, \kappa_i, \kappa'_i\,\gamma_i, \gamma'_i\}$.

Similarly, we add the following rules in R_k :

Rule applied to cell 3 with objects $\beta_i, \beta'_i, \kappa_i$ and κ'_i
$r_{1,i;3} \equiv [\,a_1\,]_3 \to [\,\beta_1\,\kappa_1\,]_3\,[\,\beta'_1\,\kappa'_1\,]_3, \quad 1 \le i \le n.$

Rule applied to generate copies of cell 4 with objects γ'_i and ϵ'_i
$r_{1,i;4} \equiv [\,a_i\,]_4 \to [\,\gamma'_i\,]_4\,[\,\epsilon'_i\,]_4, \quad 1 \le i \le n.$

Cell-labeling rules for cell 4:
$r_{30;4} \equiv [\,\kappa_1\,]_3\,[\,\gamma'_1\,]_4 \to [\,v_1\,]_3\,[\,v_1\,\gamma_1\,]_4,$
$r_{31;4} \equiv [\,\kappa'_1\,]_3\,[\,\epsilon'_1\,]_4 \to [\,v_2\,]_3\,[\,v_2\,\epsilon_1\,]_4,$
$r_{32;4} \equiv [\,\kappa_i\,]_3\,[\,\gamma'_i v_l\,]_4 \to [\,v_{2l}\,]_3\,[\,\gamma_i\,v_{2l}\,]_4, \quad l \in \{1, \ldots, 2^n\},\ 2 \le i \le n,$
$r_{33;4} \equiv [\,\kappa'_i\,]_3\,[\,\epsilon'_i v_l\,]_4 \to [\,v_{2l-1}\,]_3\,[\,\epsilon_i\,v_{2l-1}\,]_4, \quad l \in \{1, \ldots, 2^n\},\ 2 \le i \le n.$

Additional rules
$r_{38} \equiv [\, v_l^2\, y\,]_0\, [\, v_l\,]_4 \rightarrow [\, v_l\,]_0\, [\, v_l'\, y'\,]_4$,
$r_{39} \equiv [\,]_0\, [\, v_l'\, y'\,]_4 \rightarrow [\, v_l'\, y'\,]_0\, [\,]_4$.

Let $\varphi = \varphi_1 \wedge \varphi_2$, where φ_1 is assigned to Π_1 and φ_2 is assigned to Π_2. The inputs are placed in the appropriate cells of the component P systems of $\Delta_{TP_{ec}}$ in an encoded form. Solution must be made known to both component P systems of $\Delta_{TP_{ec}}$. Thus, the decision has been made known to both component P systems if object `yes` appeared in 0 or the common shared environment of Π_1 and Π_2. Note that communications start from left to right, then from right to left.

When both components of $\Delta_{TP_{ec}}$ have already been produced all labels v_l in cell 1, and if object E_{m_k+1} appears at least in a cell 1 (it means φ is satisfiable). E_{m_k+1} together with v_l will be evolved to `yes`, v_l in $\langle k, 0\rangle, k = 1, 2$. At this time, communication for the system commences.

Π_1 will communicate to Π_2, all labels v_l of cell 1 that appear in its local environment use the rule $(\langle 1, 0\rangle, v_l\, \texttt{yes}\, /\lambda, \langle 2, 0\rangle)$. Suppose there are S copies of v_l in $\langle 1, 0\rangle$, hence $2S$ copies of objects have been communicated to Π_2 in one communication using S symport-like inter-component rules. Let T_1 and T_2 denote sets of satisfying assignments obtained by Π_1 and Π_2, respectively, then $T_1 \cap T_2$ is the set of satisfying assignments for the SAT problem.

After performing rules r_{38} and r_{39}, Π_2 will send all v_l' and `yes` to Π_1 to inform the solution on the satisfiability of the SAT problem. At the same time, Π_2 sends objects `yes`, y' to 0. Finally, Π_1 is informed with the satisfying assignments for the SAT problem.

Therefore, $ComN(\Delta_{TP_{ec}}) = 2$, $ComR(\Delta_{TP_{ec}}) = S + T$, $ComW(\Delta_{TP_{ec}}) = 2(S + T)$. □

Theorem 17. *Let φ be any instance of the SAT problem in CNF with m clauses and n variables. Then under a balanced fixed-partition, there is a one-way 2-P protocol $\Delta_{TP_{ec}}$ for the SAT problem such that $ComN(\Delta_{TP_{ec}}) = 1$, $ComR(\Delta_{TP_{ec}}) = S$, $ComW(\Delta_{TP_{ec}}) = 2S$, where S is the number of satisfying truth assignment of the SAT problem.*

Proof: The same 2-$\Delta_{TP_{ec}}$ in Theorem 16 is used.

Communications between Π_1 and Π_2 end after Π_1 sends its pairs v_l, `yes` to Π_2. After using r_{39}, Π_2 sends copies of `yes` to 0 to declare satisfiability of the SAT problem.

At the end of computation/communications, both Π_1 and Π_2 know the labels of the satisfying truth assignments for the SAT problem, which requires only a single communication using S number of communication rules with a total of $2S$ objects, where S is the number of satisfying truth assignment of the SAT problem.

Therefore, $ComN(\Delta_{TP_{ec}}) = 1$, $ComR(\Delta_{TP_{ec}}) = S$, $ComW(\Delta_{TP_{ec}}) = 2S$. □

Remark 18. Suppose we consider an unbalanced fixed-partition for the input of the SAT problem. Let $|P_1| = m_1$, and $|P_2| = m - m_1$ such that $|m_1 - m_2| \geq 2$. Then 2-P protocol $\Delta_{TP_{ec}}$ would need $3n + 3m' + 3$ steps to provide a decision for the SAT problem, where $m' = \max\{m_1, m_2\}$. Eventually, results on communication complexity (Theorems 15, 16 and 17) can be re-stated for the case of unbalanced fixed-partition.

5 Solving SAT by 3-Δ_{TPec}

In this section, solution to the SAT problem will be presented using 3 components recognizer tissue P systems.

Theorem 19. *Let φ be any instance of the SAT problem in CNF with m clauses and n variables. Then under a balanced fixed-partition, there is a one-way 3-P protocol 3-$\Delta_{TP_{ec}}$ for SAT such that $ComN(\Delta_{TP_{ec}}) = 2$, $ComR(\Delta_{TP_{ec}}) = V'' + V'$, $ComW(\Delta_{TP_{ec}}) = 2(V'' + V')$, where V'' is the number of satisfying truth assignments to φ_2, and V' is the number of satisfying truth assignments of Π_1 for φ_1.*

Proof: Let 3-$\Delta_{TPec} = (\Gamma, 0, \Pi_1, \Pi_2, \Pi_3, R_\Delta)$ be a dP scheme, where each Π_k $(k = 1, 2, 3)$ is the same as those in Theorem 16. Each Π_k has almost the same set of rules presented in Theorem 16 in processing input instance φ of SAT. In this model, the following rule is added:

$$r_{40} \equiv [\, v_l' \, v_l \, y \,]_0 \, [\, v_l \,]_4 \to [\, v_l \,]_0 \, [\, v_l' \, y' \,]_4.$$

Consequently, the following inter-component communication rules for 3-Δ_{TPec} will be used.

$$\begin{aligned} R_{1\text{-}\Delta^3} \quad = \quad & \{r_1' \equiv (\langle 1, 0\rangle, v_l \, \texttt{yes} \, / \lambda, \langle 2, 0\rangle), r_2' \equiv (\langle 2, 0\rangle, v_l', \texttt{yes} \, / \, \lambda \, \langle 3, 0\rangle), \\ & r_3' \equiv (\langle 3, 0\rangle, y' \, \texttt{yes} / \lambda, 0), r_4' \equiv (\langle 3, 0\rangle, \texttt{no} \, / \lambda, 0), \\ & r_5' \equiv (\langle 2, 0\rangle, \texttt{no} / \lambda, 0), r_6' \equiv (\langle 1, 0\rangle, \texttt{no} / \lambda, 0)\}. \end{aligned}$$

Communication between components of 3-Δ_{TPec} is done successively from Π_1 to Π_2, then from Π_2 to Π_3. After each component processed their part of the input, Π_1 starts communication with Π_2 by sending all labels of cell 1. Π_2 obtained all these v_l, which are labels of cell 1 that provide satisfying truth assignments for φ from Π_1. Let T_1 be the set of all labels v_l of cell 1, if $|T_1| = V'$, then Π_1 sent $2V'$ copies of object to Π_2 in one step.

Now Π_3 obtained from Π_2 copies of object v_l' and `yes` after $3n + 3m_k + 6$ steps. Each v_l' is a label of a satisfying truth assignment made by Π_1 and Π_2, hence all copies of objects $v_l', v_l, v_{l'}$, `yes` are contained in $\langle 3, 0\rangle$ after Π_2 sent objects v_l', `yes` to Π_3. Π_3 uses r_{40} and r_{39} to prepare using $(\langle 3, 0\rangle, y' \, \texttt{yes} / \lambda, 0)$ to declare satisfiability of φ. The number of y' is equal to the number of satisfying truth assignments of φ. In particular, the number of y' is equal to $|T_1 \cap T_2 \cap T_3|$, where T_1, T_2, and T_3 are sets of satisfying truth assignments evaluated by Π_1, Π_2, and Π_3, respectively.

Let $|T_1| = V'$, and $|T_2| = V''$. Finally, we have $ComN(3\text{-}\Delta_{TPec}) = 2$, $ComR(3\text{-}\Delta_{TPec}) = V'' + V'$, and $ComW(3\text{-}\Delta_{TPec}) = 2V'' + 2V'$. □

Theorem 20. *Let φ be any instance of the SAT problem in CNF with m clauses and n variables. Then under a balanced fixed-partition, there is a two-way 3-P protocol 3-$\Delta_{TP_{ec}}$ for the SAT problem such that $ComN(3\text{-}\Delta_{TP_{ec}}) = 4$, $ComR(3\text{-}\Delta_{TP_{ec}}) = 2S + V' + V''$, $ComW(\Delta_{3\text{-}TP_{ec}}) = 2(S + V' + V'')$, where S is the number of satisfying truth assignment to φ, V' is the number of satisfying assignments of Π_1 for φ_1, and $V'' = |T_1 \cap T_2|$, where T_1 and T_2 are satisfying truth assignments of φ_1 and φ_2, respectively.*

Proof: The 3-Δ_{TPec} in Theorem 19 is used but the set of inter-component communication rules $R_\Delta = R_{2\text{-}\Delta^3}$ uses only the symport-like communication rules. Specifically,

$$R_{2\text{-}\Delta^3} = R_{1\text{-}\Delta^3} \cup \{(\langle 2, 0\rangle, \lambda \, / \, v_l', \langle 3, 0\rangle), (\langle 1, 0\rangle, \lambda \, / \, v_l', \langle 2, 0\rangle)\}.$$

From Π_1, it is easy to know that $2V'$ copies of object are sent to Π_2, where $V' = |T_1|$, T_1 being the set of satisfying truth assignments for φ_1 evaluated by Π_1.

Using r_{40} and r_{39}, Π_2 will eventually send objects v_l' and `yes` to Π_3. The total amount of objects is equal to $2V''$ in a single communication. Π_3 realizes S labels that give satisfying truth assignments for φ, where S is the total number of labels that are common to all component P systems.

After $3n + 3m_k + 10$ steps, y' and `yes` will be sent by Π_3 to 0, simultaneously, objects v_l' and `yes` are sent to Π_2. Hence Π_2 sends the same copies of objects to Π_1. The communication going back from Π_3 to Π_1 requires $2S$ copies of objects using $2S$ rules in two communications.

Therefore, $ComN(3\text{-}\Delta_{TPec}) = 4$, $ComR(3\text{-}\Delta_{TPec}) = S + V' + V''$, and $ComW(3\text{-}\Delta_{TPec}) = S + 2V' + 2V''$. □

Theorem 21. *Let φ be any instance of the SAT problem in CNF with m clauses and n variables. Then under a balanced fixed-partition, there is a bi-directional 3-P protocol 3-$\Delta_{TP_{ec}}$ for the SAT problem such that $ComN(3\text{-}\Delta_{TP_{ec}}) = 1$, $ComR(3\text{-}\Delta_{TP_{ec}}) = S$, $ComW(\Delta_{3\text{-}TP_{ec}}) = 6S$, where S is the number of satisfying truth assignment to φ.*

Proof: The 3-P protocol 3-Δ_{TPec} used in this proof will have component P systems similar to that in Theorem 15, but we use the following additional rules:

$r_{41} \equiv [\, v_l \, \mathtt{yes} \,]_3 \, [\, v_l \,]_4 \to [\,]_3 \, [\, v_l'^3 \, \mathtt{yes}^3 \,]_4$, and $r_{42} \equiv [\,]_0 \, [\, v_l' \, \mathtt{yes} \,]_4 \to [\, v_l \, \mathtt{yes} \,]_0 \, [\,]_4$,

and the set of inter-component communication rules R_Δ as follows:

$$\begin{aligned} R_{\text{bi-}\Delta^3} \quad = \quad & \{r_1' \equiv (\langle 1,0\rangle, \mathtt{yes}\, v_l/\mathtt{yes}\, v_l, \langle 2,0\rangle), r_2' \equiv (\langle 1,0\rangle, \mathtt{yes}\, v_l/\mathtt{yes}\, v_l, \langle 3,0\rangle), \\ & r_3' \equiv (\langle 2,0\rangle, \mathtt{yes}\, v_l/\mathtt{yes}\, v_l, \langle 3,0\rangle), \\ & r_4' \equiv (\langle 1,0\rangle, y\, \mathtt{yes}\, /\lambda, 0), r_5' \equiv (\langle 2,0\rangle, y\, \mathtt{yes}\, /\lambda, 0), r_6' \equiv (\langle 3,0\rangle, y\, \mathtt{yes}\, /\lambda, 0), \\ & r_7' \equiv (\langle 1,0\rangle, \mathtt{no}/\lambda, 0), r_8' \equiv (\langle 2,0\rangle, \mathtt{no}/\lambda, 0), r_9' \equiv (\langle 3,0\rangle, \mathtt{no}/\lambda, 0)\}. \end{aligned}$$

Furthermore, each component of 3-Δ_{TPec} is as follows:

$$\Pi_k(\langle n,m\rangle) = (\Gamma_k, \Sigma_k, \mathcal{E}_k, \mathcal{M}_{k,1}, \mathcal{M}_{k,2}, \mathcal{M}_{k,3}, \mathcal{M}_{k,4}, \mathcal{M}_{k,x}, R_k, i_{k_{in}} = 2, i_{k_{out}} = 0),$$

such that $\mathcal{M}_{k,4} = \{a_i \mid 1 \le i \le n\}$ and $\Gamma_k = \Gamma' \cup V \cup V' \cup \{\epsilon_i, \epsilon_i', \kappa_i, \kappa_i' \gamma_i, \gamma_i'\}, r_{34} \notin R_k$.

In this modified 3-Δ_{TPec}, the additional rules allow each component P system to triple its v_l and `yes` in order to prepare for a simultaneous antiport-like communications. If φ is satisfiable, then rules r_1', r_2', and r_3' could be used. Simultaneously, by all component P systems send y, `yes` to 0. Since the communication is bi-directional, this is done in one step, using S rules and total of 6 objects. Note that S is the number of satisfying assignments for φ.

Therefore, $ComN(3\text{-}\Delta_{TP_{ec}}) = 1$, $ComR(3\text{-}\Delta_{TP_{ec}}) = S$, $ComW(\Delta_{3\text{-}TP_{ec}}) = 6S$, where S is the number of satisfying truth assignment to φ. □

6 Relative performance of k-Δ_{TPec}

The relative performance and parallelizability of k-Δ_{TPec} is considered in this section. The concept of *weak parallelizability* introduced in [21] is also considered.

A problem L is said to be (k,m)-*weakly ComX parallelizable,* for some $k \ge 2, m \ge 1$ and $X \in \{, N, R, W\}$, if there is a dP scheme Δ with k components and there is a finite $F_\Delta \subseteq L$ such that each string $x \in L - F_\Delta$ can be written as $x = x_1 x_2 \cdots x_k$, such that $||x_i| - |x_j|| \le 1$ for all $1 \le i, j \le k$, each component Π_i takes as input the string $x_i, 1 \ge i \le k$ and string x is accepted by Δ in a halting computation δ such that $ComX \le m$. A problem L is called *weakly ComX parallelizable* if it is (k,m)-weakly $ComX$ parallelizable for some $k \ge 2, m \ge 1$.

In the case of k-Δ_{TPec}, $k = 2, 3$ deciding on the SAT problem, the following results on parallelizability are obtained. In particular, results presented in Section 3 implies the following.

Theorem 22. *Let SAT*$= \{\, \varphi \mid \varphi$ *has n variables and m clauses*$\}$.

1. *Let* 2-Δ_{TPec} *be a uniform bi-directional* 2*-P protocol for SAT under balanced fixed-partition, then SAT is* $(2,r)$*-weakly ComX parallelizable, where* $(r, ComX) \in \{\, (1, ComN), (S, ComR), (4S, ComW) \,\}$, *S is the number of satisfying truth assignments for* φ.

2. *Let* 2-Δ_{TPec} *be a uniform two-way* 2*-P protocol for SAT under balanced fixed-partition, then SAT is* $(2,r)$*-weakly ComX parallelizable, where* $(r, ComX) \in \{\, (2, ComN), (S+T, ComR), (2(S+T), ComW) \,\}$, *where S is the number of satisfying truth assignment to* φ_1, *and T is the number of satisfying truth assignments for* φ.

3. *Let* 2-Δ_{TPec} *be a uniform one-way* 2*-P protocol for SAT under balanced fixed-partition, then SAT is* $(2,r)$*-weakly ComX parallelizable, where* $(r, ComX) \in \{\, (1, ComN), (S, ComR), (2S, ComW) \,\}$, *where S is the number of satisfying truth assignment of* φ_1.

In the case of 3-Δ_{TPec} for solving the SAT problem under balanced fixed-partition, the results in Section 5 implies the following.

Theorem 23. *Let SAT= $\{\varphi \mid \varphi$ has n variables and m clauses$\}$.*

1. *Let -Δ_{TPec} be a uniform bi-directional 3-P protocol for SAT under balanced fixed-partition, then SAT is $(3, r)$-weakly ComX parallelizable, where $(r, ComX) \in \{(1, ComN), (S, ComR), (6S, ComW)\}$, where S is the number of satisfying truth assignments for φ.*

2. *Let 2-Δ_{TPec} be a uniform two-way 3-P protocol for SAT under balanced fixed-partition, then SAT is $(3, r)$-weakly ComX parallelizable, where $(r, ComX) \in \{(4, ComN), (2S+V'+V'', ComR), (2(S+V'+V''), ComW)\}$, where S is the number of satisfying truth assignment to φ, V' is the number of satisfying assignments of Π_1 for φ_1, and $V'' = |T_1 \cap T_2|$, where T_1,and T_2 are satisfying truth assignments of φ_1, and φ_2, respectively.*

3. *Let 2-Δ_{TPec} be a uniform one-way 3-P protocol for SAT under balanced fixed-partition, then SAT is $(3, r)$-weakly ComX parallelizable, where $(r, ComX) \in \{(2, ComN), (V''+V', ComR), (2(V''+V'), ComW)\}$, where V'' is the number of satisfying truth assignment to φ_2, and V' is the number of satisfying assignments of Π_1 for φ_1.*

The relative efficiency of performance of k-Δ_{TPec} $(k = 2, 3)$ can also be viewed with respect to its computation time spent solving a problem. In this respect, k-Δ_{TPec} will be compared to the efficient solution presented in [39]. Let $TIME_{\Pi(\langle n,m\rangle)}(n, m)$ be the running time of $\Pi(\langle n, m\rangle)$, and $TIME_{\Delta_{TPec}}(n, m)$ denotes the running time of Δ_{TPec}.

In [39], $TIME_{\Pi(\langle n,m\rangle)}(n, m) = 2n + 3m + 2$, while Theorem 14 gives $TIME_{2\text{-}\Delta_{TPec}}(n, m) = 3n+3\frac{m}{k}+4$. The following limit represents the speed up ratio between $\Pi(\langle n, m\rangle)$ and 2-$\Delta_{TPec}(n, m)$.

$$\lim_{n\to\infty} \frac{TIME_{\Pi(\langle n,m\rangle)}(n, m)}{TIME_{2\text{-}\Delta_{TPec}}(n, m)} = \lim_{n\to\infty} \frac{2n + 3m + 2}{3n + 3\frac{m}{k} + 4}.$$

The value of this limit is required to be at least 2, to imply improvements of the computation by 2-$\Delta_{TPec}(n, m)$ compared with that by $\Pi(\langle n, m\rangle)(n, m)$ solving the same problem.

Let $k = 2$, and $m = n$, the speed-up ratio is:

$$\lim_{n\to\infty} \frac{TIME_{\Pi(\langle n,m\rangle)}(n, m)}{TIME_{2\text{-}\Delta_{TPec}}(n, m)} = \lim_{n\to\infty} \frac{5n + 2}{4.5n + 4} \approx 1.11.$$

This suggests that for any $k \geq 2$, as long as $m \leq n$, k-Δ_{TPec} could not do significant advantage compared with $\Pi(\langle n, m\rangle)$ for solving SAT. It can also be observed that if for any k, $n = m^2$, we would have

$$\lim_{n\to\infty} \frac{TIME_{\Pi(\langle n,m\rangle)}(n, m)}{TIME_{\Delta_{TPec}}(n, m)} = \lim_{m\to\infty} \frac{2m^2 + 3m + 2}{3m^2 + 3\frac{m}{k} + 4} < \frac{2}{3}.$$

This would mean no parallelism.

If we let $m = n^2$, then speed-up ratio becomes

$$\lim_{n\to\infty} \frac{TIME_{\Pi(\langle n,m\rangle)}(n, m)}{TIME_{\Delta_{TPec}}(n, m)} = \lim_{n\to\infty} \frac{2n + 3n^2 + 2}{3n + \frac{3}{k}n^2 + 4} = k \geq 2.$$

This shows that k-$\Delta_{TP_{ec}}$ computes in at least half the required number of steps by $\Pi(\langle n, m\rangle)$, if $m \geq n^2$, for any k.

The uniform recognizer tissue P systems in [39] may not be the optimal uniform recognizer tissue P systems for solving the SAT problem, that is, deciding SAT with the smallest possible steps, but it is efficient enough to compare the relative performance of k-$\Delta_{TP_{ec}}$ for solving SAT.

7 Conclusions and discussions

In this work, a distributed P scheme that solves instances φ of SAT is presented. The power of the recognizer tissue P systems with evolutional communication rules and cell division from [39] is capitalized in a dP scheme. Labelling of all cells 1 after cell division is suggested to give precise and exact decision on the satisfiability of φ. Moreover, $\Delta_{TP_{ec}}$ requires that whatever is the decision for φ, all component P systems know the decision. Two possible types of communication that $\Delta_{TP_{ec}}$ could be performed, namely, *antiport-like inter-component communication* and *symport-like inter-component communication.* Thus, the concept of a P protocol is introduced. Taking into account the types of inter-component communications on dP scheme, one-way P protocol, two-way P protocol and bi-directional P protocol are defined. The concept of a uniform P protocol is also mentioned. The idea of balanced and unbalanced partitions are also presented and, in particular, a so-called (un)balanced fixed-partition is considered in distributing parts of the input to component P systems of dP scheme.

It is shown that under a balanced fixed-partition k-$\Delta_{TP_{ec}}$, could be able to decide on the satisfiability of any instance φ of SAT using only one communication under a bi-directional k-P protocol. The number of inter-component rules is the number of satisfying truth assignments for φ. But the number of objects sent by the k component P systems increases with respect to k. In the case $k = 2, 3$, we obtained $ComN(2\text{-}\Delta_{TP_{ec}}) = 1$, $ComR(2\text{-}\Delta_{TP_{ec}}) = S$, $ComW(2\text{-}\Delta_{TP_{ec}}) = 4S$, and $ComN(3\text{-}\Delta_{TP_{ec}}) = 1$, $ComR(3\text{-}\Delta_{TP_{ec}}) = S$, $ComW(\Delta_{3\text{-}TP_{ec}}) = 6S$, where S is the number of satisfying truth assignment to φ.

Notice that k-$\Delta_{TP_{ec}}$ is a uniform dP scheme, that is, each component P system Π_k has (almost) the same set of rules being implemented during every computation. It is also assume that each cell in tissue P systems is connected to every other cells in the system and can communicate directly with each other. The only trade-off is extra steps for each component to reproduce the objects to be communicated using bi-directional mode of communication. This is polynomial with respect to the number of component P systems. This implies that under k-$\Delta_{TP_{ec}}$, SAT is $(k, 1)$-*weakly ComN parallelizable,* for all k. Note that this invariance with respect to *weakly ComN* is obtained under a balanced fixed-partition of input, fixed encoding and with a bi-directional k-P protocol.

The same invariance could be observed in the case of (k, S)- *weakly ComR parallelizability* of k-$\Delta_{TP_{ec}}$ under a balanced fixed k-partition using bi-directional communication mode, for any k. Notice that the k components P systems in k-$\Delta_{TP_{ec}}$ will have to produce k copies of the labels of cell with satisfying truth assignments of their respective part of the input. Eventually, k-$\Delta_{TP_{ec}}$ uses the antiport-like inter component communication rule that matches these labels together with `yes`. Finally every cell in k-$\Delta_{TP_{ec}}$ sends object `yes` to 0 to signal the end of the computation and decided the satisfiability of φ.

Note that in Remark 18, it was stated that Theorem 15 and Theorem 16 could be re-stated in the case of unbalanced partition. Then at least for $k = 2, 3$, SAT is $(k, 1)$-*weakly ComN parallelizable* and (k, S)-*weakly ComR parallelizable* under an unbalanced fixed-partition. It is believe that SAT is $(k, 1)$-*weakly ComN parallelizable* and is also (k, S)-*weakly ComR parallelizable* for any k under an unbalanced k fixed-partition.

Statement 1 of both Theorem 22 and Theorem 23 shows that SAT belongs to the class of problems that could be solved by uniform k-$\Delta_{TP_{ec}}$, $k = 2, 3$ with $ComW(2\text{-}\Delta_{TP_{ec}}) = 4S$, and $ComW(3\text{-}\Delta_{TP_{ec}}) = 6S$, where S is the number of satisfying truth assignment to φ, which implies that the objects needed to be communicated by the system increases with the number components. Note that a uniform 3-$\Delta_{TP_{ec}}$ for SAT needs more $2S$ objects to decide the satisfiability of φ compared with 2-$\Delta_{TP_{ec}}$. Using the uniform k-$\Delta_{TP_{ec}}$ in this paper, it might be reasonable to believe that SAT may be (k, s)-*weakly ComW parallelizable*, but it is not $(k + 1, s)$-*weakly ComR parallelizable,* for any k and for some s.

In the case of one-way and two-way uniform k-P protocols under balanced fixed-partition ($k = 2, 3$), it was demonstrated that the total amount of objects to be communicated and the total number of inter-component rules are increased with respect to the number of component P systems of k-$\Delta_{TP_{ec}}$. These results suggest that $ComX$, $X \in \{N, R, W\}$ is directly proportional to k. In particular, SAT belongs to the class of problems that is $(2, r)$-*weakly ComX parallelizable*, which do not belong to the class of problems that are $(3, r)$-*weakly ComX parallelizable,* where $(r, ComX) \in \{(r, ComN), (r, ComR), (r, ComW)\}$. It is of interest to know if these observed relations between 2-Δ and 3-Δ could be extended to k-Δ and $(k + 1)$-Δ with one-way and two-way uniform k P protocols under balanced fixed-partition.

It is also realized that the amount of clauses related to the number of variables is quite necessary in order to obtain efficiency in using k-$\Delta_{TP_{ec}}$ to solve SAT. In particular, if $m \leq n$, the relative efficiency of k-$\Delta_{TP_{ec}}$ cannot be equal to 2. This is regardless if we increase the number k of component P systems. But at $m = n^2$, we obtain a reasonable relative efficiency k, for any $k \in O(n)$. Notice here that this efficiency is an upper bound of the precise efficiency we wanted to obtain, since k-$\Delta_{TP_{ec}}$ is compared only to a particular $\Pi(\langle n, m\rangle)$ for solving SAT.

In [21], a problem L is said to be (k, r, s)-*efficiently ComX parallelizable,* for some $k \geq 2, r \geq 1, s \geq 2$, and $X \in \{N, R, W\}$, if it is (k, r)-*weakly ComX parallelizable,* and there is a dP scheme Δ such that

$$\lim_{n\to\infty} \frac{TIME_{\Pi}(x)}{TIME_{\Delta}(x)} \geq s,$$

for all P systems Π such that $L = L(\Pi)$. Moreover, $TIME_{\Pi}(x)$ is the smallest number of steps need for Π to accept string x should be estimated with respect to all Π for L, while $TIME_{\Delta}(x)$ is just given by means of a construction of a suitable dP scheme Δ. It might be reasonable to believe that SAT is (k, r, s) *efficiently ComX parallelizable*, where $(rComX) \in \{(r_1, ComN), (r_2, ComR), (r_3, ComW)\}$, for some real numbers r_i, $i = 1, 2, 3$; $s \leq k$ is the speed up ratio and k is the number of components in the uniform dP scheme under bi-directional, one-way and two-way uniform k-P protocol.

Notice that in order to minimize the amount of objects to be communicated proper, encoding of objects is necessary. We need not to communicate the whole multiset of objects, but an encoded version of them. This encoding add-up to the time and number of cells to be used by component P systems in the systems. In the case of this paper, cell labelling is proposed to encode the truth assignment uniquely to maintain consistency of assigning truth values to variable being evaluated by the whole systems. In order to keep the use of rules efficiently, we have to expect to produce at most exponential amount of cells. Finally, we suggest that one of possible path to take in this line of research is to minimize the amount of objects to be communicated by component P systems in solving problems, keeping the performance of component P systems within reasonable efficiency.

Uniform P protocols under balanced fixed-partition are the ones considered, and remarked on the unbalanced fixed-partition for solving SAT is provided. It would be nice to consider what may be called *optimal-partition*, where we design partition of the objects of the input and see how it fared with fixed-partition with respect to communication measures. Non-uniform k-P

protocol solving hard problems might also be a nice direction to pursue. By non-uniform, means allowing each component P system to perform what it thinks necessary with respect to the input part. Furthermore, it is of interest to consider communication resources with respect to some communication P protocols or dP schemes for solving other hard problems.

Acknowledgements

The work was supported by National Natural Science Foundation of China (61320106005, 61602192, and 61772214), China Postdoctoral Science Foundation (2016M600592 and 2017T100554), and the Innovating Scientists and Technicians Troop Construction Projects of Henan Province (154200510012). H. Adorna was also supported by Semirara Mining Corporation, Inc. Professorial Chair of the College of Engineering, U.P. Diliman, DOST-Engineering Research and Development for Technology Program grant, and an OVCRD RLC grant 2017-2018.

Bibliography

[1] Adorna, H., Păun, Gh. Pérez-Jiménez, M.J. (2010): On Communication Complexity in Evolution-Communication P Systems, *Rom. J. Inf. Sci. Tech.*, 13(2), 113–130, 2010.

[2] Alhazov, A. (2004): On Determinism of Evolution-Communication P Systems, *J. Univers. Comput. Sci.*. 10(5), 502–508, 2004.

[3] Buño, K., Adorna, H., Pan, L. Song, B.(2017): Communicaton Complexity of Distributed Tissue-Like P Systems for Solving SAT Problem, *Proc. of the 18th International Conference on Membrane Computing*, Bradford, UK, 373–383, 2017.

[4] Buño, K. Cabarle, F., Adorna, H., Calabia, M.(2018): Solving N-Queens Problem using dP Systems with Active Membranes, *Theor. Comput. Sci.*, 2018.

[5] Cavaliere, M. (2003): Evolution-Communication P Systems, In: *Păun, Gh. et al, (Eds.) LNCS*, Springer, Heidelberg, 2597, 134–145, 2003.

[6] Csuhaj-Varjú, E., Margenstern, M., Vazil, G., Verlan, S. (2007): On Small Universal Antiport P System, *Theor. Comput. Sci.*, 372(2-3), 152–164, 2007.

[7] Csuhaj-Varjú, E., Vaszil, G. (2012): Finite dP Automata Versus Multi-head Finite Automata. In: *Gheorghe, M. et al. (Eds.): LNCS*, Springer, Heidelberg, 7184, 120–138, 2012.

[8] 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.

[9] Dzitac, I. (2015): Impact of Membrane Computing and P Systems in ISI WoS. Celebrating the 65th Birthday of Gheorghe Păun, *Int. J. Comput. Commun*, 10(5), 617–626, 2015.

[10] Elias, S., Gokul, V, Krithivasan, K., Gheorghe, M., Zhang, G. (2012): A Variant of Distributed P Systems for Real Time Cross Layer Optimization, *J. Univers. Comput. Sci.*, 18(13), 1760–1781, 2012.

[11] Frisco, P., Govan, G., Leporati, A. (2012): Asynchronous P Systems with Active Membranes, *Theor. Comput. Sci.*, 429, 74–86, 2012.

[12] Gazdag, Z. (2014): Solving SAT by P Systems with Active Membranes in Linear Time in the Number of Variables. In: *Alhazov, A. et al. (Eds.) LNCS*, Springer, Heidelberg, 8340, 189–205, 2014.

[13] Gutiérrez-Naranjo, M.A., Martínez-del-Amor, M.A., Pérez-Jurtado, I., Pérez-Jiménez, M.J. (2009): Solving N-Queens Puzzle with P Systems. In: *Proc. of the 7th Brainstorming Week on Membrane Computing*, Sevilla, Spain, 199–210, 2009.

[14] Hernandez, N., Juayong, R., Adorna, H. (2014): On the Communication Complexity of Some Hard Problems in ECPe Systems. In: *Alhazov, A. et al. (Eds.): LNCS*, Springer, Heidelberg, 8340, 206–224, 2014.

[15] Macías-Ramos, L.F., Valencia-Cabrera, L., Song, B., Pan, L., Pérez-Jiménez, M.J. (2015): A P-Lingua Based Simulator for P Systems with Symport/Antiport Rules, *Fund. Informa.*, 139(2), 211–277, 2015.

[16] Martínez-del-Amor, M.A., García-Quismondo, M., Macías-Ramos, L.F., Valencia-Cabrera, L., Riscos-Núñez, A., Pérez-Jiménez, M.J. (2015): Simulating P Systems on GPU Devices: A Survey, *Fund. Informa.*, 136(3), 269–284, 2015.

[17] Pan, L., Păun, Gh. (2009): Spiking Neural P Systems with Anti-Spikes, *Int. J. Comput. Commun.*, 4(3), 273–282, 2009.

[18] Pan, L., Pérez-Jiménez, M.J. (2010): Computational Complexity of Tissue-Like P Systems, *J. Complexity*, 26(3), 296–315, 2010.

[19] Păun, Gh. (2000): Computing with Membranes, *J. Comput. Syst. Sci.*, 61(1), 108–143, 2000.

[20] Păun, Gh. (2016): Membrane Computing and Economics: A General View, *Int. J. Comput. Commun.*, 11(1), 105–112, 2016.

[21] Păun, Gh., Pérez-Jiménez, M.J. (2010): Solving Problem in a Distributed Way in Membrane Computing: dP Sytems, *Int. J. Comput. Commun.*, 5, 238–259, 2010.

[22] Păun, Gh., Pérez-Jiménez, M.J. (2012): An Infinite Hierarchy of Languages Defined by dP Systems, *Theor. Comput. Sci.*, 431, 4–12, 2012.

[23] Păun, Gh. Rozenberg, G., Salomaa, A. (Eds.)(2010): *The Oxford Handbook of Membrane Computing*, Oxford University Press, New York, 2010.

[24] Peng, H., Wang, J., Pérez-Jiménez, M.J., Riscos-Núñez, A. (2015): An Unsupervised Learning Algorithm for Membrane Computing, *Inf. Sci.*, 304, 80–91, 2015.

[25] Song, T., Pan, L. (2016): Spiking Neural P Systems with Request Rules, *Neurocomputing*, 193, 193–200, 2016.

[26] Song, T., Pan, L., Păun, Gh. (2013): Asynchronous Spiking Neural P Systems with Local Synchronization, *Inf. Sci.*. 219, 197–207, 2013.

[27] Song, B., Pan, L. (2016): The Computational Power of Tissue-Like P Systems with Promoters, *Theor. Comput. Sci.*, 641, 43–52, 2016.

[28] Song, B., Song, T., Pan, L. (2017): A Time-Free Uniform Solution to Subset Sum Problem by Tissue P Systems with Cell Division, *Math. Struct. Comp. Sci.*, 27(1), 17–32, 2017.

[29] Song, B., Zhang, C., Pan, L. (2017): Tissue-Like P Systems with Evolutional Symport/Antiport Rules, *Inf. Sci.*, 378, 177–193, 2017.

[30] Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del-Amor, M.A., Riscos-Núñez, A., Pérez-Jiménez, M.J. (2017): Cooperation in Transport of Chemical Substances: A Complexity Approach within Membrane Computing, *Fund. Informa.*, 154(1-4), 373–385, 2017.

[31] Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del-Amor, M.A., Riscos-Núñez, A., Pérez-Jiménez, M.J. (2017): Reaching Efficiency through Collaboration in Membrane Systems: Dissolution, Polarization and Cooperation, *Theor. Comput. Sci.*, 701, 226–234, 2017.

[32] Valencia-Cabrera, L., Wu, T., Zhang, Z.,Pan, L., Pérez-Jiménez, M.J. (2016): A Simulation Software Tool for Cell-Like Spiking Neural P Systems, *Rom. J. Inf. Sci. Tech.*, 20(1), 71–84, 2016.

[33] Wu, T., Zhang, Z., Păun, Gh., Pan, L. (2016): Cell-Like Spiking Neural P Systems, *Theor. Comput. Sci.*, 623, 180–189, 2016.

[34] Yao, A.C. (1979): Some Complexity Questions Related to Distributed Computing, In: *Proceedings of the eleventh annual ACM symposium on Theory of computing*, 209–213, 1979.

[35] Zandron, C., Leporati, A., Ferretti, C., Mauri, G., Pérez-Jiménez, M.J. (2008): On the Computational Efficiency of Polarizationless Recognizer P Systems with Strong Division and Dissolution, *Fund. Informa.*, 87(1), 79–91, 2008.

[36] Zhang, G., Rong, H., Neri, F., Pérez-Jiménez, M.J. (2014): An Optimization Spiking Neural P System for Approximately Solving Combinatorial Optimization Problems, *Int. J. Neural Syst.*, 24, 1–16, 2014.

[37] Zhang, X., Pan, L., Păun, A. (2015): On the Universality of Axon P Systems, *IEEE T. Neur. Net. Lea.*, 26(11), 2816-2829, 2015.

Appendix: Some Tables for Inter-component Communications

The following tables below show how communication between component P systems transfer. The table of communications starts when the systems already obtained a satisfying truth assignments of their respective input parts. Notice that Table 6 which is continued in Table 7 starts at step 4. This initially started when E_{m_k+1} appeared in any of the cell 1, which will be the step 1 of the table.

Below are tables for cases where there are at most three component recognizer tissue P systems with evolutional communication rules and cell division. one-way, two-way and bi-directional P protocol are considered below. Notice that the set of communication rules differs per kind of P protocol model. The set of rules used by each component P systems are mostly based on the results in [39]. Variations on rules R_k for each k depends on the P communication mode required of the systems.

The tables provide labels of each column and each row provides information on specific action transferred with respect to the preceding row.

One would notice that the set of inter-component rules of other P communication mode are similar as stated in the proof of some of the Theorems.

Table 1: Expanded Communication Configuration for k-Δ. Rejecting Communication (1).

step	rule	cell 1	cell 2	cell 3	cell x	$\langle 1,0\rangle$	0	$\langle 2,0\rangle$	cell x	cell 3	cell 2	cell 1
1	r_{26}	v_l	α_p, d	v_l						v_l	α_p	v_l
2	r_{29}	y	α_{p+1}, d	v_l,						v_l, yes	α_{p+1}, d	y
3	r'_6					no		v_l, yes		y'	α_{p+2}, d	
4							no	v_l, yes		y'	α_{p+3}, d	

Table 2: Expanded Communication Configuration for k-Δ. Rejecting Communication (2).

step	rule	cell 1	cell 2	cell 3	cell x	$\langle 1,0\rangle$	0	$\langle 2,0\rangle$	cell x	cell 3	cell 2	cell 1
1	r_{26}	v_l	α_p, d	v_l						v_l	α_p	v_l
2	r_{29}	y	α_{p+1}, d	v_l,						v_l	α_{p+1}, d	y
3	r'_2	y				no		no				y
4		y					no					y

Table 3: Expanded Communication Configuration for bi-Δ^2. Accepting Communication.

step	rule	cell 1	cell 2	cell 3	cell x	$\langle 1,0\rangle$	0	$\langle 2,0\rangle$	cell x	cell 3	cell 2	cell 1
1	r_{26}, r_{27}	E_{m_k+1}, v_l	α_p, d	v_l						v_l	α_p	E_{m_k+1} v_l
2	r_{28}, r_{35}	y	α_{p+1}, d	v_l, yes						v_l, yes	α_{p+1}, d	y
3	r'_1, r_{34}, r_{36}		α_{p+2}, d	y'		v_l, yes		v_l, yes		y'	α_{p+2}, d	
4	r'_7, r'_5, r_{37}		α_{p+3}, d		yes $v_l \notin T_1$	v_l yes		v_l yes	yes $v_l \notin T_2$		α_{p+3}, d	
5					y, v_l yes		yes		y, v_l yes			

Table 4: Expanded Communication Configuration for $2\Delta^2$ and 1-Δ^2. Accepting Communication. Note that $v_{l'}$ found in cell x are those which are elements of $T_1 \cap T_2$.

step	rule	cell 1	cell 2	cell 3	cell 4	cell x	$\langle 1,0\rangle$	0	$\langle 2,0\rangle$	cell x	cell 4	cell 3	cell 2	cell 1
1	r_{26} r_{27}	E_{m_k+1} v_l	α_p d	v_l	ϵ_i γ_i v_l						ϵ_i γ_i v_l	v_l	α_p d	E_{m_k+1} v_l
2	r_{26} r_{28} r_{35}	y	α_{p+1} d	v_l yes	ϵ_i γ_i v_l						ϵ_i γ_i v_l	v_l yes	α_{p+1} d	y
3	r_{26} r_{36} r'_1		α_{p+2} d	y'	ϵ_i γ_i v_l		v_l yes		$v_l$yes		ϵ_i γ_i v_l	y'	α_{p+2} d	
4	$r_{26}, r_{34},$ r_{38}, r_{37}		α_{p+3}, d		$v_l, \epsilon_i, \gamma_i$		y		v_l^2 yes^2 y $v_{l'} \notin T_1$ yes y		$v_l, \epsilon_i, \gamma_i$ $v_l, \epsilon_i, \gamma_i$		α_{p+3}, d α_{p+3}, d	
5	r_{26}, r_{39}		α_{p+4}, d		$v_l, \epsilon_i, \gamma_i$	y			v_l yes^2	yes y $v_{l'}$	v'_l y' ϵ_i γ_i $v_{l''}$, $l'' \neq l$		α_{p+6}, d	
6	r_{26}, r'_2		α_{p+7}, d		$v_l, \epsilon_i, \gamma_i$	y			v_l v'_l y' yes^2	yes y $v_{l'}$	v'_l y' ϵ_i γ_i $v_{l''}$, $l'' \neq l$		α_{p+7}, d	
7			α_{p+8}, d		$v_l, \epsilon_i, \gamma_i$	y	v'_l yes	y' yes	v_l	yes y $v_{l'}$	v'_l y' ϵ_i γ_i $v_{l''}$, $l'' \neq l$		α_{p+7}, d	

Table 5: Expanded Communication Configuration for $1\Delta^3$. Accepting Communication.

step	rule	cell 1	cell 2	cell 3	cell 4	cell x	$\langle 2,0\rangle$	0	$\langle 3,0\rangle$	cell x	cell 4	cell 3	cell 2	cell 1
4	r_{26} r_{37} r_{38}		α_{p+3} d		ϵ_i γ_i v_l		$v_l^2\ y$ $\texttt{yes}^2$ $v_{l'}$, yes		v_l yes y		ϵ_i γ_i v_l		α_{p+3} d	
5	r_{26} r_{39}		α_{p+4} d		ϵ_i γ_i v_l' $v_{l,}$ y		v_l $\texttt{yes}^2$ $v_{l'}$ yes		v_l yes	y	ϵ_i γ_i v_l		α_{p+4} d	
6	r_{26} r'		α_{p+5} d		ϵ_i γ_i $v_{l'}$		v_l' y' $\texttt{yes}^2$		v_l yes	y	ϵ_i γ_i v_l		α_{p+5} d	
7	r_{26} r_{40}		α_{p+6} d		ϵ_i γ_i $v_{l'}$		y' yes		v_l' yes v_l yes $v_{l'}$ yes		ϵ_i γ_l v_l		α_{p+6} d	
8	r_{26} r_{39}		α_{p+7} d		ϵ_i γ_i $v_{l'}$		y' yes		v_l yes $v_{l'}$ yes		$\epsilon_i\ v_l'$ $\gamma_i\ y'$ v_l		α_{p+7} d	
9	r_{26} r'		α_{p+8} d		ϵ_i γ_i v_l		y' yes		$v_l'\ y'$ v_l yes $v_{l'}$ yes	y	ϵ_i γ_i v_l		α_{p+8} d	
10			α_{p+8} d		ϵ_i γ_i v_l			y' yes	v_l' v_l $v_{l'}$ yes	y	ϵ_i γ_i v_l		α_{p+8} d	

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 323-336, June 2018.

Multi-Objective Binary PSO with Kernel P System on GPU

N. Elkhani, R. C. Muniyandi, G. Zhang

Naeimeh Elkhani*, Ravie Chandren Muniyandi
Centre for Cyber Security
Faculty of Information Sciene and Technology
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia
*Corresponding author: naeimeh.elkhani@siswa.ukm.edu.my
ravie@ukm.edu.my

Gexiang Zhang
School of Electrical Engineering
Southwest Jiaotong University
Chengdu 610031
Sichuan, P.R. China
zhgxdylan@126.com

Abstract: Computational cost is a big challenge for almost all intelligent algorithms which are run on CPU. In this regard, our proposed kernel P system multi-objective binary particle swarm optimization feature selection and classification method should perform with an efficient time that we aimed to settle via using potentials of membrane computing in parallel processing and nondeterminism. Moreover, GPUs perform better with latency-tolerant, highly parallel and independent tasks. In this study, to meet all the potentials of a membrane-inspired model particularly parallelism and to improve the time cost, feature selection method implemented on GPU. The time cost of the proposed method on CPU, GPU and Multicore indicates a significant improvement via implementing method on GPU.
Keywords: parallel membrane computing, GPU based membrane computing, kernel P system, parallel multi-objective binary PSO, parallel kernel P system-multi objective binary PSO.

1 Introduction

In the literature [1], from one point of view, methods of feature selection for classification can be divided into three families 1) methods for flat features (filter models, wrapper models, embedded models), 2) methods for structured features (graph structure) and 3) methods for streaming features. A main disadvantage of the filter approach despite its lower time consumption is the fact that it does not interact with the classifier, usually leading to worse performance results than those obtained with wrappers. However, the wrapper model comes with an expensive computational cost. An intermediate solution for researchers can be the use of embedded methods that are usually a mix of two or more feature selection methods from different origins which use the core of the classifier to establish a criterion to rank features.

From another perspective for the division of feature selection and classification methods, wide range of mixed methods are developed mainly based on evolutionary learning methods such as genetic algorithm (GA), neighborhood search like K nearest neighbor (KNN) and swarm intelligence algorithms such as particle swarm optimization (PSO). Based on our review on GA and KNN, the problems of various proposed methods can be categorized to three parts. First; e.g., pure genetic algorithm generally has limitations such as 1) slow convergence, 2) lacks of rank based fitness function and 3) being a time-consuming approach. Mixed methods of GA and

KNN were not capable to tackle with these problems completely. Second; in terms of classification accuracy, resulted accuracy in intelligent feature selection and classification algorithms varies greatly either in different datasets, also due to building unstable method overfitting risk increases dramatically when they examine on high density datasets. PSO because of first; its ability to match with graph model as genes (nodes) and define relationship between them (edge), second; higher accuracy in compared with flat (filter and wrapper) methods third; reasonable time complexity on CPU is our candidate in proposing a membrane-inspired feature selection method.Computational cost is a big challenge for almost all intelligent algorithms which are run on CPU. Recently new attempts have been started to develop parallel feature selection and classification methods such as [24] and some efforts are focused on parallelization of intelligent optimization algorithms, such as parallel genetic algorithm on CPUs/computers to identify informative genes for classification [16, 23], parallel Genetic algorithm on GPU [6, 15, 22], parallel PSO on GPU [14, 19–21, 28] and parallel processing of microarray data [13]. In this regard, our proposed membrane-inspired feature selection method should perform with an efficient time that we aimed to settle via using potentials of membrane computing in parallel processing Due to the inherent large-scale parallelism feature of membrane computing, any membrane computing inspired model can fully represent this computation model only in the case of using the parallel platform. From the beginning of introducing this model, it was a big concern in all membrane related studies. For instance, to fully implement parallelism of such membrane computing model and to support an efficient execution [25] used a platform based on reconfigurable hardware. Without parallelism, all subsequent studies face a challenge of how to make rules available in all steps of computation. In [2] a sequential computing of membrane computing, they just had an option of using one membrane and made the rules periodically available based on time-varying sequential P system. A sequential kernel P system multi objective binary particle swarm optimization feature selection and classification method proposed in the study of [8,9]. Even by using minimal parallelism of using rule; at least a rule from a set of rules in a membrane, e.g., with the active membrane; solving NP-complete problems in polynomial time through trading space for time leads to make a more efficient model of membrane computing. The architectural differences between CPUs and GPUs cause CPUs to perform better on latency-sensitive, partially sequential, single sets of tasks. In contrast, GPUs performs better with latency-tolerant, highly parallel and independent tasks. Recently, several studies attempted to utilize membrane computing to improve intelligent algorithms. For instance, multi-core processing used in the study of [17] utilized a membrane computing inspired genetic algorithm and [18] have highlighted parallelism in membrane computing in the case of solving the N-queen's problem.

As a variant of P system, kernel P system (KP system) introduced for the first time in the study of [11,12]. This variant of P system integrates most of the features of membrane computing which have been successfully used for modelling problems and are applied in various application. Generally, there is two types of the rules in KP systems: first type of rules deals with the objects to transfer them between compartments or send the objects from compartment to environment and vice versa; second type of rules deal with the membrane structure to change the topology of the compartments. Multi objective optimization refers to the problem of finding a set of values which meet the limitations and are capable of optimizing the set of values to another set of values which are the objective of optimization. PSO method itself is divided to two different approaches called single objective and multi-objective. These two approaches meet different requirements. Single objective is appropriate for the problems have only one correct solution. In versus, most of the hard problems are often confronted with multi-objective decision problem that their goal is to find the "best" solution which corresponds to the minimum or maximum value of a single objective that lumps all different objectives into one. The combination of membrane computing with optimization algorithm has been used in many studies such as [26, 27].

In this study, all the rules of KP system as rewriting and communication rule, division, input/output, link creation have used to develop the proposed KP-MObPSO model. A multi-core kernel P system multi objective binary particle swarm optimization feature selection and classification method proposed in the study of [8]. The most important attempts to parallelize membrane computing models are being done via using of graphic processing units (GPUs) [3–5, 7, 10]. All of these efforts have demonstrated that a parallel architecture is better positioned in performance than traditional CPUs to simulate P systems, due to the inherently parallel nature of them, and specifically GPUs obtain very good preliminary results simulating P systems.

2 Criteria to execute kernel P system multi objective binary PSO on GPU

The important factor in implementing a P system-based model on GPU is to attention the rate of communication between the threads, which is related to the dependencies between objects [17, 18]. Previous approaches of implementing P system on GPU did not consider this factor that exert effect on the performance of executing model on GPU. According to Figure 1, every single thread using the local memory and a thread block uses the share memory and a grid of thread blocks use global memory. The main strategy will be assigning dependent objects to the same thread for execution. Dependent objects in the proposed model are those objects that should be produced by prior rule and enter the compartment as input object to trigger the next rule. This is the reason rules are following priority for execution means those rules have higher priority should be execute first to generate the objects which are necessary to trigger the execution of other rules. Thus, in our model those objects that their existence is dependent to the existence of other objects in the compartment will execute on the same thread along with their parent objects. To design KP-MObPSO-SVM model on GPU, two important points are concerned, first, the dependency between the objects and rules to decrease the rate of communication, second; access to the lesser cost memories in the execution of threads like local and shared memory. As it is shown in the Figure 1, a single thread is used to assign objects and rules which are dependent to each other and they can use the local memory to keep the value for the objects and send the value to another rule to trigger its execution. When the execution of dependent rules is done, and completed in single threads, it will be needed to share the result of the threads and make a decision to choose the best result to continue the execution of model. To do this, a thread block which belongs to the current single threads can exchange the result via shared memory.

According to [12], A KP system of degree n is a tuple,

$$k_{\Pi} = (O, \mu, C_1, \ldots, C_n, i_0)$$

where O is a finite set of objects, called an alphabet; μ defines the membrane structure, which is a graph, (V, E), where V represents vertices indicating compartments and belongs to a set of labels $L(l_i, \ldots)$, and E represents edges; $C_i = (t_i, w_i), 1 \leq i \leq n$, is a compartment of the system consisting of a compartment type from T and an initial multiset, w_i, over O; i_0 is the output compartment, where the result is obtained (this will not be used in this study). Each rule r may have a guard g, in which case r is applicable when g is evaluated to true. Its generic form is $r\{g\}$. KP systems use a graph-like structure (similar to that of tissue P systems) and two types of rules

1. Rules to process objects: these rules used to transform object or to move objects inside compartments or between compartments. These rules are called rewriting, communication and input-output rules:

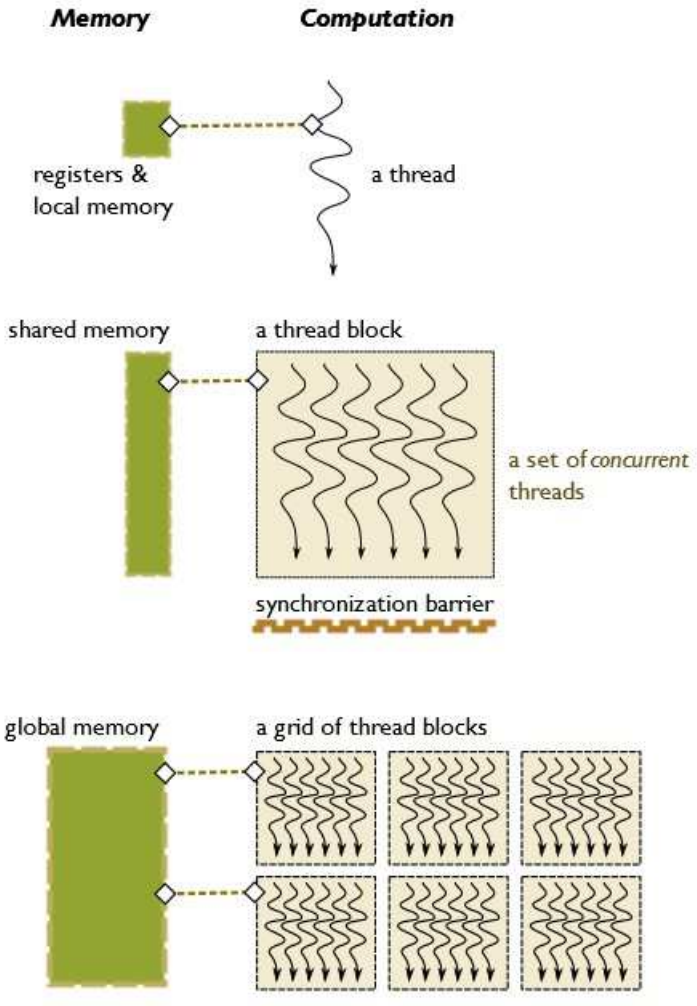


Figure 1: GPU Memory

a Rewriting and communication rule: $x \to y\{g\}$, where $x \in A^+$, $y \in A^?$, $g \in$ Finite regular Expressions FE over $(A \bigcup \bar{A})$; y at the right side defines as $y = (a_1, t_1), \ldots, (a_h, t_h)$, where $a_j \in A$ and $t_j \in L$, $1 \leq j \leq h$, a_j is an object and t_j is a target, respectively.

b The input-output rule: $(x/y)\{g\}$, where $x, y \in A^?$, $g \in$ Finite regular Expressions FE over $(A \bigcup \bar{A})$; means that x can be sent from current compartment to the environment or y can be brought from environment to the target compartment.

2. System structure rules: these rules make a fundamental change in the topology of the membranes for example with division rule on a compartment, dissolution rule on a specific compartment, make a link between compartments or dissolve the link between them. These rules are described as follow:

a Division rule: $[]_{l_i} \to_{l_i l \ldots l_i h} \{g\}$, where $g \in$ Finite regular Expressions FE over $(A \bigcup \bar{A})$; means compartment l_i can be replaced with h number of compartments. All newly created compartments inherit objects and links of l_i;

b Dissolution rule: $[]_{l_i} \to \lambda\{g\}$; means compartment l_i is not exist anymore as well as all its links with other compartments.

c Link-creation rule: $[]_{l_i}; []_{l_j} \rightarrow []_{l_i} - -[]_{l_j}\{cg\}$; means a link will be created between compartment l_i with compartment l_j. If there is more than one compartment with the label l_j, one of them will have a link with l_j non-deterministically.

d Link-destruction rule: $[]_{l_i} - -[]_{l_j} \rightarrow []_{l_i}; []_{l_j}\{cg\}$; means the existence link between l_i and l_j will eliminate and there will not be any link between them anymore. The same as link creation, if there are more than one compartment which have a link with l_i then one of them will be selected non-deterministically to apply this rule.

3 Proposed model

The entire proposed model includes two main parts, first part, features selection based on KP-MObPSO plus classification based on (kernel P system-support vector machine) KP-SVM and second part, KP-embedded feature selection/SVM classification. The first part defines modelling and implementing previous MObPSO based on KP system rules with some improvements which leads to the result consists of different set of marker genes, so called KP-MObPSO feature selection. Thereafter, an error rate calculator based on KP-SVM applied to measure the error rate of marker gene sets. To design the first part of the model, first it is needed to design improved version of KP-MObPSO on GPU. To do so, we assume there are 4 compartments each including 6 particles as Figure 2 and Figure 3. The same process will repeat for each particle as follow:

Step 1: to assign one thread for each gene of dataset (which is including of 100 genes) first we need to allocate memory in the Host for dataset of genes. Each gene keeps the values of six samples, therefore an array of threads of at least size 6 is needed. Moreover, some other threads are defined to save the values will be generated in the processing steps on the GPU and will return back to the Host. Step 2: after allocating Host memory to each gene in the dataset, genes are needed to transfer to Device memory to execute on GPU. Step 3: by assigning random number of genes inside each particle, the main process will start by executing KP-MObPSO-SVM. The kernel defined as "addkernel" will add the genes inside the particles and will execute the rules on each thread. The sets of rules called "subgraph" and "Mycost" carry the dependent rules as R1 to R10 (Table 1). Thus, these rules will execute on each single threads of genes. Local memory will keep the value of all variables which are defined as objects and all the rules have access to the same local memory to pull and push the value of objects. After that, the value of object "Fit" for each thread and the initial value for the object "pBestScore" need to save in shared memory to execute the function called "compare" according to the rules R11 and R12 to refresh the value of object "pBestScore" with the minimum value of objects "Fit". This function will implement on all threads from thread 1 to thread 4. Then, the velocity function according to the R13 to R16 will be executed on each thread with utilizing of local memory Figure **??**. Also, two sets of rules including replacing rules and decision rules will be executed on threads to initiate the cycle of particle preparation inside the compartments and restart the first part of model again till predefined iteration (it=100). Then after getting the marker genes collected by each thread, KP-SVM rules will apply to calculate the error rate for each set of marker genes. These rules are reminded in (Table 2). Step 4: at the end of first part of the model, the error rates of each set of marker genes will be calculated. According to the Table 2, (R1) is flag which is an object with default value zero. R2, flag value will rewrite to 1 if it meets the guard values including an error rate between 0 and 0.3 as well as having at least two cancerous genes indexes in marker genes. R3, q and e are the counters of normal genes and cancerous genes respectively which rewrite to a default value zero, and marker genes 2 keep a backup of marker genes resulted from first part of the model.

Table 1: KP-MObPSO Rules

R1:Rewriting $[[p, max\_c]position]_0 \rightarrow [[(position_1 \dots position_{\mathrm{n}})_1 \dots (position_1 \dots position_{\mathrm{n}})_{\mathrm{p}}]_0[[]_1[]_{\mathrm{position}}]_0$
R2:Communication $[[(position_1 \dots position_{\mathrm{n}})_1 \dots (position_1 \dots position_{\mathrm{n}})_{\mathrm{p}}]position]_0 \rightarrow$ $[(position_1 \dots position_{\mathrm{n}})_1 \dots (position_1 \dots position_{\mathrm{n}})_{\mathrm{p}}]_1]_0$
R3:Communication $[(position_1 \dots position_{\mathrm{n}}) \dots (position_1 \dots position_{\mathrm{n}})_{\mathrm{p}}]_1 \rightarrow [(position_1 \dots position_{\mathrm{n}})]p_1 \dots [(position_1 \dots position_{\mathrm{n}})_{\mathrm{p}}]]_1$ $Rulesinsideeachp : []_{\mathrm{p1}} \dots []_{\mathrm{pn}} : r4 > r5 > r6 > r7$
R4:Rewriting $[(position, a, max_c, p)] \rightarrow [NGENES], [NGENES] \rightarrow [NewNGENES, Q, c]$
R5:(Communication/Rewriting) $[(NewNGENES, c, p, a)] \rightarrow [C], [C] \rightarrow [sumdiss], [a] \rightarrow [snr]$ $[snr] \rightarrow [sumsnr], [sumdiss, sumsnr] \rightarrow [FIT]$
R6:Link creation $[]position - - - - []master$
R7:(Communication/Rewriting) $[FIT] \rightarrow [pBestScoren]master, [Q^n] \rightarrow [Q^n]master$
R8:Division $[[P_{\mathrm{n}}][pBestScore_{\mathrm{n}}, Q_{\mathrm{n}}]_{\mathrm{master}}]_1]_0 \rightarrow$ $[P]_1 \dots [P]_{\mathrm{n}}[pBestScore_{\mathrm{n}}, Q_{\mathrm{n}}, gBestScore]_{\mathrm{master}}]_1[[]_{\mathrm{p1}} \dots [P]_{\mathrm{n}}$ $[fitness, pBest, gBest, Velocity, c1, c2, w, Vmax, s]_{\mathrm{master}}]_1]_0$
R9: Membrane Dissolution $[[]P_1 \dots []P_n[]_{\mathrm{master}}]_1]\theta \rightarrow \lambda$
R10: Link Creation $[[]P_1 \dots []P_n[pBestScoren]_{\mathrm{master}}]_1]_0, [[]P_1 \dots []P_n[fitnessn]_{\mathrm{master}}]_1]_0 \rightarrow$ $[]P_1 \dots []P_n[pBestScoren]_{\mathrm{master}}]_1]_0 - - - - - [[]P_1 \dots []P_n[fitnessn]_{\mathrm{master}}]_1]_0$
R11: Communication/rewriting $[[[pBestScore_n]_{\mathrm{master}}]_1]_0 \rightarrow [[fitness_n]_{\mathrm{master}}]_1]_0, [[[pBest_n]_{\mathrm{master}}]_2]_0 \rightarrow$ $[[[fitness_n]_{\mathrm{master}}]_2]_0 < [[[pBestScore_n]_{\mathrm{master}}]_1]_0, 1 \leq n \leq p$ $[[[gBestScore]_{\mathrm{master}}]_1]_0 \rightarrow [[pBestScore_n]_{\mathrm{master}}]_1]_0, [[[gBest_n]_{\mathrm{master}}]_2]_0 \rightarrow$ $[[[pBestScore_n]_{\mathrm{master}}]_1]_0 < [[[gBestScore]_{\mathrm{master}}]_1]_0, 1 \leq n \leq p$ $[[[converge]_{\mathrm{master}}]_1]_0 \rightarrow in[[[[pBestScore]]_{\mathrm{master}}]_1]_0$
R12:Communication $[[[position]]_1]_0 \rightarrow [[[position]_{\mathrm{master}}]_1]_0$
R13: Communication/rewriting $[[[position_n, c1, c2, c, w, pBest, gBest, p, max_c, rand]_{\mathrm{master}}]_1]_0 \rightarrow [[[Velocity]_{\mathrm{master}}]_2]_0$ $[[[Velocity]_{\mathrm{master}}]_2]_0 \rightarrow [Vmax]_{\mathrm{master}}]_1, Velocity > Vmax$ $[[[Velocity]_{\mathrm{master}}]_1]_0 \rightarrow [-Vmax]_{\mathrm{master}}]_2, Velocity < -Vmax$ $[[[[position]]_{\mathrm{master}}]_1]_0 \rightarrow, rand <= 1/(1 + exp(-2 * Velocity)$ $[[[[position]]_{\mathrm{master}}]_2]_0 \rightarrow 0, rand > 1/(1 + exp(-2 * Velocity)$
R14:Output/link creation/communication & rewrite $[[[[position]]_{\mathrm{master}}]_1]_0 \rightarrow [[position]_1]_0[[]_1]_0, [[]position]_0 \rightarrow []_1]_0 - - - -$ $[[]position]_0[[[position]]_1]_0 \rightarrow [[position]_position]_0$
R15: Division rule $[[]p1 \dots []pn[]master]12[[]p1 \dots []pn[]master]121[[]p1 \dots []pn[]master]122$
r16: Membrane dissolution $[[]p1 \dots []pn[]master]12 \rightarrow \lambda$

Table 2: KP-SVM Rules

R1:Rewriting $(s) \rightarrow find(M(:) == i), table(i) \rightarrow lenght(s), 1 \leq i \leq 100$
R2:Rewriting $markergenes(i, it+1) \rightarrow i, table(i) > it, 1 \leq i \leq 100, 1 \leq it \leq n$
R3:Rewriting $y(i,1) \rightarrow +1, max\_j \rightarrow max\_j + 1, 1 \leq i \leq 50$
R4:Rewriting $y(i,1) \rightarrow -1, max\_f \rightarrow max\_f + 1, 51 \leq i \leq 100$
R5:Rewriting $Badgenes(it+1,1) \rightarrow max\_f$
R6:Rewriting $Y(j,1) \rightarrow +1, 1 \leq j \leq max\_j * 3$ $Y(j,1) \rightarrow -1, max\_j * 3 \leq j \leq max\_j * 3 + max\_f * 3$
R7:Input $wholedata(k,1) \rightarrow a(i,j), 1 \leq i \leq 100, table(i) > it, j = 1, k \rightarrow k+1$ $wholedata(k,1) \rightarrow a(i,j), 1 \leq i \leq 100, table(i) > it, j = 3or5$ $wholedata(k,2) \rightarrow a(i,j) 1 \leq i \leq 100, table(i) > it, j = 2or4$ $wholedata(k,2) \rightarrow a(i,j), k \rightarrow k+1 1 \leq i \leq 100, table(i) > it, j = 6$
R8:Rewriting $wholedata(k,1), wholedata(k,2) \rightarrow XY, Holdout = 0.10 \rightarrow Pcvpartition$ $X(P.training), Y(P.training) \rightarrow SVMStructsvmtrain$ $SVMStruct, X(P.test) \rightarrow Csvmclassify Sum(Y(P.test)C)/P.testsize \quad errRate$ $Y(P.test), C \rightarrow conMat$ $ERR(1,1) \rightarrow errRatefirst \quad 100 \quad times \quad iteration, \quad gBestScore = inf$ $ERR(it+1,1) \rightarrow errRate$ $Constant \rightarrow ERR(it+1,1) not \quad first \quad 100 \quad times \quad iteration, \quad gBestScore \neq qinf$
R9:Rewriting $particle2 \rightarrow Markergenes(:, it+1) 1 \leq it \leq n, 0 \leq ERR(it+1,1) \leq 0.3$
R10:Division $a(i,j) \rightarrow (Particle2, it = 1), (Particle2, it = 2), \ldots, (Particle2, it = n)$

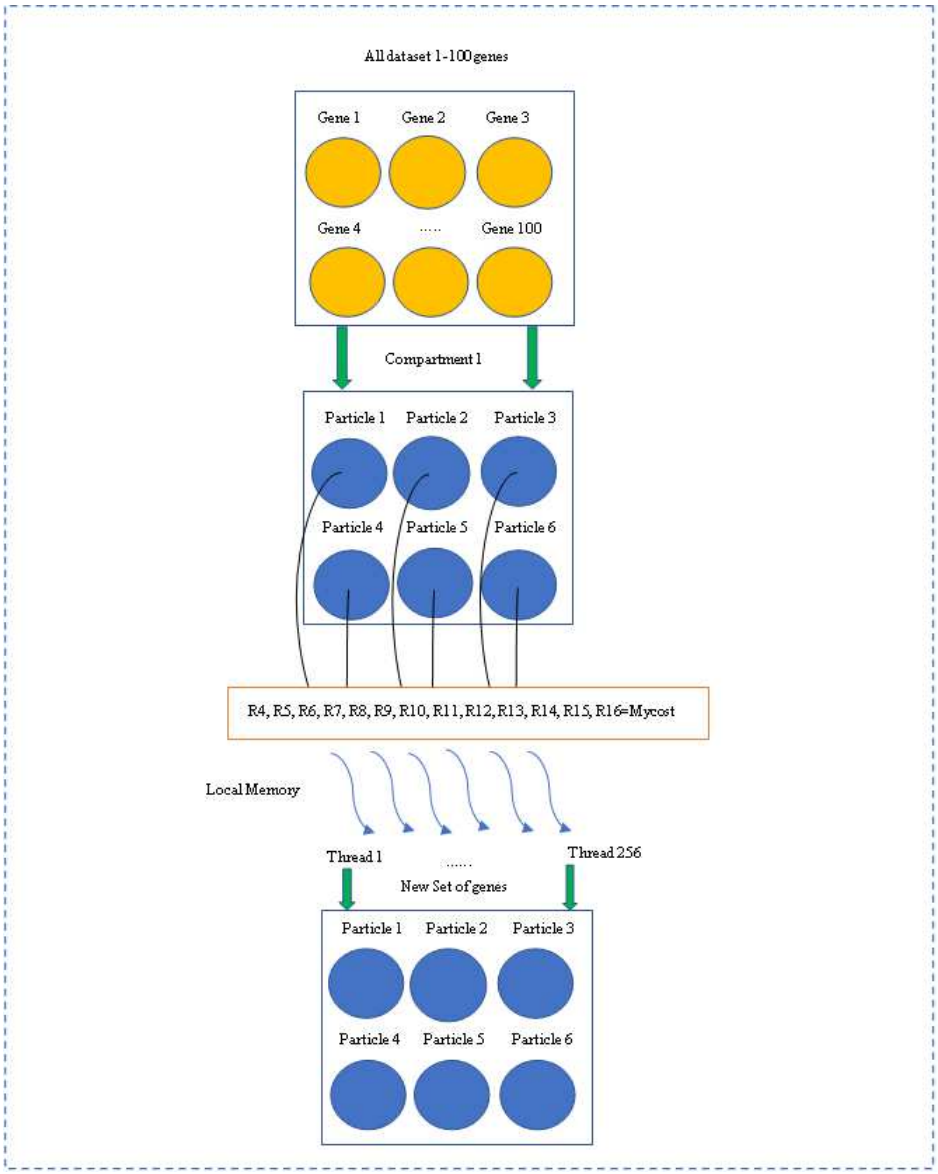


Figure 2: Designing on GPU

Step 5: elapsed time so far for feature selection executed on one particle is 5.914 Sec.

Step 6: Let assume compartment number 1 is chosen because of meeting the criteria of having better error rate. Marker genes 2 object will be used for further procedure on GPU. To implement embed feature selection method, another kernel defined as "second kernel". For each gene number from 1 to 100, rules number r4-r10 will apply to see whether entering a new gene can improve the error rate of that particle or no. R4 and R5, gives a flag to index of genes based on the type of genes whether they are belonged to normal genes or cancerous genes as +1 and -1, respectively.

In parallel, q and e which are the counters of normal genes and cancerous genes will be update. R6, the object max_j and max_f updates the number of normal and cancerous genes. R7, clear the value of q and e. R8 reserve a place for the samples of gene indexes are selected as marker genes and R9, inputs the real value of reserved samples inside a compartment called *wholedata*. *Wholedata* compartment keeps real data samples for gene indexes are already highlighted as

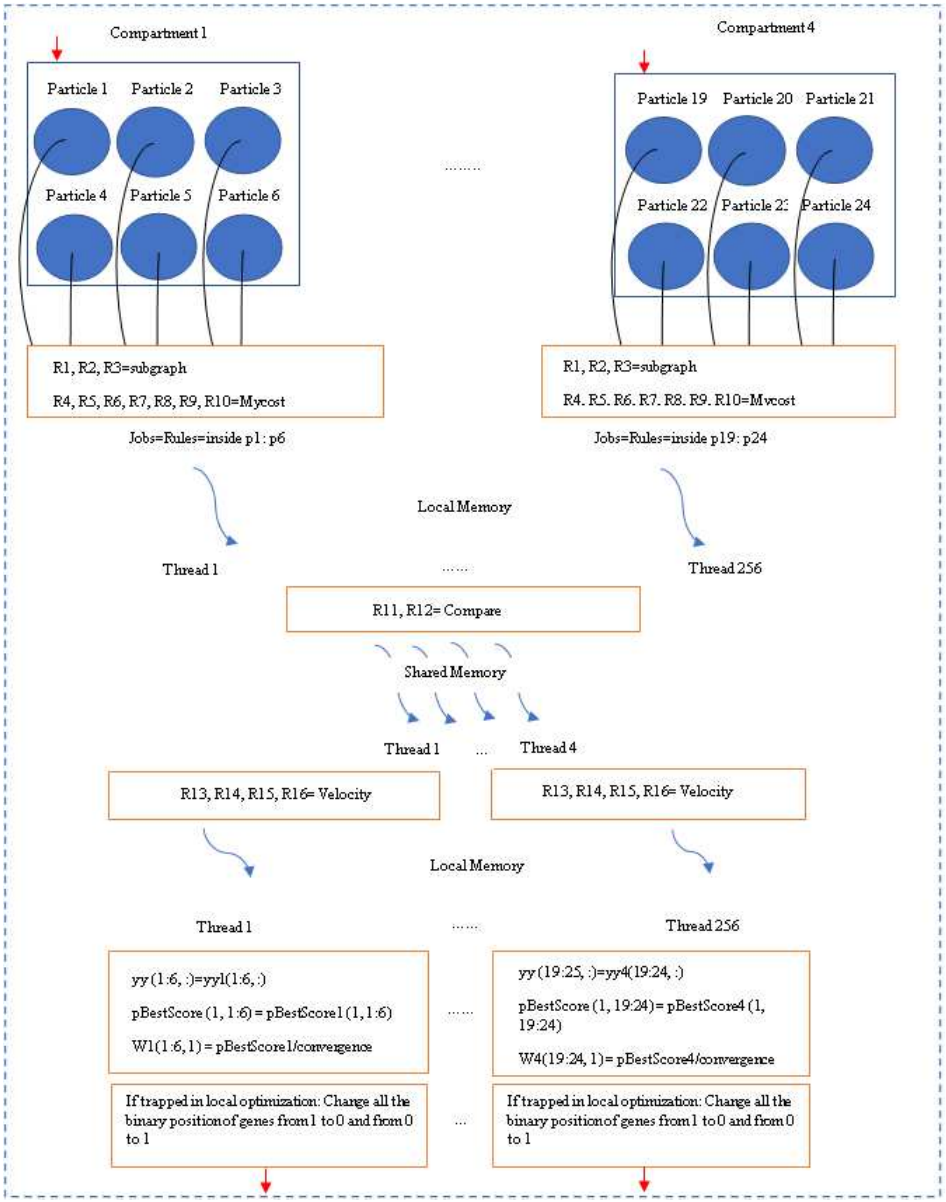


Figure 3: Continue Designing on GPU

marker genes. R10, applies an SVM package with rewriting rules to evaluate error rates. To decide whether adding a new gene can improve the error rate or no, rule number 11 compares error rates after adding each gene (from 1 to 100) with the constant error rate object resulted in the first part of modelling for each set of marker genes. If adding a new can already can improve constant error rate of that set of marker genes, the r11 will add this gene index to the set of marker genes. Otherwise if a gene value already enter to the particle cannot improve the error rate, it should be exit from the particle. Rules, R12-15 apply output and rewriting rules to eliminate gene values inside the particle. After executing the computation model for n number of iterations which each iteration represents one particle from it=1 to it=n, particles of marker genes will update by new genes and result a new set of particles as (particle 3, it=1/ particle 3, it=2/. . . / particle 3, it=n). It means by the end of embedded feature selection and classification; real data set will divide to a new compartment. Every time one gene will be added to the set of genes and error rate will calculate to decide adding the gene to the set is suitable or no. If

Table 3: Benchmark

KP-MObPSO-SVM	
Max_c =100	Maximum number of genes
P=25	Number of particles
$Max_iteration$=100	Maximum number of generating
fitness, gbestscore, pbestscore	Objects to save the results
Hardware CPU	Core(TM)i5-6200U CPU @2.30 GHz (RAM): 12.0 GB
Hardware GPU	NVIDA Geforce 680

adding the gene decrease the error rate means the gene will be kept and if increase the error rate means the gene will be removed of the chosen gene set. New value for "SUM", "mean" value, "SNR" (signal to noise) value, "dissimilarity" value, "fit" (fitness) value, "velocity" value, "std" (standard deviation) value, "gBestvalue" value. Total elapsed time for all processes of 2 particles is 13.12 Sec.

4 Evaluation and result

The objective of this study was to improve the time efficiency in implementing a membrane-based optimization algorithm. In this regard, two models as KP-MObPSO and KP-MObPSO-SVM are implemented on CPU, multicore and GPU to compare their time complexity. The comparison of the time cost between two models indicates GPU based executions can increase time efficiency of the models drastically. The benchmark of evaluation in terms of the maximum number of the genes, number of particles, maximum number of iterations, and hardware specification of CPU and GPU are explained in the Table 3. According ot Table 4 and Table 5 with the same example of 25 particles executed on multi-core and GPU with 4 workers and independent iteration respectively, time cost dropped significantly from 5.5 min to 73.87 sec in KP-MObPSO feature selection and from 15 min to 164 sec in KP-MObPSO feature selection and classification. While execution on CPU was taken 3.68 min in KP-MObPSO feature selection and 8 min in Embed KP-MObPSO-SVM feature selection and classification. Comparing CPU and multicore indicates, due to the frequent interaction between clients and workers, it takes longer time and does not lead to improvement in timely execution of multicore KP-MObPSO. Therefore, as it is shown by example, a GPU based KP-MObPSO and KP-MObPSO-SVM are more time efficient than CPU based models. Regardless of taking example, Table 3 indicates the time complexity of the parallel KP-MObPSO-SVM on GPU according to the big O calculated as $O(NM) + O(N) + O(N/P * M) + O(N/P) + O(M/2P)) + O(M/P) + O(M/(P * Q)) + O(M2/P^2) + O(M^2/2P^2) + O(Q^2/P^2) + O(2Q/P) + O(M * Q^2/P^2)$ where (N=Max number of particles, M=Max number of genes, I=Max number of iteration, Q: Max number of samples, P=Max number of processors). The time complexity of sequential KP-MObPSO-SVM is $O(NM) + O(INM)) + O(IN) + O(M/2) + O(M^2/2) + O(M/2) + O(MQ) + O(M) + O(M^2) + O(Q^2) + O(Q) + O(MQ^2) + O(N)$ where (N=Max number of particles, M=Max number of genes, I=Max number of iteration, Q: Max number of samples). $O(M^2) + O(MQ^2) and O(M^2/P^2) + O(Q^2/P^2)$ are the highest time cost for sequential and parallel KP-MObPSO-SVM respectively. It is shown how the number of processors decrease the time cost of implementing KP-MObPSO-SVM on GPU and leads to more time efficient method.

Table 4: KP-MObPSO

KP-MObPSO-SVM	
CPU	25 particles: 3.68 min (10 times of 100 iteration)
Multi-Core	25 particles: 5.5 min (4 workers)
GPU	2 particles: 5.91 Sec 25 particles: 73.87 Sec (independent iteration)

Table 5: KP-MObPSO-SVM

KP-MObPSO-SVM	
CPU	25 particles: 8 min (10 times of 100 iteration)
Multi-Core	25 particles: 15 min (4 workers)
GPU	2 particles: 13.12 Sec 25 particles: 164 Sec (independent iteration)

5 Conclusion

To design KP-MOBPSO-SVM model on GPU, two important points are concerned, first, the dependency between the objects and rules to decrease the rate of communication, second; access to the lesser cost memories in the execution of threads like local and shared memory. Thus, according to the first criteria, those objects that their existence is dependent to the existence of other objects in the compartment will execute on the same thread along with their parent objects. Based on the second criteria, objects and rules which are dependent to each other will use the local memory to keep the value for the objects and will send the value to another rule to trigger its execution. When the execution of dependent rules is done, and completed in single threads, it will be needed to share the result of the threads and make a decision to choose the best result to continue the execution of model. To do this, a thread block which belongs to the current single threads can exchange the result via shared memory. The time cost of KP-MObPSO and Embedded KP-MObPSO-SVM on GPU, CPU and Multicore are compared in the Table 3 which indicates a significant improvement in time cost via executing both KP-MObPSO and Embedded KP-MObPSO-SVM on GPU. In 25 particles, KP-MObPSO takes 3.68 min to complete a set of 100 iteration. While the multicore due to frequent visiting of client to exchange the result was not capable of improving time complexity, GPU does better. In GPU-based execution of KP-MObPSO, 25 particles take 73.87 sec to complete. Therefore, a four times improvement has happened in time efficiency. In terms of KP-MObPSO-SVM, 8 min execution time have not improved by multicore while GPU has improved the efficiency to 5-fold. The big O calculation indicates the time efficiency of the proposed KP-MObPSO-SVM improved from O(M2)+O(MQ2) in sequential method to O(M2/P2)+O(Q2/P2) in parallel execution.

Acknowledgment

The work of N. Elkhani and R. C. Muniyandi has been supported by FRGS/1/2015/ICT04/UKM /02/3, National University of Malaysia, Ministry of Higher Education, Malaysia. The work of G. Zhang was supported by National Natural Science Foundation of China (61672437, 61702428) and by Sichuan Science and Technology Program (18ZDYF2877, 18ZDYF1985, 2017GZ0159).

Bibliography

[1] Alelyani, S.; Tang, J.; Liu, H. (2013); Feature Selection for Clustering: A Review, *Data Clustering: Algorithms and Applications*, 29, 110-121, 2013.

[2] Alhazov, A.; Freund, R.; Heikenwalder, H.; Oswald, M; Rogozhin, Y.; Verlan, S. (2012); Sequential P systems with regular control, Paper presented at the *International Conference on Membrane Computing*, 2012.

[3] Cabarle, F. G. C.; Adorna, H.; Martinez-Del-Amor, M. A.; Perez-Jimenez, M. J. (2012); Improving GPU simulations of spiking neural P systems, *Romanian Journal of Information Science and Technology*, 15(1), 5-20, 2012.

[4] Cecilia, J. M.; Garcia, J. M.; Guerrero, G. D.; Martinez-del-Amor, M. A.; Perez-Hurtado, I.; Perez-Jimenez, M. J. (2009), Simulation of P systems with active membranes on CUDA, *Briefings in bioinformatics*, 11(3), 313-322, 2009.

[5] Cecilia, J. M.; Garcia, J. M.; Guerrero, G. D.; Martinez-del-Amor, M. A.; Perez-Hurtado, I.; Perez-Jimenez, M. J. (2010); Simulating a P system based efficient solution to SAT by using GPUs, *The Journal of Logic and Algebraic Programming*, 79(6), 317-325, 2010.

[6] Cano, A.; Zafra, A.; Ventura, S. (2010); Solving classification problems using genetic programming algorithms on GPUs, *Hybrid Artificial Intelligence Systems*, 17-26, 2010.

[7] Dematte, L.; Prandi, D. (2010); GPU computing for systems biology, *Briefings in bioinformatics*, 11(3), 323-333, 2010.

[8] Elkhani, N.; Chandren Muniyandi, R. (2017); A Multiple Core Execution for Multiobjective Binary Particle Swarm Optimization Feature Selection Method with the Kernel P System Framework, *Journal of Optimization*, 2017.

[9] Elkhani, N.; Muniyandi, R. C. (2015); Membrane computing to model feature selection of microarray cancer data, *Proceedings of the ASE BigData & SocialInformatics*, 2015.

[10] Garcia-Quismondo, M.; Perez-Jimenez, M. J. Implementing ENPS by Means of GPUs for AI Applications, *Proc. Beyond AI: Interdisciplinary Aspects of Artificial Intelligence*, 27-33, 2011.

[11] Gheorghe, M.; Ceterchi, R.; Ipate, F.; Konur, S.; Lefticaru, R. (2018); Kernel P systems: from modelling to verification and testing, *Theoretical Computer Science*, 724, 45-60, 2018.

[12] Gheorghe, M.; Ipate, F.; Dragomir, C.; Mierla, L.; Valencia-Cabrera, L.; Garcia-Quismondo, M.; Perez-Jimenez, M. J. (2013); Kernel P Systems-Version I, Membrane Computing, *Eleventh Brainstorming Week*, BWMC, 97-124, 2013.

[13] Guzzi, P. H.; Agapito, G.; Cannataro, M. (2014); coreSNP: Parallel processing of microarray data, *IEEE Transactions on Computers*, 63(12), 2961-2974, 2014.

[14] Kentzoglanakis, K.; Poole, M. (2012); A swarm intelligence framework for reconstructing gene networks: searching for biologically plausible architectures, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 9(2), 358-371, 2012.

[15] Li, J.-M.; Wang, X.-J.; He, R.-S.; Chi, Z.-X. (2007); An efficient fine-grained parallel genetic algorithm based on GPU-accelerated, *Network and parallel computing workshops, 2007, NPC workshops, IFIP international conference on*, 855-862, 2007.

[16] Liu, J.; Iba, H.; Ishizuka, M. (2001); Selecting informative genes with parallel genetic algorithms in tissue classification, *Genome Informatics*, 12, 14-23, 2009.

[17] Maroosi, A.; Muniyandi, R. C. (2013); Accelerated simulation of membrane computing to solve the n-queens problem on multi-core, *International Conference on Swarm, Evolutionary, and Memetic Computing*, 257-267, 2013.

[18] Maroosi, A.; Muniyandi, R. C. (2013); Membrane computing inspired genetic, *Journal of Computer Science*, 9(2), 264-270, 2013.

[19] Mussi, L.; Daolio, F.; Cagnoni, S. (2011); Evaluation of parallel particle swarm optimization algorithms within the CUDA(TM) architecture, *Information Sciences*, 181(20), 4642-4657, 2011.

[20] Nobile, M.; Besozzi, D.; Cazzaniga, P.; Mauri, G.; Pescini, D. (2012); A GPU-based multi-swarm PSO method for parameter estimation in stochastic biological systems exploiting discrete-time target series, *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics*, 74-85, 2012.

[21] Nobile, M. S.; Besozzi, D., Cazzaniga; P., Pescini, D.; Mauri, G. (2013); Reverse engineering of kinetic reaction networks by means of Cartesian Genetic Programming and Particle Swarm Optimization, *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 1594-1601, 2013.

[22] Pospichal, P.; Jaros, J.; Schwarz, J. (2010); Parallel genetic algorithm on the cuda architecture, *Applications of Evolutionary Computation*, 442-451, 2010.

[23] Sarkar, B. K.; Sana, S. S.; Chaudhuri, K. (2011); Selecting informative rules with parallel genetic algorithm in classification problem, *Applied Mathematics and Computation*, 218(7), 3247-3264, 2011.

[24] Slavik, M.; Zhu, X.; Mahgoub, I.; Shoaib, M. (2009); Parallel Selection of Informative Genes for Classification, *Bioinformatics and Computational Biology. Lecture Notes in Computer Science*, 5462, 388-399, 2009.

[25] Van Nguyen, D. K.; Gioiosa, G. (2010); A region-oriented hardware implementation for membrane computing applications, *Membrane Computing. WMC 2009. Lecture Notes in Computer Science*, 5957, 385-409, 2009.

[26] Zhang, G.; Perez-Jimenez, M. J.; Gheorghe, M. (2017), *Real-life applications with membrane computing*, (Vol. 25): Springer, 2017.

[27] 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.

[28] Zhou, Y.; Tan, Y. (2009); GPU-based parallel particle swarm optimization, *Evolutionary Computation, 2009, CEC'09. IEEE Congress on*, 1493-1500, 2009.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 337-352, June 2018.

A Multi-criteria Decision-making Model for Evaluating Suppliers in Green SCM

W. Jiang, C. Huang

Wen Jiang*, Chan Huang
School of Electronics and Information,
Northwestern Polytechnical University
Xi'an, Shaanxi Province, 710072, China
*Corresponding author: jiangwen@nwpu.edu.cn
huangchan@mail.nwpu.edu.cn

Abstract: In order to develop recycle economy and friendly saving environment, many business enterprises have deployed green supply chain management (GSCM) practices. By employing related theorise of GSCM, organizations expect to minimize the environment impact caused by their commercial and industrial activities in supply chain. Different suppliers may provide different GSCM practices, so evaluating their GSCM performance to rank the green suppliers is an important aspect in practice. In this paper, a novel decision method named fuzzy generalized regret decision-making method is proposed. The fuzzy generalized regret decision-making method is based on ordered weighted averaging (OWA) operator, which is used to effectively aggregate individual regrets related to all stats of nature for an alternative under fuzzy decision-making environment. By combing the proposed method with the application background of GSCM practices, a novel fuzzy decision model for evaluating GSCM performance is further proposed. In the proposed model, the regret of decision maker is taken into consideration with an aim of minimizing the dissatisfaction when choosing the best green supplier. Individual regrets related to all criteria for a green supplier are aggregated to obtain effective regret. Finally, the green suppliers can be ranked according to the effective regrets. A numerical example is used to illustrate the effectiveness of the proposed method.
Keywords: generalized regret decision making; green supply chain; multi-criteria decision making; fuzzy set theory.

1 Introduction

As the awareness of environment protection is increasing and the concerning regulations from government become more strict, green supply chain management (GSCM) plays more and more important role nowadays [2, 7, 17]. GSCM is widely used particularly in commercial and industrial applications all over the world [1,20,37]. The main purpose of using GSCM is to avoid the negative effects on the environment caused by the commercial and industrial activities [31,59]. A lot of companies have adopted concerning theory of GSCM in order to reduce the environmental and legal risk during the supply chain and enhance international competitiveness [6].

When applying GSCM practices, it is very necessary for companies to evaluate their own GSCM performance. Besides, some companies also need to green their supply chains by selecting the better supplier from the existing green suppliers [25, 26]. Therefore, an effective tool for evaluating GSCM performance is vital in practice [22, 30, 40]. Many factors should be taken into consideration during the green supply chain, such as production, material, transformation, storage, purchasing, after-sales service and so on [3, 5, 28, 50, 60]. It is common that information is related to multiple aspects in many applications [9, 23, 51]. Actually, evaluating and selecting suppliers is a multi-criteria decision making problem [18,48]. It is usually inevitable that information contains some uncertainty when the suppliers are evaluated by human judge. Therefore the

environment is fuzzy for majority of multi-criteria decision making problems in practice. Many multi-criteria decision making methods have developed in fuzzy environment [21,24,39]. Among these methods, AHP(Analytical Hierarchy Process) and TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) method are the most popular [21,38]. Besides, many other theories are developed and applied to supplier evaluation and selection, such as ANP(Analytic Network Process), VIKOR(VlseKriterijumska Optimizacija I Kompromisno Resenje), DEMATEL(D ecision-making Trial and Evaluation Laboratory), PROMETHEE(Preference ranking organisation method for enrichment of evaluations), COPRAS(Complex Proportional Assessment) and so on. For instance, in [20] when rating and selecting of potential suppliers, economics (cost), operational factors (quality and delivery), and environmental criteria, recycle capability and GHG emission control are considered by using Fuzzy TOPSIS method. Later the PROMETHEE method is proposed to rank the suppliers according to each decision maker's preferences in [20]. In [40] the multiple criteria evaluation method for green supply programs is based on integrating rough set theory elements and fuzzy TOPSIS. Literature [49] introduces a new decision framework to evaluate GSCM practices by combining Monte Carlo simulation, AHP and VIKOR methods under fuzzy environment. However, few work pay attention to apply regret theory to solve the problem of evaluating green supplier's performance in green supply chain.

The regret decision theory is developed by Loomes and Sugden [42], Bell [4] and Fishburn [19]. For a basic regret decision-making model, there are several different regrets for an alternative under different states of nature. The received payoffs exist some difference when decision makers choosing different alternatives under each state of nature. The regret value is defined as a reflection of the difference between the payoff from the choice of alternative and the best payoff from another choice of alternative under the same state of nature. Different from the classic decision making theory, the basic point of regret decision making is trying to minimize the dissatisfaction when not making the best decision. By applying the regret decision theory, the decision maker can choose the alternative that bringing the minimum regret. Therefore the dissatisfaction of decision maker is minimum if the alternative with the minimum regret is carried out. In this sense, the alternative with the minimum regret is the best choice, which is the most reliable one and can bringing the best payoff compared other alternatives. Naturally, the aim is to find the alternative with the minimum regret in regret decision-making model. So it is a key step to find an exact regret associated with all states of nature for each alternative. However, the regret aggregation method has some limitations in basic regret decision-making theory. For this reason, Yager proposed the generalized regret decision-making method based on OWA operator [55]. This method provides a parameterized family of operations, which can be used to more effectively aggregate an alternative's individual regrets than the basic regret decision theory [55]. However Yager's method is designed for certain decision environment and cannot settle the problem in fuzzy environment. In practice, the information obtained often is uncertain [10, 13, 19]. Many decision-making problems are under uncertain environment [8, 10, 36, 39]. In these cases, the method proposed by Yager is not applicable even though this method is an effective tool to handle multi-decision making problem.

The main contribution of this paper is summarized as follows. First, a new fuzzy multiple criteria decision-making method based on regret theory is proposed. Extended the Yager's generalized decision-making method, the proposed method, called fuzzy generalized decision-making method, can effectively handle the decision making problems in fuzzy environment. Second, the proposed method is combined with the application background of GSCM practices, a novel multi-criteria decision-making model for evaluating the green supplier's performance is further proposed. By aggregating the individual regrets associated to a supplier, the effective regret can be obtained. The effective regret of a green supplier is smaller, and the decision makers

are more satisfied with the decision of choosing this supplier. Our aim is to select the supplier with the minimum effective regret, which is the best supplier. According to all effective regrets obtained, suppliers can be ranked. A numerical example is used to illustrate the effectiveness of the proposed decision model.

The rest of this paper is organized as follows. Section 2 shows the basic concepts, including fuzzy set theory, OWA operator and presented regret type decision making. In Section 3, the proposed decision-making method and decision model for suppliers evaluation are introduced in detail. Section 4 presents a numerical application to illustrate the effectiveness of the proposed model. In the end, conclusion and the future works are shown in Section 5.

2 Preliminaries

2.1 Fuzzy set theory

The fuzzy set theory proposed by Zadeh [58] is widely applied in many fields. Nowadays many studies are related to the fuzzy set theory, such as intelligent event process [43, 52, 57], evidence theory [7,9,33,34], aggregation operator [49,56,74], decision making [27,76] and so on [16,63]. Let X be the universe of discourse, $X = \{x_1, x_2, \cdots, x_n\}$, a fuzzy set $\tilde{A}$ defined on X is characterized by a membership function $\mu_{\tilde{A}}(x)$, which can be denoted as: $\tilde{A} = \{\langle x, \mu_{\tilde{A}}(x)\rangle \,|x \in X\}$, $\mu_{\tilde{A}}(x) \to [0, 1]$. $\mu_{\tilde{A}}(x)$ indicates the degree of $x \in X$ in $\tilde{A}$.

Triangular fuzzy number is a special type of fuzzy sets, $\mu_A(x)$ for a triangular fuzzy number $A = (a, b, c)$ is defined as following:

$$\mu_A(x) = \begin{cases} 0, & x < a \\ \frac{x-a}{b-a}, & x < a, x < c, a \le x \le b \\ \frac{c-x}{c-b}, & b \le x \le c \\ 0, & x > c \end{cases} \tag{1}$$

Let $A = (a_1, b_1, c_1)$ and $B = (a_2, b_2, c_2)$ be two triangular fuzzy numbers. Some basic arithmetic operations for triangular fuzzy numbers are given as follows [41]:

$$A+B = (a_1, b_1, c_1) + (a_2, b_2, c_2) = (a_1 + a_2, b_1 + b_2, c_1 + c_2) \tag{2}$$

$$A - B = (a_1, b_1, c_1) + (a_2, b_2, c_2) = (a_1 - a_2, b_1 - b_2, c_1 - c_2) \tag{3}$$

$$\lambda A = (\lambda a_1, \lambda b_1, \lambda c_1) \tag{4}$$

where λ is a real number.

The distance between two triangular fuzzy numbers is a basic concept for triangular fuzzy number. There are many methods for calculating the distance. One popular and classical distance function is defined as follows [15]:

$$d(A, B) = \sqrt{[(a_1 - a_2)^2 + (b_1 - b_2)^2 + (c_1 - c_2)^2]} \tag{5}$$

The similarity of two triangular fuzzy numbers A and B can be defined as [29]:

$$s(A, B) = 1 - \frac{|a_1 - a_2| + |b_1 - b_2| + |c_1 - c_2|}{3} \tag{6}$$

Obviously, the value of $s(A, B)$ is larger, the two triangular fuzzy numbers A and B are more similar.

2.2 Ordered weighted averaging (OWA) operator

Averaging operator proposed by Yager is a important tool for information fusion [46, 57]. An OWA operator of dimension m is a mapping OWA: $R^m \to R$ that has an associated m dimensional weighting vector $w = (w_1, w_2, \cdots w_m)$. If $j = 1, \cdots, m$ the y_j are a collection of numeric values. Then the OWA aggregation of these value is

$$OWA(y_1, y_2, \cdots, y_m) = \sum\nolimits_{k=1}^{m} w_k y_{\rho(k)} \tag{7}$$

where $w_k \in [0, 1]$ and $\sum_{k=1}^{m} w_k = 1$. $\rho(k)$ is the index of the kth largest of y_j.

A number of approaches have been introduced to obtain the OWA weights [47]. In [56] Yager introduced a functional method of obtaining w_k from function $f : [0, 1] \to [0, 1]$:

$$w_k = f(\frac{k}{n}) - f(\frac{k-1}{n}), \quad k = 1, 2, \cdots, n \tag{8}$$

where function f satisfies $f(x) \geq f(y)$ if $x > y$, and $f(0) = 0$ and $f(1) = 1$.

2.3 Generalized regret decision-making theory

For a decision-making problem, assume there are m possible states of nature and n alternatives, then this problem can be shown as following in a matrix $S = [s_{ij}]_{m\times n}$, where s_{ij} is the payoff received from the result of decision about jth alternative under ith state of nature. According to the basic regret decision making theory, the decision of selecting an alternative should meet the desire that the regret of this choice is minimum [4,42]. For this point, the regret matrix V is $R = [r_{ij}]_{m\times n}$, where $r_{ij} = s_{i\max} - s_{ij}$ represents the difference between the payoff $s_{i\max}$ received from the jth alternative and the maximal payoff s_{ij} received from another alternative under ith state of nature. The regret $R_j = Agg_{i=1\ to\ m}[r_{ij}]$ for each alternative is calculated and then select the alternative with minimum regret $R' = Minn_{j=1\ to\ n}[R_j]$.

In the framework of generalized regret type decision-making based on OWA, the equation of calculating aggregated regret R_j is defined as [55]:

$$R_j = OWA(r_{1j}, r_{2j}, \cdots, r_{mj}) = \sum\nolimits_{k=1}^{m} w_k r_{jp_j(k)} \tag{9}$$

One condition is that a smaller regret is assigned no more weight than a bigger regret. That is if $r_{i\rho_i(k_1)} > r_{i\rho_i(k_2)}$ then $w_{k_1} > w_{k_2}$, which ensures a greater regret to dominate among all individual regrets for an alternative. The OWA weights of regrets can be obtained by Eq. (8), in which one feasible function f is $f(x) = x^r$, and $r \in (0, 1)$ [55]. So the equal for obtaining OWA weights of regrets is shown as follows [55]:

$$R_j = \sum\nolimits_{k=1}^{m} w_k r_{jp_j(k)} = \sum\nolimits_{k=1}^{m} ((\frac{k}{m})^r - (\frac{k-1}{m})^r) r_{jp_j(k)} \tag{10}$$

3 The proposed method and model for GSCM practices

3.1 The proposed fuzzy generalized regret decision-making method

The generalized regret decision-making method proposed by Yager is an effective decision making-method and overcomes the limitations of the basic regret decision making method. However this method is designed for exact number and can not be applied for fuzzy environment. As we all know many decision-making problems are presented in the fuzzy environment in practice. For this issue, a fuzzy generalized regret decision making method is proposed in this paper. The

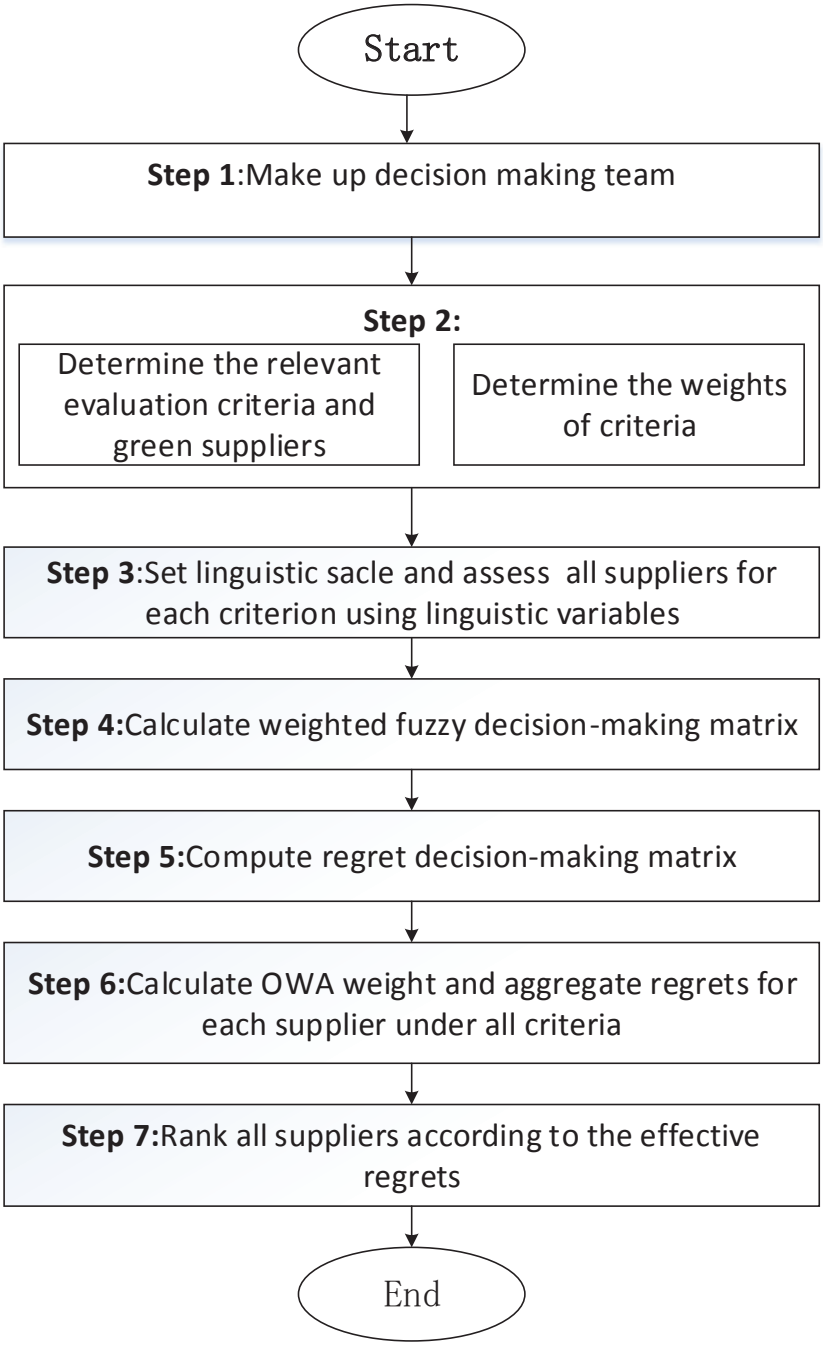


Figure 1: The proposed model for ranking green suppliers in GSCM practice

proposed method extends Yager's generalized regret decision making method to the fuzzy environment and this can be used to effectively handle fuzzy decision making problem. The proposed fuzzy generalized regret decision-making method consists of the following steps:

Step 1: For a decision making problem assume C_i for $i = 1$ to m is the state of nature ,the existing alternative is A_j for $i = 1$ to n. And the weight of the states of nature should be taken into consideration, which is expressed by triangular fuzzy number and is called the fuzzy weight in this paper. Then matrix can be indicated as:

$$S = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1n} \\ s_{21} & s_{22} & \cdots & s_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{m1} & s_{m2} & \cdots & s_{mn} \end{bmatrix} \tag{11}$$

where $s_{ij} = (a_{ij}, b_{ij}, c_{ij})$, which is a triangular fuzzy number, represents the payoff for alternative j under state of nature i.

Step 2: The fuzzy weight of each states of nature is transformed to crisp number, then the matrix multiplied by defuzzified weights of the decision criteria is transformed to weighted fuzzy-decision matrix:

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix} \tag{12}$$

where $v_{ij} = v_{ij} \cdot w_i$ and w_i is the defuzzified weight of the ith state of nature.

Step 3: In this step the regret matrix is expected to get from the weighted fuzzy-decision matrix. Assume $v_{i\max}$ is the maximum fuzzy number of ith row matrix.

First the maximum $v_{i\max}$ under the state of nature C_i is obtained by sorting the triangular fuzzy numbers of ith row of matrix V. According to the regret theory, the regret value r_{ij} is represented by the difference between matrix element v_{ij} and the maximum $v_{i\max}$. A difference measure dif is given based on the similarity of two triangular fuzzy numbers in this paper and is defined as:

$$dif(v_{ij}, v_{i\max}) = 1 - s(v_{ij}, v_{i\max}) \tag{13}$$

Obviously the value of $dif(v_{ij}, v_{j\max})$ is larger, the difference degree between two triangular fuzzy numbers v_{ij} and $v_{i\max}$ is bigger. Based on the difference measure dif, the regret matrix R is obtained as:

$$R = [r_{ij}]_{m\cdot n}\,, \quad i = 1, 2, \cdots, m; j = 1, , 2, \cdots, n$$

where $r_{ij} = dif(v_{ij}, v_{j\max})$.

Step 4: Since the regret matrix elements obtained though the last step are crisp numbers, thus the effective regret of each alternative can be calculated by Eq. (9).

Step 5: Finally all the alternatives should be ranked according to the effective regrets obtained in the step 4. The effective regret value is smaller indicating the corresponding alternative is better. Among all alternatives the best one is the one with the minimum effective regret.

3.2 The decision model for GSCM practice

The aim of using GSCM practices is developing friendly environment and reduce the adverse effects on the environment. It is important to evaluate suppliers according to their comprehensive performance about GSCM's criteria. In this part, based on the proposed fuzzy generalized regret decision-making method, a new decision model is proposed to evaluate the green supplier's

performance in GSCM. When using the proposed model evaluates the performance of suppliers, different criteria can be seen as different states of nature and green suppliers are equivalent to alternatives. Assessments made by experts reflect the performance of green suppliers under different criteria. The assessment matrix is the decision-making fuzzy matrix in the proposed method. The process of ranking green suppliers with the proposed model is shown in Figure 1 and indicated as follows.

Step 1: The environmental experts are selected to form a decision-making team.

Step 2: Experts determine the relevant criteria for selecting and evaluating green suppliers. Then the weights of all criteria are given by experts according to the importance of each criterion.

Step 3: Set the appropriate linguistic scale for green suppliers related to criteria for alternatives. Those linguistic scale is used to evaluate all green suppliers under each criterion by environmental experts. Then those linguistic scale are transformed to scale with fuzzy numbers, which is used to transform the linguistic assessments to the assessment matrix expressed by fuzzy number.

Step 4: According to the weights of each criterion, the assessment matrix are disposed to get the weighted fuzzy assessment matrix. First assessments from different experts for each criterion should be fused into one fuzzy number so that fused assessment matrix is obtained. In addition, the weights of criteria with fuzzy number is transformed to crisp number. Then the weighted fuzzy assessment matrix V is obtained by multiplying the defuzzified weight and fused assessment matrix.

Step 5: The regret decision-making matrix is calculated according to weighted fuzzy assessment matrix V. First by ranking the triangular fuzzy numbers, the maximal fuzzy number $v_{i\max}$ is selected from ith row of weighted fuzzy assessment matrix V. Then based on matrix V and $v_{i\max}$ for $i = 1$ to $i = m$, the regret decision-making matrix R is obtained by Eq. (6).

Step 6: OWA weight is calculated by Eq. (10) and individual regrets under each criteria for each green suppliers are aggregated in this step. In order to aggregate the regrets based on OWA operator, each column elements in matrix R is sorted in descending order to get $r_{jp_j(k)}$. Then Eq. (7) is used to aggregate all individual regrets under each criteria for each supplier to obtain effective regrets.

Step 7: Alternative suppliers can be ranked according to the effective regrets. The alternative with smaller effective regret, then it ranks higher.

4 An illustrative application and discussing

In this section, a numerical example from [49] is used to show the proposed decision model in detail and illustrate its effectiveness. It is considered that a manufacture employs GSCM practices and needs to evaluate four green suppliers according to their performance in GSCM. And the managers of this manufacture make use of our proposed decision model to make this evaluation.

First, the basic example data in [49] is given. The criteria for evaluating four suppliers is shown in Table 1. Those criteria are divided into four groups, which is inbound,operations,production operations, outbound operations, and reverse logistics and four suppliers. Each criterion group consists of several sub-criteria. Globe fuzzy weights of all criteria based on the importance of each criterion and linguistic assessments of three experts are presented in Table 3. In addition, linguistic scale is shown in Table 4.

Then the assessment matrix is disposed to get the weighted fuzzy assessment matrix. Assessments of three experts can be fussed by equal because there is no weight information about these experts. Let $A = (a_1, b_1, c_1)$, $B = (a_2, b_2, c_2)$, $C = (a_3, b_3, c_3)$ be the assessments given by three experts, then those three assessments can be fused by $D = \frac{1}{3} \times (A + B + C)$. The fused

assessments matrix S is shown in Table 4. The fused assessments matrix is weighted to ensure that can reflect the importance degree of criteria. The weights of criteria can be transformed to the crisp number first. The weight $\tilde{W} = (w_1, w_2, w_3)$ of each criterion is defuzzified by using following function:

$$W = \frac{(w_1 + 4w_2 + w_3)}{6} \tag{14}$$

After defuzzfying, the fuzzy weights are transformed to crisp numbers. By multiplying the defuzzified weight and fussing assessment matrix, the crisp weights and a weighted fuzzy assessment matrix V are shown in Table 5.

The next step is to calculate the regret decision-making matrix according to weighted fuzzy assessment matrix V. First, all triangular fuzzy numbers need to be ranked to find the the maximal fuzzy number $v_{i\max}$. It can be seen that any two triangular fuzzy numbers $V_1 = (a_1, b_1, c_1)$ and $V_2 = (a_2, b_2, c_2)$ of case in [49] always satisfy $b_1 > b_2$ and $c_1 > c_2$ if $a_1 > a_2$. Distance between two triangular fuzzy numbers defined in Section 2, which is commonly applied to rank triangular fuzzy number. The value of distance can be calculated by Eq. (5 in this example. It is noted that the best assessment given by expert is VG with triangular fuzzy scale $(7, 9, 10)$. Let $L = (7, 9, 10)$, $v_{i\max} = Max[v_{ij}]$. Thus the distance between L and $v_{i\max}$ is minimum among all elements in row i. It can be seen that any two triangular fuzzy numbers $V_1 = (a_1, b_1, c_1)$ and $V_2 = (a_2, b_2, c_2)$ of case in [49] always satisfy $b_1 > b_2$ and $c_1 > c_2$ if $a_1 > a_2$. Eq. (13) is used to calculate distance between L and v_{ij} so that we can find $v_{i\max}$ according to the minimum value of distance. The result of largest fuzzy numbers of each row is showed in Table 5. Based on the data in Table 5, the fusing regret matrix R shown in Table 6 is obtained by Eq. (6).

At this stage, each column elements in matrix R are sorted in descending in order to get $r_{jp_j(k)}$. The effective regrets for four suppliers can be obtained by Eq. (10). Same as the case for calculating effective regrets in literature [55], r in Eq. (10) takes 0.5 in this case. And there are 18 criteria, so m in Eq.(10) is 18. OWA weights W_k and $r_{jp_j(k)}$ is shown in Table 7.

After obtaining the effective regrets, the alternative suppliers can be ranked. The greater regret for a supplier, the higher the ranking. Finally, alternative suppliers can be ranked according to the effective regrets. The result of effective regrets and the both ranking result obtained by proposed model and by the method in [49] are presented in Table 8. According to the ranking result shown in Table 8, supplier D ranks number 1, which shows that the performance of supplier D is the best among four suppliers. Therefore, the decision of this GSCM problem is choosing supplier D to support GSCM practice for this manufacture.

By comparing two ranking results shown in Table 8, it can be seen that both two results support that: (1) supplier A and supplier D rank higher than supplier B and supplier C, that is to say, the performance of supplier A and supplier D are better than supplier B and supplier C; (2) supplier C rank higher than supplier D, that is, supplier C is better than supplier B. The difference of two ranking results is the ordering of supplier A and supplier D. The ranking result obtained by the proposed method shows the supplier D rank higher than supplier A. However, the result obtained by method in [49] shows that supplier A and supplier D both rank number one.

In [49] supplier A and supplier D both rank number one, which means there is no difference between good or bad for supplier A and supplier D. So in practice it is difficult for managers to make a choice between supplier A and supplier D. However, the proposed method overcomes this shortcoming. According to Table 8, it is known that the effective regrets of supplier A and supplier D are much lager than supplier B and supplier C and the regret of supplier A is lager than supplier D. This result indicates that supplier A and supplier D are better alternatives for

Table 1: Criteria for evaluating GSCM [49]

The group of the criteria	Criteria	Concrete content of criteria
inbound operations	C1.1	Choosing suppliers by environmental criteria
	C1.2	Guiding suppliers to establish their own environmental programs
	C1.3	Urging/pressuring suppliers to take environmental actions
	C1.4	Purchasing environment friendly items
production operations	C2.1	Design products for recycling
	C2.2	Using cleaner technology
	C2.3	Improving capacity utilization
	C2.4	Promoting remanufacturing
outbound operations	C3.1	Enhancing vehicle operating efficiency
	C3.2	Encouraging eco-driving
	C3.3	Using environmental friendly packaging
	C3.4	Reducing empty running
	C3.5	Improving vehicle routing using GPS (Global Positioning System) and other systems
	C3.6	Increasing vehicle payload capacity
reverse logistics	C4.1	Re-use of products and components
	C4.2	Recycling of materials
	C4.3	Waste management
	C4.4	Taking back packaging

Table 2: Linguistic assessments of suppliers and Weights of criteria [49]

Criteria	Supplier A	Supplier B	Supplier C	Supplier D	Weight
C1.1	VG, G, G	F, G, P	F, G, G	G, VG, F	(0.03, 0.05, 0.08)
C1.2	G, F, P	VG, G, G	G, F, F	VG, VG, G	(0.03, 0.05, 0.08)
C1.3	P, F, F	F, P, G	P, F, VP	G, F, VG	(0.04, 0.07, 0.12)
C1.4	F, G, G	F, G, P	G, F, VG	F, VG, F	(0.01, 0.02, 0.04)
C2.1	F, P, P	F, G, F	F, G, F	VG, G, G	(0.03, 0.05, 0.09)
C2.2	F, G, F	G, VG, VG	F, F, G	VG, G, VG	(0.03, 0.06, 0.10)
C2.3	VG, G, VG	P, F, P	F, G, G	VG, G, VG	(0.05, 0.09, 0.16)
C2.4	G, VG, VG	P, P, F	F, P, P	P, G, F	(0.02, 0.04, 0.07)
C3.1	VG, G, G	G, P, G	F, G, F	VG, F, G	(0.03, 0.06, 0.10)
C3.2	VG, G, G	F, P, G	G, VG, VG	F, G, G	(0.04, 0.07, 0.11)
C3.3	VG, G, VG	G, F, P	F, G, F	VG, G, G	(0.05, 0.11, 0.20)
C3.4	G, F, VG	P, F, F	F, G, G	VG, VG, VG	(0.03, 0.05, 0.09)
C3.5	G, F, F	P, VP, VP	F, G, F	P, VP, F	(0.04, 0.06, 0.10)
C3.6	VG, G, G	F, P, F	P, VP, VP	G, VG, G	(0.04, 0.07, 0.12)
C4.1	G, VG, VG	F, G, G	G, VG, F	F, P, G	(0.02, 0.03, 0.06)
C4.2	VG, G, VG	P, F, VP	F, F, G	VG, G, G	(0.02, 0.04, 0.08)
C4.3	F, G, G	F, P, G	G, VG, F	F, P, F	(0.03, 0.06, 0.11)
C4.4	VP, P, P	F, F, G	F, G, G	VG, G, G	(0.02, 0.03, 0.05)

Table 3: Linguistic scale for evaluating GSCM performance of suppliers [49]

Linguistic scale for evaluating suppliers	Triangular fuzzy scale
Very Poor (VP)	(0, 1, 3)
Poor (P)	(1, 3, 5)
Fair (F)	(3, 5, 7)
Good (G)	(5, 7, 9)
Very Good (VG)	(7, 9, 10)

Table 4: The fused assessments matrix [49]

Criteria	Supplier A	Supplier B	Supplier C	Supplier D
C1.1	(5.667, 7.667, 9.333)	(2.333, 4.333, 6.333)	(4.333 ,6.333, 8.333)	(5.000 , 7.000, 8.667)
C1.2	(3, 5, 7)	(5.667, 7.667, 9.333)	(3.667, 5.667, 7.667)	(6.333, 8.333, 9.667)
C1.3	(2.333 ,4.333, 6.333)	(3, 5, 7)	(1.333, 3.000, 5.000)	(5.000, 7.000, 8.667)
C1.4	(4.333, 6.333, 8.333)	(3, 5, 7)	(5.000, 7.000, 8.667)	(4.3333, 6.3333, 8.0000)
C2.1	(1.667, 3.667, 5.667)	(3.667, 5.667, 7.667)	(3.667, 5.667, 7.667)	(5.667, 7.667, 9.333)
C2.2	(3.667, 5.667, 7.667)	(6.333, 8.333, 9.667)	(3.667, 5.667, 7.667)	(6.333, 8.333, 9.667)
C2.3	(6.333, 8.333, 9.667)	(1.667, 3.667, 5.667)	(4.333, 6.333, 8.333)	(6.333,8.333,9.667)
C2.4	(6.333, 8.333, 9.6667)	(1.667, 3.667, 5.667)	(1.667, 3.667, 5.667)	(3, 5, 7)
C3.1	(5.667, 7.667, 9.333)	(3.667, 5.667, 7.667)	(3.667, 5.667, 7.667)	(5.000, 7.000, 8.667)
C3.2	(5.667, 7.667, 9.333)	(3, 5, 7)	(6.333, 8.333, 9.667)	(4.3333, 6.3333, 8.3333)
C3.3	(6.333, 8.333, 9.667)	(3, 5, 7)	(3.667, 5.667, 7.667)	(5.667,7.667,9.333)
C3.4	(5.000, 7.000, 8.667)	(2.333, 4.333, 6.333)	(4.333, 6.333, 8.333)	(7, 9, 10)
C3.5	(3.667, 5.667, 7.667)	(0.333, 1.667, 3.667)	(3.667, 5.667, 7.667)	(1.333, 3.000, 5.000)
C3.6	(5.667, 7.667, 9.333)	(2.333,4.333,6.333)	(0.333,1.667,3.667)	(5.6667, 7.6667, 9.3333)
C4.1	(6.333, 8.333, 9.667)	(4.333, 6.333, 8.333)	(5.000, 7.000, 8.667)	(3, 5, 7)
C4.2	(6.333, 8.333, 9.667)	(1.333, 3.000, 5.000)	(3.667, 5.667, 7.667)	(5.667, 7.667, 9.333)
C4.3	(4.333, 6.333, 8.333)	(3, 5, 7)	(5.000, 7.000, 8.667)	(2.333, 4.333, 6.333)
C4.4	(0.667, 2.333, 4.333)	(3.667, 5.667, 7.667)	(4.333, 6.333, 8.333)	(5.667, 7.667, 9.333)

Table 5: The weighted fuzzy assessment matrix and the best assessment on each criteria

Criteria	Defuzzified weight	Supplier A	Supplier B	Supplier C	Supplier D	The largest fuzzy number
C1.1	0.052	(0.293, 0.396, 0.483)	(0.121, 0.224, 0.327)	(0.224, 0.327, 0.431)	(0.259, 0.362, 0.448)	(0.293, 0.396, 0.483)
C1.2	0.052	(0.155, 0.259, 0.362)	(0.293, 0.396, 0.483)	(0.190, 0.293, 0.396)	(0.327, 0.431, 0.500)	(0.327, 0.431, 0.500)
C1.3	0.073	(0.175, 0.325, 0.475)	(0.225, 0.375, 0.525)	(0.100, 0.225, 0.375)	(0.375, 0.525, 0.650)	(0.375, 0.525, 0.650)
C1.4	0.022	(0.094, 0.137, 0.181)	(0.065, 0.109, 0.152)	(0.109, 0.152, 0.188)	(0.094, 0.137, 0.174)	(0.109, 0.152, 0.188)
C2.1	0.053	(0.089, 0.195, 0.302)	(0.195, 0.302, 0.409)	(0.195, 0.302, 0.409)	(0.302, 0.409, 0.498)	(0.302, 0.409, 0.498)
C2.2	0.062	(0.226, 0.350, 0.473)	(0.391, 0.514, 0.596)	(0.226, 0.345, 0.473)	(0.391, 0.514, 0.596)	(0.391, 0.514, 0.596)
C2.3	0.095	(0.602, 0.792, 0.918)	(0.158, 0.348, 0.538)	(0.412, 0.602, 0.792)	(0.602, 0.792, 0.918)	(0.602, 0.792, 0.918)
C2.4	0.042	(0.264, 0.348, 0.403)	(0.070, 0.153, 0.236)	(0.070, 0.153, 0.236)	(0.125, 0.209, 0.292)	(0.264, 0.348, 0.403)
C3.1	0.062	(0.347, 0.473, 0.576)	(0.226, 0.350, 0.473)	(0.226, 0.350, 0.473)	(0.309, 0.432, 0.535)	(0.350, 0.473, 0.576)
C3.2	0.072	(0.406, 0.550, 0.669)	(0.215, 0.359, 0.502)	(0.454, 0.598, 0.693)	(0.311, 0.454, 0.598)	(0.454, 0.598, 0.693)
C3.3	0.115	(0.728, 0.958, 1.112)	(0.345, 0.575, 0.805)	(0.422, 0.652, 0.882)	(0.652, 0.882, 1.073)	(0.728, 0.958, 1.112)
C3.4	0.053	(0.267, 0.373, 0.462)	(0.124, 0.231, 0.338)	(0.231, 0.338, 0.444)	(0.373, 0.480, 0.533)	(0.373, 0.480, 0.533)
C3.5	0.063	(0.401, 0.528, 0.612)	(0.106, 0.232, 0.359)	(0.274, 0.401, 0.528)	(0.401, 0.528, 0.612)	(0.401, 0.528, 0.612)
C3.6	0.073	(0.415, 0.562, 0.684)	(0.171, 0.318, 0.464)	(0.024, 0.122, 0.269)	(0.415, 0.562, 0.684)	(0.415, 0.562, 0.684)
C4.1	0.033	(0.211, 0.278, 0.322)	(0.144, 0.211, 0.278)	(0.167, 0.233, 0.289)	(0.100, 0.167, 0.233)	(0.211, 0.278, 0.322)
C4.2	0.043	(0.274, 0.361, 0.419)	(0.058, 0.130, 0.217)	(0.159, 0.245, 0.332)	(0.245, 0.332, 0.404)	(0.274, 0.361, 0.419)
C4.3	0.063	(0.274, 0.401, 0.528)	(0.190, 0.317, 0.443)	(0.317, 0.443, 0.549)	(0.148, 0.274, 0.401)	(0.317, 0.443, 0.549)
C4.4	0.032	(0.021, 0.074, 0.137)	(0.116, 0.180, 0.243)	(0.137, 0.201, 0.264)	(0.180, 0.243, 0.296)	(0.180, 0.243, 0.296)

Table 6: Regret matrix

Criteria	Supplier A	Supplier B	Supplier C	Supplier D
C1.1	0	0.167	0.063	0.035
C1.2	0.161	0.029	0.126	0
C1.3	0.192	0.142	0.283	0
C1.4	0.012	0.041	0	0.015
C2.1	0.207	0.101	0.101	0
C2.2	0.151	0	0.151	0
C2.3	0	0.422	0.169	0
C2.4	0	0.185	0.185	0.130
C3.1	0	0.117	0.117	0.041
C3.2	0.040	0.223	0	0.128
C3.3	0	0.358	0.281	0.064
C3.4	0.095	0.231	0.124	0
C3.5	0	0.281	0.113	0
C3.6	0	0.236	0.415	0
C4.1	0	0.059	0.041	0.104
C4.2	0	0.217	0.106	0.024
C4.3	0.035	0.120	0	0.162
C4.4	0.162	0.060	0.039	0

Table 7: Sorted regret values and OWA weights

ρ_k	Supplier A	Supplier B	Supplier C	Supplier D	OWA weight
ρ_1	0.207	0.422	0.415	0.162	0.236
ρ_2	0.192	0.358	0.283	0.130	0.098
ρ_3	0.162	0.281	0.281	0.128	0.075
ρ_4	0.161	0.236	0.185	0.104	0.063
ρ_5	0.151	0.231	0.169	0.064	0.056
ρ_6	0.095	0.223	0.151	0.041	0.050
ρ_7	0.040	0.217	0.126	0.035	0.046
ρ_8	0.035	0.185	0.124	0.024	0.043
ρ_9	0.012	0.167	0.117	0.015	0.040
ρ_{10}	0	0.142	0.113	0	0.038
ρ_{11}	0	0.120	0.106	0	0.036
ρ_{12}	0	0.117	0.101	0	0.035
ρ_{13}	0	0.101	0.063	0	0.033
ρ_{14}	0	0.060	0.041	0	0.032
ρ_{15}	0	0.059	0.039	0	0.031
ρ_{16}	0	0.041	0	0	0.030
ρ_{17}	0	0.029	0	0	0.029
ρ_{18}	0	0	0	0	0.028

Table 8: The effective regrets and the ranking results for four suppliers

	Regret	The proposed model	The method in [49]
Supplier A	0.107	2	1
Supplier B	0.242	4	3
Supplier C	0.210	3	2
Supplier D	0.076	1	1

decision makers, and the best choice is supplier D which has the best performance. From Table 8, the effective regrets of supplier D and supplier A is 0.076, and 0.107 respectively. According to these two regret values, the manager can easily make a clear choice to choose supplier D.

5 Conclusion

A good GSCM is important for reducing the environmental harm cased by industrial activities through the supply chain. The companies need to evaluate their own GSCM performance and select the best supplier. Therefore a good evaluation model plays an important part. In this paper, a novel fuzzy multi-criteria decision making method for GSCM is proposed. In the proposed decision-making method, the regret sense of decision-maker is taken into consideration, which improves the accuracy and reliability of decision making. Besides, instead of using the basic regret decision-making method, the fuzzy generalized regret decision-making method is proposed to obtain effective regret. The generalized regret decision making method based on OWA is used for certain decision-making environment in [55]. On this basic, the proposed method can be used to settle problems under fuzzy decision-making environment. Then the proposed method is implemented to the case about GSCM practices introduced in [49], as a result of which, a more accurate and clearer ranking result about green suppliers is obtained. A comparative analysis of the result verifies the effectiveness and feasibility of the proposed method in this paper. In the future research, the proposed method is expected to extend its applications. More multi-decision making problems in any other application background can be considered to resolve with this method.

Acknowledgments

The work is partially supported by National Natural Science Foundation of China (Program No. 61671384,61703338), the Seed Foundation of Innovation and Creation for Graduate Students in Northwestern Polytechnical University (Program No. ZZ2017126).

Conflict of interests

The authors declare that there is no conflict of interests.

Bibliography

[1] Andiç E., Yurt Ö., and Baltacıoğlu Baltacıoğlu T.(2012); Green supply chains: Efforts and potential applications for the turkish market, *Resources Conservation & Recycling*, 58, 50-68, 2012.

[2] Azevedo S. G., Carvalho H., and Machado V.C. (2011); The influence of green practices on supply chain performance: A case study approach, *Transportation Research Part E Logistics & Transportation Review*, 47(6), 850-871, 2011.

[3] Beamon B. M. (1998); Supply chain design and analysis: Models and methods, *International Journal of Production Economics*, 55(3), 281–294, 1998.

[4] Bell D. E. (1980); *Regret in decision making under uncertainty*, 30, Informs, 1982.

[5] Büyüközkan G. and Çifçi G. (2011); A novel fuzzy multi-criteria decision framework for sustainable supplier selection with incomplete information, *Computers in Industry*, 62(2), 164-174, 2011.

[6] Chan H. K., He H., and Wang W. Y. C. (2012); Green marketing and its impact on supply chain management in industrial markets, *Industrial Marketing Management*, 41(4), 557-562, 2012.

[7] Chien M. K. and Shih L. H. (2007); An empirical study of the implementation of green supply chain management practices in the electrical and electronic industry and their relation to organizational performances, *International Journal of Environmental Science & Technology*, 4(3), 383–394, 2007.

[8] Chao X., Peng Y., and Kou G. (2016); A similarity measure-based optimization model for group decision making with multiplicative and fuzzy preference relations, *International Journal of Computers Communications & Control*, 12(1), 26-40, 2016.

[9] Dou Y., Zhu Q., and Sarkis J. (2014); Evaluating green supplier development programs with a grey-analytical network process-based methodology, *European Journal of Operational Research*, 233(2), 420-431, 2014.

[10] Deng X., Xiao F., and Deng Y. (2017); An improved distance-based total uncertainty measure in belief function theory, *Applied Intelligence*, 46(4), 898-915, 2017.

[11] Deng X. and Jiang W. (2018); Dependence assessment in human reliability analysis using an evidential network approach extended by belief rules and uncertainty measures, *Annals of Nuclear Energy*, 117, 183-193, 2018.

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

[13] Deng X., Han D., Dezert J., Deng Y., and Yu S. (2016); Evidence combination from an evolutionary game theory perspective, *IEEE Transactions on Cybernetics*, 46(9), 2070-2082, 2016.

[14] Deng X. and Jiang W. (2018); An evidential axiomatic design approach for decision making using the evaluation of belief structure satisfaction to uncertain target values, *International Journal of Intelligent Systems*, 33(1), 15-32, 2018.

[15] Deza M. M. and Deza E. (2009); Encyclopedia of distances, *Encyclopedia of Distances*, 24, 1-583, Springer, 2009.

[16] Dzitac I. (2015); The fuzzification of classical structures: A general view, *International Journal of Computers Communications & Control*, 10(6), 772-778, 2015.

[17] Eltayeb T. K.,Zailani S., and Ramayah T. (2011); Green supply chain initiatives among certified companies in malaysia and environmental sustainability: Investigating the outcomes, *Resources Conservation & Recycling*, 55(5), 495-506, 2011.

[18] ElSayed A., Kongar E., and Gupta S. M. (2015); Fuzzy linear physical programming for multiple criteria decision-making under uncertainty, *International Journal of Computers Communications & Control*, 11(1), 26-38, 2015.

[19] Filiz-Ozbay E. and Ozbay E. Y. (2007); Auctions with anticipated regret: Theory and experiment, *American Economic Review*, 97(4), 1407–1418, 2007.

[20] Govindan K., Kadziński M., and Sivakumar R. (2017); Application of a novel prometheebased method for construction of a group compromise ranking to prioritization of green suppliers in food supply chain, *Omega*, 71, 129-145, 2017.

[21] Ghorabaee M.K, Amiri M., and Zavadskas E. K., Antucheviciene and J. (2017); Supplier evaluation and selection in fuzzy environments: a review of madm approaches, *Economic research-Ekonomska istraživanja*, 30(1), 1073-1118, 2017.

[22] Ghorabaee M. K. , Zavadskas E. K., Amiri M., and Turskis Z. (2016); Extended edas method for fuzzy multi-criteria decision-making: an application to supplier selection, *International Journal of Computers Communications & Control*,11(3), 358-371, 2016.

[23] Gencer C. and Gürpinar D. (2007); Analytic network process in supplier selection: A case study in an electronic firm, *Applied Mathematical Modelling*, 31(11), 2475-2486, 2007.

[24] GrecoS., Kadziński M., Mousseau V., and Słowiński R. (2012); Robust ordinal regression for multiple criteria group decision: Uta-group and utadis-group, *Decision Support Systems*, 52(3), 549-561, 2012.

[25] Handfield R., WaltonS. V., Sroufe R., and Melnyk S. A. (2002); Applying environmental criteria to supplier assessment: A study in the application of the analytical hierarchy process, *European Journal of Operational Research*, 141(1), 70-87, 2002.

[26] Humphreys P. K., Wong Y. K., and Chan F. T. S. (2003); Integrating environmental criteria into the supplier selection process, *Journal of Materials Processing Technology*, 138(1), 349–356, 2003.

[27] He Z. and Jiang W. (2018); An evidential dynamical model to predict the interference effect of categorization on decision making, *Knowledge-Based Systems*, p. Published on line. Doi: 10.1016/j.knosys.2018.03.014, 2018.

[28] HervaniA. A., Helms M. M., and Sarkis J. (2005); Performance measurement for green supply chain management, *Benchmarking*, 12(4), 330–353, 2005.

[29] Hsieh C. H. and Chen S. H. (1999); Model and algorithm of fuzzy product positioning, *Information Sciences*, 121(1), 61–82, 1999.

[30] Igarashi M., Boer L. D., and Fet A. M.(2013); What is required for greener supplier selection? A literature review and conceptual model development, *Journal of Purchasing & Supply Management*, 19(4), 247–263, 2013.

[31] Jabbour C. J. C. (2015); Green human resource management and green supply chain management: Linking two emerging agendas, *Journal of Cleaner Production*, 112, 1824–1833, 2015.

[32] Jiang W. and Wei B. (2018); Intuitionistic fuzzy evidential power aggregation operator and its application in multiple criteria decision-making, *International Journal of Systems Science*, 49(3), 582–594, 2018.

[33] Jiang W., Chang Y., and Wang S. (2017); A method to identify the incomplete framework of discernment in evidence theory, *Mathematical Problems in Engineering*, 2017, Article ID 7635972, 2017.

[34] Jiang W. and Hu W. (2018); An improved soft likelihood function for Dempster-Shafer belief structures, *International Journal of Intelligent Systems*, Published on line. Doi: 10.1002/int.219809, 2018.

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

[36] Jiang W., Wei B., Liu X., Li X., and Zheng H. (2018); Intuitionistic fuzzy power aggregation operator based on entropy and its application in decision making, *International Journal of Intelligent Systems*, 33(1), 49–67, 2018.

[37] Kannan D. and Jabbour C. J. C. (2014); Suppliers based on gscm practices: Using fuzzy topsis applied to a brazilian electronics company, *European Journal of Operational Research*, 233(2), 432–447, 2014.

[38] Kaplinski O., Peldschus F., and Tupenaite L. (2014); Development of mcdm methods–in honour of professor edmundas kazimieras zavadskas on the occasion of his 70th birthday, *International Journal of Computers Communications & Control*, 9(3), 305–312, 2014.

[39] Kannan D., Khodaverdi R., Olfat L., Jafarian A., and Diabat A. (2013); Integrated fuzzy multi criteria decision making method and multi-objective programming approach for supplier selection and order allocation in a green supply chain, *Journal of Cleaner Production*, 47(9), 355–367, 2013.

[40] Kusi-Sarpong S., Sarkis J., and Wang X. (2016); Assessing green supply chain practices in the ghanaian mining industry: A framework and evaluation, *International Journal of Production Economics*, 181, 325–341, 2016.

[41] Kaufmann A. (1987); Introduction to fuzzy arithmetic : Theory and applications, *International Journal of Approximate Reasoning*, 1(1), 141–143, 1987.

[42] Loomes G. and Sugden R. (1982); Regret theory: An alternative theory of rational choice under uncertainty, *Economic Journal*, 92(368), 805–824, 1982.

[43] Liang W., He J., Wang S., Yang L., and Chen F. (2018) Improved cluster collaboration algorithm based on wolf pack behavior, *Cluster Computing*, Published online, doi: 10.1007/s10586–018–1891–y, 2018.

[44] Liu W., Liu H. B., and Li L. L. (2017); A multiple attribute group decision making method based on 2-D uncertain linguistic weighted heronian mean aggregation operator, *International Journal of Computers Communications & Control*, 12(2), 254–264, 2017.

[45] Mo H. and Deng Y. (2016); A new aggregating operator in linguistic decision making based on D numbers, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 24(6), 831–846, 2016.

[46] Merigo J. M. and Casanovas M. (2010); The fuzzy generalized owa operator and its application in strategic decision making, *Cybernetics and Systems: An International Journal*, 41(5), 359–370, 2010.

[47] O'Hagan M. (1988) Aggregating template or rule antecedents in real-time expert systems with fuzzy set logic, *Signals, Systems and Computers, 1988. IEEE Conference Twenty-Second Asilomar*, 2, 681–689, 1988.

[48] Rajendran K., S., Sarkis J.,Murugesan and P. (2015); Multi criteria decision making approaches for green supplier evaluation and selection: A literature review, *Journal of Cleaner Production*, 98, 66–83, 2015.

[49] Sari K. (2017); A novel multi-criteria decision framework for evalutaing green supply chain management practices, *Computers & Industrial Engineering*, 105, 338–347, 2017.

[50] Walton S. V., Handfield R. B., and Melnyk S. A. (1998); The green supply chain: Integrating suppliers into environmental management processes, *Journal of Supply Chain Management*, 34(1), 2–11, 1998.

[51] Xiao F. (2017); An improved method for combining conflicting evidences based on the similarity measure and belief function entropy, *International Journal of Fuzzy Systems*,Published online, DOI: 10.1007/s40815–017–0436–5, 2017.

[52] Xiao F. (2016); An intelligent complex event processing with D numbers under fuzzy environment, *Mathematical Problems in Engineering*, 2016, p. Article ID: 3713518, 2016.

[53] Xu S., Jiang W., Deng X., and Shou Y. (2018); A modified physarum-inspired model for the user equilibrium traffic assignment problem, *Applied Mathematical Modelling*, 55, 340–353, 2018.

[54] Xu H. and Deng Y. (2018); Dependent evidence combination based on shearman coefficient and pearson coefficient, *IEEE Access*, 6, 11634–11640, 2018.

[55] Yager R. R. (2017); Generalized regret based decision making, *Engineering Applications of Artificial Intelligence*, 65, 400–405, 2017.

[56] Yager R. R. (1996); Quantifier guided aggregation using owa operators, *International Journal of Intelligent Systems*, 11(1), 49–73, 1996.

[57] Yager R. R. (1998); On ordered weighted averaging aggregation operators in multicriteria decisionmaking, *IEEE Transactions on systems, Man, and Cybernetics*, 18(1), 183–190, 1988.

[58] Zadeh L. A. (1965); Fuzzy sets, *Information & Control*, 8(3), 338–353, 1965.

[59] Zhu Q., Sarkis J., and Lai K. H.(2012); Green supply chain management innovation diffusion and its relationship to organizational improvement: An ecological modernization perspective, *Journal of Engineering and Technology Management*, 29(1), 168–185, 2012.

[60] Zhu Q., Sarkis J., and Lai K. H. (2008); Confirmation of a measurement model for green supply chain management practices implementation, *International Journal of Production Economics*, 111(2), 261–273, 2008.

[61] Zhou X., Hu Y., Deng Y., Chan F. T. S., and Ishizaka A. (2018); A DEMATEL-Based completion method for incomplete pairwise comparison matrix in AHP, *Annals of Operations Research*, Published online, doi: 10.1007/s10479–018–2769–3, 2018.

[62] Zheng X. and 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.

[63] Zheng H., Deng Y., and Hu Y. (2017); Fuzzy evidential influence diagram and its evaluation algorithm, *Knowledge-Based Systems*, 131, 28–45, 2017.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 353-364, June 2018.

Implementation of Arithmetic Operations by SN P Systems with Communication on Request

Y. Jiang, Y. Kong, C. Zhu

Yun Jiang*, Chaoping Zhu
1. Detection and Control of Integrated Systems Engineering Laboratory
2. School of Computer Science and Information Engineering
Chongqing Technology and Business University
Chongqing 400067, China
*Corresponding author: jiangyun@email.ctbu.edu.cn
jsjzcp@163.com

Yuan Kong
College of Mathematics and System Science
Shandong University of Science and Technology
Qingdao 266590, China
kongyuan1122@126.com

Abstract: Spiking neural P systems (SN P systems, for short) are a class of distributed and parallel computing devices inspired from the way neurons communicate by means of spikes. In most of the SN P systems investigated so far, the system communicates on command, and the application of evolution rules depends on the contents of a neuron. However, inspired from the parallel-cooperating grammar systems, it is natural to consider the opposite strategy: the system communicates on request, which means spikes are requested from neighboring neurons, depending on the contents of the neuron. Therefore, SN P systems with communication on request were proposed, where the spikes should be moved from a neuron to another one when the receiving neuron requests that. In this paper, we consider implementing arithmetical operations by means of SN P systems with communication on request. Specifically, adder, subtracter and multiplier are constructed by using SN P systems with communication on request.
Keywords: membrane computing, spiking neural P system, communication on request, arithmetic operation.

1 Introduction

Since Gh. Păun first circulated his idea of membrane computing in 1998 [3] [22] (first circulated as a Turku Center for Computer Science (TUCS) Report 208, 1998), membrane computing has developed rapidly for almost two decades. As a branch of natural computing, membrane computing aims on abstract computing ideas from the structure and the functioning of a single cell, and also from complexes of cells, such as tissues and organs (including the brain) [23]. The computational devices in membrane computing are known as membrane systems (P systems, for short). Till now, three main classes of P systems have been investigated: cell-like P systems [22], tissue-like P systems [13, 50] and neural-like P systems [9]. The present paper deals with a class of neural-like P system, called spiking neural P systems (SN P systems, for short) [9].

SN P systems are a class of distributed parallel computing devices inspired from the way the neurons communicate by sending spikes to each other. In SN P systems, neurons (in the form of membranes) are placed in the nodes of a directed graph, with the edges representing synapses. Each neuron contains a number of identical objects, denoted by a and called spikes. Each neuron may also contain a number of firing rules and forgetting rules. When the contents

of a neuron satisfy some regular expression, a firing rule allows a neuron to send information to other neurons in the form of spikes. On the other hand, forgetting rule removes from the neuron a specified number of spikes. The system evolves by means of firing rules and forgetting rules. And it evolves synchronously, in each time unit, each neuron which can use a rule, no matter firing or forgetting, should use one. When the computation halts, no further rule can be used, and a result is obtained, e.g., in the form of the distance between the first two spikes of the output neuron, or the number of spikes present in a specified neuron in the halting configuration.

Since 2006 there have been quite a few research efforts put forward to SN P systems. Many variants of SN P systems have been proposed, such as asynchronous SN P systems [3], sequential SN P systems [8], SN P systems with anti-spikes [16], homogenous SN P systems [47], SN P systems with astrocytes [19], SN P systems with weighted synapses [21], SN P systems with rules on synapses [32], SN P systems with weights [36], SN P systems with a generalized use of rules [52], SN P systems with white hole neurons [27], SN P systems with request rules [30], cell-like SN P systems [43], extended SN P systems [1], SN P systems with scheduled synapses [2], SN P systems with polarizations [19]. Most of the classes of SN P systems obtained are computationally universal, equivalent in power to Turing machines [4, 11, 15, 31, 33, 35, 40, 42, 46, 51, 53]. An interesting topic is to find small universal SN P systems [14,20,25,28,29,44,54]. In certain cases, polynomial solutions to computationally hard problems can also be obtained in this framework [10, 17]. Moreover, SN P systems have been applied to solve real-life problems [24] [21], for example, to design logic gates, logic circuits [34] and databases [5], to represent knowledge [38], to diagnose faults [26,37,39], or to approximately solve combinatorial optimization problems [49].

SN P systems can also be applied in a very different way, where they are viewed as components of a restricted Arithmetic Logic Unit. Some SN P systems were constructed for dealing with basic arithmetic operations. These systems apply different encoding method. In [7], the binary number is encoded as a sequence of spikes: at each time unit, zero or one spike will be provided to the input neuron, depending on the corresponding bit being 0 or 1. The numbers used in [45] are encoded as the interval of time elapsed between two spikes. Under the third encoding mechanism, natural numbers are encoded in the form of spike train and introduced in the system through the input neurons, while the results of arithmetic operations are encoded in the form of the number of spikes emitted to the environment [13].

These SN P systems mentioned above perform communication on command, that is the initiative for communication belongs to the emitting neuron. Specifically speaking, the application of evolution rules depends on the contents of a neuron, (as mentioned above, checked by a regular expression), a specified number of spikes are consumed and a specified number of spikes are produced, and then sent to each neurons linked to the evolving neuron by a synapse. Inspired from parallel-cooperating grammar systems, it is natural to consider the opposite strategy – communication on request. In this case, spikes are requested from neighboring neurons, depending on the contents of the neighboring neuron (also checked by a regular expression). On the other hand, no spike is consumed or created, they are only moved from a neuron to another one along synapses when the receiving neuron requests that. This request-response communication is an important concept in software engineering, and computers use it as a basic method to communicate with each other. Recently, communication on request was introduced into SN P systems by Pan et al., and this variant of SN P systems is shortly called SNQ P systems [18]. Communication on request is a powerful feature in SN P systems: SNQ P systems using two types of spikes are proved to be universal, equivalent with Turing machines, and it is reported that 49 neurons are sufficient for SNQ P systems to achieve Turing universality.

In this work, SN P systems with communication on request for performing the arithmetic operations are introduced. The arithmetic operations we will consider are addition, subtraction and multiplication. Natural numbers can be encoded in the form of the number of spikes and

introduced in the system through input neurons. And then by performing the computation of the system, a number of spikes are present in the output neuron when the system halts. By analyzing the number of specific spikes in the output neuron, we can obtain the result of this arithmetic operation.

The paper is organized as follows. In the next section we recall some preliminaries that will be used in the following, including the formal definition of SN P systems with communication on request. In Section 3.1 we present an SN P system with communication on request that is used to add two natural numbers. A subtracter based on SN P system with communication on request is given in Section 3.2. An SN P system with communication on request for multiplication is constructed in Section 3.3. Conclusions and some open problems for future works are present in Section 4.

2 Spiking neural P systems with communication by request

Formally, a spiking neural P system with communication on request (shortly, SNQ P system), with k types of spikes, is a construct of the form (this form is almost the same as the one appearing in [18], except one reasonable change by introducing input neurons)

$$\Pi = (O, \sigma_1, \sigma_2, \ldots, \sigma_m, a_{i_{in}}, a_{i_{out}}, in, out),$$

where:

1. $O = \{a_1, a_2, \ldots, a_k\}$ is an alphabet (a_i is a type of spikes), $k \geq 1$;
2. $\sigma_1, \sigma_2, \ldots, \sigma_m$ are neurons, of the form

$$\sigma_i = (a_1^{n_1} a_2^{n_2} \ldots a_k^{n_k}, R_i), 1 \leq i \leq m, n_j \geq 0, 1 \leq j \leq k,$$

   where:

   a) $n_j \geq 0$ is the initial number of spikes of type a_j contained in the neuron σ_j, $1 \leq j \leq k$;
   b) R_i is a finite set of rules of the form E/Qw, with w a finite non-empty list of queries of the forms (a_s^p, j) and (a_s^∞, j), $1 \leq s \leq k$, $p \geq 0$, $1 \leq j \leq m$, or $j = env$;

3. $a_{i_{in}}$, $a_{i_{out}}$, $1 \leq i_{in}, i_{out} \leq k$, are the types of input spikes and output spikes,
4. $in \subseteq \{1, 2, \ldots, m\}$ indicate the input neurons, and $out \in \{1, 2, \ldots, m\}$ indicates the output neurons, respectively.

A query (a_s^p, j) means that neuron σ_i requests p copies of a_s from neuron σ_j, while the meaning of (a_s^∞, j) is that all spikes of type a_s from σ_j, no matter how many they are, are requested by σ_i. Specifically, a query of the form (a, env) is allowed to be used, which means that one copy of a is requested from the environment – with the environment supposed to contain arbitrarily many copies of a. This kind of rules can be removed [18], so it will not effect the arithmetic operations.

A rule of the form E/Qw can be used if both of the following conditions are satisfied: (1) the contents of the neuron are described by the regular expression E; (2) all queries formulated in w are satisfied (for example, if σ_j contains strictly less than p spikes, then the query (a_s^p, j) is not satisfiable). Specifically, there is a situation called the conflicting queries, where two different neurons σ_{i_1}, σ_{i_2} ask different numbers of occurrences of the same spike a_s from the same neuron σ_j (namely, two queries of the forms (a_s^p, j), (a_s^r, j) with $p \neq r$, or of the forms (a_s^p, j), (a_s^∞, j) for

p a given number). In the case of conflicting queries, the two rules cannot be used simultaneously, but one of them, non-deterministically chosen, can be used.

In SNQ P systems, the definition of a computational step is quite delicate because of the interplay of the queries. A computational step is described in terms of three sub-steps: (1) In each neuron, a rule is chosen to be applied, and its applicability is checked; (2) The requested spikes are removed from the neurons where they were present. (3) The queries are satisfied, the requested spikes are moved to the requesting neuron. The three sub-steps together form a step, which lasts one time unit.

An SNQ P system starts from the initial configuration, which is described by the number of spikes of each type present in each neuron in the beginning of the computation. Then it proceeds by applying the rules synchronously, which means that in each neuron if a rule can be used, then it is applied according to the procedure described above. After a computation step, we can define transitions configurations. Any sequence of transitions starting from the initial configuration is called a computation. A computation halts if reaches a configuration where no rule can be used. The result of a halting computation is the number of copies of spikes $a_{i_{out}}$ present in neuron σ_{out} in the halting configuration.

In order to perform arithmetic operations, it is necessary to introduce the numbers to be computed into the system, which may be encoded in many different ways. Here, we use the way discussed in [7]. A positive integer number is given as input to a specified input neuron. The number is specified as the number of input spikes initially contained in the input neuron. The result of the operation is encoded as the number of output spikes present in the output neuron when a computation halts.

In the next sections SNQ P systems are represented graphically, which is easy to understand. An oval with the initial number of spikes and rules inside is used to represent a neuron. The input neurons have incoming arrows and the output neuron have outgoing arrows, suggesting their communication with other devices (or the environment).

3 Performing arithmetic operations by SN P systems with communication on request

3.1 An SNQ P system for addition

In this section we present an SN P system with communication on request, as shown in Fig. 1, for dealing with the addition of two arbitrary natural numbers. System Π_{add} is composed of 5 neurons: two specified neurons are used as input neurons, where the summand and addend are introduced, and one neuron is used for giving the obtained result.

Theorem 1. *For two arbitrary natural numbers x and y, SN P system with communication on request Π_{add} computes the addition of x and y.*

Proof: We construct a system Π_{add} of the form

$$\Pi_{add} = (\{a, b, c_1\}, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \{a, b\}, \{a\}, \{1, 2\}, 5),$$

where:
$\sigma_1 = (a^x, \emptyset)$,
$\sigma_2 = (b^y, \emptyset)$,
$\sigma_3 = (\lambda, \{\lambda/Q(c_1, env), c_1/Q(a^\infty, 1)\})$,
$\sigma_4 = (\lambda, \{\lambda/Q(c_1, env), c_1/Q(b, 2), bc_1/Q(a, env)\})$,
$\sigma_5 = (\lambda, \{\lambda/Q(c_1, env), a^*b^*c_1/Q(a^\infty, 3)(ab, 4)\})$.

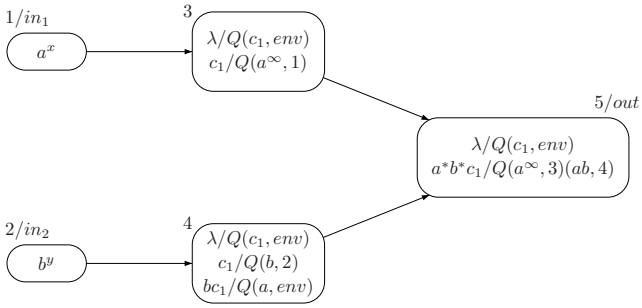


Figure 1: The structure of adder Π_{add}

SNQ P system Π_{add} functions as follows. In the initial configuration of Π_{add}, all neurons are empty. The summand x and the addend y are encoded as spikes a^x and b^y, and are provided to neuron σ_{in_1} and neuron σ_{in_2}, respectively. Since neurons σ_3, σ_4 and σ_{out} are empty, all of them can use the rule $\lambda/Q(c_1, env)$, the spikes c_1 arrives into them, and these neurons become active. With spike c_1 inside, neuron σ_3 can absorb from neuron σ_{in_1} all spikes a , which will be requested by neuron σ_{out} in a later step. In the meantime, neuron σ_4 absorb spike b, one by one, from neuron σ_{in_2}. In the next step, the spike b in neuron σ_4 will absorb from the environment one spike a, and then both the spike a and spike b are requested by neuron σ_{out} together. In this way, through neuron σ_3, the spikes a in σ_{in_1} move to σ_{out} at one time, and through neuron σ_4, the spikes b in σ_{in_2} move to σ_{out} one by one, together with a spike a every time. When the last spike b in σ_{in_2} is requested by neuron σ_4, and then moves to σ_{out} together with a spike a, all the spikes a and b in neurons σ_{in_1}, σ_{in_2}, σ_3 and σ_4 are exhausted. The computation halts, because there is no rule that can be used in the system. At this time, the spikes present in neuron σ_{out} are $a^{x+y}b^y$ (a^x absorbed from neuron σ_3 and $(ab)^y$ absorbed from neuron σ_4). The number of spikes a from the output neuron is $x + y$, which means that the result computed by the system is $x + y$.

Table 1: Spikes in each neuron of Π_{add} at each step during the computation of the addition $5 + 2 = 7$

step	$1/in_1$	$2/in_2$	3	4	$5/out$
0	$\mathbf{a^5}$	$\mathbf{b^2}$	λ	λ	λ
1	a^5	b^2	c_1	c_1	c_1
2	λ	b	a^5c_1	bc_1	c_1
3	λ	b	a^5c_1	abc_1	c_1
4	λ	b	c_1	c_1	a^6bc_1
5	λ	λ	c_1	bc_1	a^6bc_1
6	λ	λ	c_1	abc_1	a^6bc_1
7	λ	λ	c_1	c_1	$\mathbf{a^7}b^2c_1$

With the explanation above, readers can check that, for given $x, y > 0$, system Π_{add} can correctly compute the addition of x and y, which completes the proof. □

As an example, let us consider the addition $5 + 2 = 7$. Table 1 reports the spikes contained in each neuron of Π_{add} at each step during the computation. The input and output spikes are written in bold.

3.2 An SNQ P system for subtraction

We now describe an SN P system with communication on request Π_{sub} used as subtracter, which is shown in Fig. 2. System Π_{sub} is composed of 6 neurons, where two specified neurons are used to introduce the minuend and subtrahend into the system, and one neuron is used for giving the obtained result.

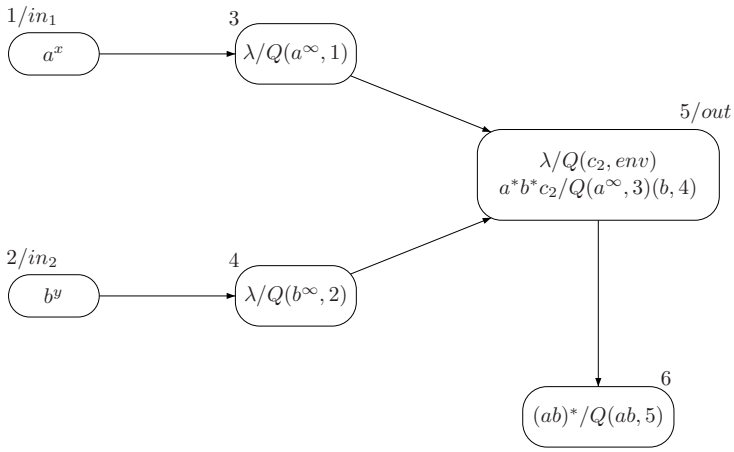


Figure 2: The structure of subtracter Π_{sub}

Theorem 2. *For two arbitrary natural numbers x and y, where $x > y > 0$, SN P system with communication on request Π_{sub} computes the subtraction of x and y.*

Proof: We construct a system Π_{sub} of the form

$$\Pi_{sub} = (\{a, b, c_2\}, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \{a, b\}, \{a\}, \{1, 2\}, 5),$$

where:
$\sigma_1 = (a^x, \emptyset)$,
$\sigma_2 = (b^y, \emptyset)$,
$\sigma_3 = (\lambda, \{\lambda/Q(a^\infty, 1)\})$,
$\sigma_4 = (\lambda, \{\lambda/Q(b^\infty, 2)\})$,
$\sigma_5 = (\lambda, \{\lambda/Q(c_2, env), a^*b^*c_2/Q(a^\infty, 3)(b, 4)\})$,
$\sigma_6 = (\lambda, \{(ab)^*/Q(ab, 5)\})$.

SNQ P system Π_{sub} functions as follows. In the initial configuration of Π_{sub}, all neurons are empty. The minuend x and the subtrahend y are encoded as spikes a^x and b^y, and are provided to neuron σ_{in_1} and neuron σ_{in_2}, respectively. Since neurons σ_3, σ_4 are empty, the spikes a^x will be absorbed by σ_3, and the spikes b^y by σ_4, respectively. In the meantime, neuron σ_5 can use the rule $\lambda/Q(c_2, env)$, and spike c_2 arrives in it. With spike c_2 inside, the neuron σ_5 will absorb one spike a from neuron σ_3 and one spike b from σ_4. In the next step, this pair of spikes a and b will move to the neuron σ_6, and spike c_2 remains in the neuron σ_5. So the absorbability of pairs of spikes a and b continues. When the spikes b in σ_2 get exhausted, the last spike b will be absorbed by σ_4, moves to σ_5 together with one spike a from σ_3, and this last pair of spikes of a and b moves to σ_6 at last. At this time, there is no rule can be used in the system, so the computation halts. During the computation there are y pairs of a and b moves to σ_6, the spikes b get exhausted in neuron σ_4, and there are $x - y$ spikes of a left in neuron σ_3. At the end of computation, the number of spikes a present in the output neuron is $x - y$, which means that the result computed by the system is $x - y$.

Table 2: Spikes in each neuron of Π_{sub} at each step during the computation of the subtraction $5-2=3$

step	$1/in_1$	$2/in_2$	3	4	$5/out$	6
0	$\mathbf{a^5}$	$\mathbf{b^2}$	λ	λ	λ	λ
1	λ	λ	a^5	b^2	c_2	λ
2	λ	λ	λ	b	a^5bc_2	λ
3	λ	λ	λ	λ	a^4bc_2	ab
4	λ	λ	λ	λ	$\mathbf{a^3}c_2$	a^2b^2

With the explanation above, readers can check that, for given $x > y > 0$, system Π_{sub} can correctly compute the subtraction of x and y, which completes the proof. □

As an example let us calculate $5-2=3$. Table 2 reports the spikes that occur in each neuron of Π_{sub} at each step during the computation. Also, the input and output spikes are written in bold.

3.3 An SNQ P System for multiplication

In this section, we present an SN P system with communication on request Π_{mul}, as shown in Fig. 3 with 6 neurons, for implementing the multiplication of two arbitrary natural numbers.

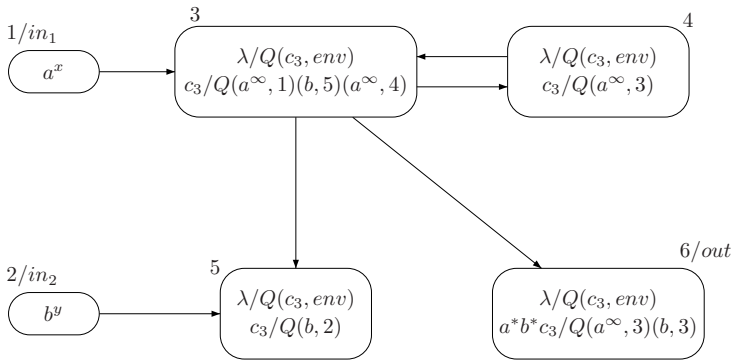


Figure 3: The structure of multiplier Π_{mul}

Theorem 3. *For two arbitrary natural numbers x and y, where $x, y > 0$, SN P system with communication on request Π_{mul} computes the multiplication of x and y.*

Proof: We construct a system Π_{mul} of the form

$$\Pi_{mul} = (\{a, b, c_3\}, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \sigma_6, \{a, b\}, \{a\}, \{1, 2\}, 6),$$

where:
$\sigma_1 = (a^x, \emptyset)$,
$\sigma_2 = (b^y, \emptyset)$,
$\sigma_3 = (\lambda, \{\lambda/Q(c_3, env), c_3/Q(a^\infty, 1)(b, 5)(a^\infty, 4)\})$,
$\sigma_4 = (\lambda, \{\lambda/Q(c_3, env), c_3/Q(a^\infty, 3)\})$,
$\sigma_5 = (\lambda, \{\lambda/Q(c_3, env), c_3/Q(b, 2)\})$,
$\sigma_6 = (\lambda, \{\lambda/Q(c_3, env), a^*b^*c_3/Q(a^\infty, 3)(b, 3)\})$.

SNQ P system Π_{mul} functions as follows. In the initial configuration of Π_{mul}, all neurons are empty. The multiplicand x and the multiplier y are encoded as spikes a^x and b^y, and are provided to neuron σ_{in_1} and neuron σ_{in_2}, respectively. Since neurons σ_3, σ_4, σ_5 and σ_{out} are empty, all of them can use the rule $\lambda/Q(c_3, env)$, the spikes c_3 arrive in them, and these neurons become active.

With spike c_3 inside, neuron σ_5 can absorb from neuron σ_{in_2} one spike b. Now spike b is present in σ_5, so the rule $c_3/Q(a^\infty, 1)(b, 5)(a^\infty, 4)$ in neuron σ_3 can be used: all spikes a in neuron σ_{in_1} and the one spike b in neuron σ_5 are requested by neuron σ_3 (there is no spike a in neuron σ_4, so no spike a requested). After this rule is used, neuron σ_{in_1} becomes empty, and spike c_3 is left in neuron σ_5, which means neuron σ_5 can use again the rule $c_3/Q(b, 2)$. In the next step, the spikes a^x and b are requested by the neuron σ_{out}, and the spike c_3 is left in neuron σ_5. This is the first time the spikes a^x arrive in the output neuron. In the meantime, with spike c_3 present in neuron σ_4 and σ_5, neuron σ_5 absorb the second spike b from neuron σ_{in_2}, neuron σ_4 absorb the spikes a^x from neuron σ_3, Also with the present of spike c_3, these spikes a^x and b will be requested by neuron σ_3 in the next step, and then moves to neuron σ_{out}, which is the second time the spikes a^x arrive in the output neuron.

Table 3: Spikes in each neuron of Π_{mul} at each step during the computation of the subtraction $5 \times 2 = 10$

step	$1/in_1$	$2/in_2$	3	4	5	$6/out$
0	$\mathbf{a^5}$	$\mathbf{b^2}$	λ	λ	λ	λ
1	a^5	b^2	c_3	c_3	c_3	c_3
2	a^5	b	c_3	c_3	bc_3	c_3
3	λ	b	a^5bc_3	c_3	c_3	c_3
4	λ	λ	c_3	a^5c_3	bc_3	a^5bc_3
5	λ	λ	a^5bc_3	c_3	c_3	a^5bc_3
6	λ	λ	c_3	a^5c_3	c_3	$\mathbf{a^{10}}b^2c_3$

With the above explanation, readers can check that, the spikes a^x in neuron σ_{in_1} finally arrive y times at the output neuron, and at the end of the computation, the number of spikes a present in the output neuron is xy, which means that the result computed by the system is xy. So for given $x, y > 0$, system Π_{mul} can correctly compute the product of x and y, which completes the proof. □

For example let us consider $5 \times 2 = 10$. Table 3 reports the spikes that occur in each neuron of Π_{mul} at each step during the computation. Also, the input and output spikes are written in bold.

4 Conclusions and future work

Using the SN P systems with communication on request instead of the traditional SN P systems communicating on command, we have restudied the problem of considering SN P systems as components of an arithmetic logic unit. Specifically speaking, we have proposed three SN P systems with communication on request to implement addition, subtraction and multiplication of two arbitrary natural numbers, respectively. In these systems, natural numbers are introduced into the system as the number of some spike in input neuron, while the result of an arithmetic operation is the number of a specified spike present in output neuron at the end of computation.

First of all, it is an urgent task to propose an SNQ P system to compute the division between two natural numbers, and this one is probably the most difficult to design. In this work, the adder, subtracter and multiplier contain 5 neurons, 6 neurons and 6 neurons, respectively. The number of neurons is less than that is used in [45] (10 neurons, 12 neurons and 26 neurons, respectively), but it has no obvious advantage when compared to that is used in [13] (2 neurons, 2 neurons and 11 neurons, respectively). Therefore, it deserves to be investigated whether the SNQ P systems for arithmetic operations can be simplified by carefully examining the structure, or by using a different construction. Besides, for the further investigation, it is natural to mention this problem: how to construct an SNQ P system to implement arithmetic operations with signed number.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61502063 and 61602188) and Chongqing Social Science Planning Project (2017YBGL142).

Bibliography

[1] Alhazov A., Freund R., Ivanov S., Oswald M., Verlan S. (2017); Extended spiking neural P systems with hole rules and their red-green variants. *Natural Computing*, 2-3, 1–14, 2017.

[2] Cabarle F., Adorna H., Jiang M., Zeng X. (2017); Spiking neural p systems with scheduled synapses. *IEEE Transactions on Nanobioscience*, 16, 792–801, 2017.

[3] Cavaliere M., Ibarra O.H., Păun Gh., Egecioglu O., Ionescu M., Woodworth S. (2009); Asynchronous spiking neural P systems. *Theoretical Computer Science*, 410, 2352–2364, 2009.

[4] Chen H., Freund R., Ionescu M. (2007); On string languages generated by spiking neural P systems, *Fundamenta Informaticae*, 75, 141–162, 2007.

[5] Díaz-Pernil, D., Gutiérrez-Naranjo, M.A. (2018); Semantics of Deductive Databases with Spiking Neural P Systems, *Neurocomputing*, 272, 365-373, 2018

[6] 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.

[7] Gutiérrez-Naranjo, M.A., Leporati, A. (2009); First steps towards a CPU made of spiking neural P systems, *International Journal of Computers Communications & Control*, 4(3), 244–252, 2009.

[8] Ibarra O.H., Păun A., Rodríguez-Patón A. (2009); Sequential SN P systems based on min/-max spike number, *Theoretical Computer Science*, 410, 2982–2991, 2009.

[9] Ionescu M., Păun Gh., Yokomori T. (2006); Spiking neural P systems, *Fundamenta Informaticae*, 71, 279–308, 2006.

[10] Ishdorj T.-O., Leporati A., Pan L., Zeng X., Zhang X. (2010); Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources, *Theoretical Computer Science*, 411, 2345–2358, 2010.

[11] Krithivasan K., Metta V.P., Garg D. (2011); On string languages generated by spiking neural P systems with anti-spikes. *International Journal of Foundations of Computer Science*, 22, 15–27, 2011.

[12] Liu X., Li Z., Liu J., Liu L., Zeng X. (2015); Implementation of arithmetic operations with time-free spiking neural P systems, *IEEE Transactions on Nanobioscience*, 14, 617–624, 2015.

[13] Martín-Vide C., Păun Gh., Pazos J., Rodríguez-Patón A. (2003); Tissue P systems, *Theoretical Computer Science*, 296, 295–326, 2003.

[14] Metta V.P., Raghuraman S., Krithivasan K. (2014); Small universal simple spiking neural P systems with cooperating rules as function computing devices, *Lecture Notes in Computer Science*, 8961, 300–313, 2014.

[15] Neary T. (2009); A boundary between universality and non-universality in extended spiking neural P systems, *Lecture Notes in Computer Science*, 6031, 475–487, 2009.

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

[17] Pan L., Păun Gh., Pérez-Jiménez M.J. (2011); Spiking neural P systems with neuron division and budding, *Science China Information Sciences*, 54, 1596–1607, 2011.

[18] Pan L., Păun Gh., Zhang G., Neri F. (2017); Spiking neural P systems with communication on request, *International Journal of Neural Systems*, 27, 1750042, 2017.

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

[20] Pan L., Zeng X. (2010); A note on small universal spiking neural P systems, *Lecture Notes in Computer Science*, 5957, 436–447, 2010.

[21] Pan L., Zeng X., Zhang X., Jiang Y. (2012); Spiking neural P systems with weighted synapses, *Neural Processing Letters*, 35, 13–27, 2012.

[22] Păun Gh. (2000); Computing with membranes, *Journal of Computer and System Sciences*, 61, 108–143, 2000.

[23] Păun Gh. (2002); *Membrane Computing: An Introduction*, Springer, 2002.

[24] Păun Gh. (2016); Membrane Computing and Economics: A General View, *International Journal of Computers Communnication & Control*, 11, 105–112, 2016.

[25] Păun Gh., Păun A. (2007); Small universal spiking neural P systems, *Biosystems*, 90, 48–60, 2007.

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

[27] Song T., Gong F., Liu X., Zhao Y., Zhang X. (2016); Spiking neural P systems with white hole neurons, *IEEE Transactions on Nanobioscience*, 15, 666-673, 2016.

[28] Song T., Jiang Y., Shi X., Zeng X. (2013); Small universal spiking neural P systems with anti-spikes, *Journal of Computational and Theoretical Nanoscience*, 10, 999–1006, 2013.

[29] Song T., Pan L. (2014); A small universal spiking neural P systems with cooperating rules, *Romanian Journal of Information Science and Technology*, 17, 177–189, 2014.

[30] Song T., Pan L. (2016); Spiking neural P systems with request rules, *Neurocomputing*, 193, 193–200, 2016.

[31] Song T., Pan L., Jiang K., Song B., Chen W. (2013); Normal forms for some classes of sequential spiking neural P systems, *IEEE Transactions on Nanobioscience*, 12, 255–264, 2013.

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

[33] Song T., Xu J., Pan L. (2015); On the universality and non-universality of spiking neural P systems with rules on synapses, *IEEE Transactions on Nanobioscience*, 14, 960–966, 2015.

[34] Song T., Zheng P., Wong M.L., Wang X. (2016); Design of logic gates using spiking neural P systems with homogeneous neurons and astrocytes-like control, *Information Sciences*, 372, 380–391, 2016.

[35] Su Y., Wu T., Xu F., Păun A. (2017); Spiking neural p systems with rules on synapses working in sum spikes consumption strategy, *Fundamenta Informaticae*, 156, 187–208, 2017.

[36] Wang J., Hoogeboom H.J., Pan L., Păun Gh., Pérez-Jiménez M.J. (2014); Spiking neural P systems with weights, *Neural Computation*, 22, 2615–2646, 2014.

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

[38] Wang J., Shi P., Peng H., Pérez-Jiménez M.J., Wang T. (2013); Weighted fuzzy spiking neural P systems, *IEEE Transactions on Fuzzy Systems*, 21, 209–220, 2013.

[39] Wang T., Zhang G., Zhao J., He Z., Wang J., Pérez-Jiménez M.J. (2015); Fault diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems, *IEEE Transactions on Power Systems*, 30, 1182–1194, 2015.

[40] Wang X., Song T., Gong F., Zheng P. (2016); On the computational power of spiking neural P systems with self-organization, *Scientific Reports*, 6: 27624, 2016.

[41] Wu T., Păun A., Zhang Z., Pan L. (2017); Spiking neural P systems with polarizations, *IEEE Transactions on Neural Networks and Learning Systems*, 1–12, 2017.

[42] Wu T., Zhang Z., Pan L. (2016); On languages generated by cell-like spiking neural P systems, *IEEE Transactions on Nanobioscience*, 15, 455–467, 2016.

[43] Wu T., Zhang Z., Păun Gh., Pan L. (2016); Cell-like spiking neural P systems, *Theoretical Computer Science*, 623, 180–189, 2016.

[44] Zeng X., Pan L., Pérez-Jiménez M.J. (2014); Small universal simple spiking neural P systems with weights, *Science China Information Sciences*, 57, 1–11, 2014.

[45] Zeng X., Song T., Zhang X., Pan L. (2012); Performing four basic arithmetic operations with spiking neural P systems, *IEEE Transactions on Nanobioscience*, 11, 366–374, 2012.

[46] Zeng X., Xu L., Liu X. (2014); On string languages generated by spiking neural P systems with weights, *Information Sciences*, 278, 423–433, 2014.

[47] Zeng X., Zhang X., Pan L. (2009); Homogenous spiking neural P systems, *Fundamenta Informaticae*, 97, 275–294, 2009.

[48] Zhang G. (2017); *Real-life applications with membrane computing*, Springer, 2017.

[49] 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, 1440006, 2014.

[50] Zhang X., Liu Y., Luo B., Pan L. (2014); Computational power of tissue P systems for generating control languages, *Information Sciences*, 278, 285–297, 2014.

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

[52] Zhang X., Wang B., Pan L. (2014); Spiking neural P systems with a generalized use of rules, *Neural Computation*, 26, 2925–2943, 2014.

[53] Zhang X., Zeng X., Luo B., Pan L. (2014); On some classes of sequential spiking neural P systems, *Neural Computation*, 26, 974–997, 2014.

[54] Zhang X., Zeng X., Pan L. (2008); Smaller universal spiking neural P systems, *Fundamenta Informaticae*, 87, 117–136, 2008.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 365-382, June 2018.

Identifying Essential Proteins in Dynamic PPI Network with Improved FOA

X. Lei, S. Wang, L. Pan

Xiujuan Lei, Siguo Wang
School of Computer Science
Shaanxi Normal University
Xian 710119,Shaanxi, China
xjlei@snnu.edu.cn, wangsiguo@snnu.edu.cn

Linqiang Pan*
1. Key Laboratory of Image Information Processing and
Intelligent Control of Education Ministry of China
School of Automation
Huazhong University of Science and Technology
Wuhan 430074, Hubei, China
2. School of Electric and Information Engineering
Zhengzhou University of Light Industry
Zhengzhou 450002, Henan, China
*Corresponding author: lqpan@mail.hust.edu.cn

Abstract: Identification of essential proteins plays an important role for understanding the cellular life activity and development in postgenomic era. Identification of essential proteins from the protein-protein interaction (PPI) networks has become a hot topic in recent years. In this work, fruit fly optimization algorithm (FOA) is extended for identifying essential proteins, the extended algorithm is called EPFOA, which merges FOA with topological properties and biological information for essential proteins identification. The algorithm EPFOA has the advantage of identifying multiple essential proteins simultaneously rather than completely relying on ranking score identification individually. The performance of EPFOA is analyzed on dynamic PPI networks, which are constructed by combining the gene expression data. The experimental results demonstrate that EPFOA is more efficient in detecting essential proteins than the state-of-the-art essential proteins detection methods.
Keywords: essential proteins, protein-protein interaction (PPI), dynamic PPI networks, subcellular localization data, fruit fly optimization algorithm (FOA).

1 Introduction

Protein plays an important role in the cellular life activity, and essential proteins are critical for the growth and development of organisms under a variety of conditions [27]. The absence of a single essential protein is sufficient to cause lethality or infertility [50]. Some recent results suggest that a comprehensive analysis of essential proteins can provide a deeper understanding of the relationship between mutations and human diseases, revealing the general principles of human diseases [12, 15, 59]. Therefore, the identification of essential proteins is closely related to disease prediction and drug design [53].

With the development of high-throughput technologies, various biological data are available, e.g., yeast-two-hybrid, tandem affinity purification, and mass spectrometry. In [2], a greedy algorithm is proposed to optimize the detection of protein communities.

Existing methods for identifying essential proteins can be roughly divided into two types. The first type includes the biological experiment-based methods, e.g., gene knockouts [11], RNA interference [7], and conditional knockouts [35], which are expensive and time-consuming. The

other type includes the topology-based centrality method, e.g., Degree Centrality (DC) [14], Betweenness Centrality (BC) [24], Closeness Centrality (CC) [52], Subgraph Centrality (SC) [9], Eigenvector Centrality (EC) [3], Information Centrality (IC) [40], Neighborhood Centrality (NC) [20], and Local Average Connectivity-based method (LAC) [19]. By defining and computing the topologically potential value of each protein, these methods can obtain a precise ranking score reflecting the importance of proteins in the protein-protein interaction (PPI) network [18]. Some centrality analysis tools and RNA detection tools [54] have been developed. For example, CytoNCA [43], a Cytoscape plugin, integrated eight centrality measures, i.e., DC, BC, CC, EC, IC, SC, NC and LAC. Obviously, the topology-based centrality methods can improve the efficiency with less cost. However, these centrality methods also have their own shortcomings. It is well known that the performance of topology-based methods is closely related to the quality of the PPI networks, but there are many false positive and false negative in the PPI networks.

In order to deal with the drawbacks of these methods, some new methods are proposed to predict essential proteins by integrating their topological properties with their biological properties. Considering the interaction data and Gene Ontology (GO) annotations, Hsing et al. introduced a method to predict highly-interacting proteins [13]. Later, a new prediction method called PeC was proposed by Li et al. [21], and another method called WDC was proposed by Tang et al. [41], which integrate network topology with gene expression profiles. Afterwards, Tang introduced a new method to identify essential proteins in which topological features of PPI network is combined with subcellular localization information [42]. Next, a new centrality measure is proposed by Ren et al. to discover essential proteins, named harmonic centricity, which merges subgraph centrality with protein complexes to discover essential proteins [34]. Recently, a new prediction method, named UDoNC, that combine the domain features of proteins with their topological properties in PPI networks, was proposed by Peng et al. [30]. Some machine learning methods, e.g., Support Vector Machine, Naive Bayes, Bayes Network, and NBTree, were also adopted to predict essential proteins by using different features. For example, the random forest was adopted to predict essential proteins by Qin et al. [32]. These methods that combine the network topological features with biological data is capable of improving the accuracy and efficiency of prediction significantly. These existing methods regard the PPI networks as static networks that ignore the time-course of the networks. The real PPI networks in cell keeps changing over different stages of the cell cycle [31], and they can be classified into stable or transient PPI networks [46], which are usually described as dynamic PPI networks (DPIN). Thus it is important to construct dynamic PPI networks to investigate the temporal properties of individual proteins and protein interactions. Based on dynamic network topology and complex information, Luo and Kuang proposed a new method to predict essential proteins [22]. The results show that the identification of essential proteins in dynamic networks is more conducive than in static networks.

Fruit fly optimization algorithm (FOA) is a novel swarm intelligent algorithm that mimics the foraging behavior of fruit flies for global optimization [25]. FOA is easy to be understood and implemented, which has few parameters to be adjusted. Due to its simplicity and efficiency, FOA showed great success in solving some real-world complex problems like multidimensional knapsack problem [48]. Here, FOA will be used to find the essential proteins.

In this work, we present a new algorithm, called EPFOA, in which FOA is merged with topological properties and biological information for essential proteins identification. To the best of my knowledge, most of the methods of essential proteins identification focus on static PPI networks and ignore the intrinsic features of organisms.

In our method, we first integrate gene expression data with static PPI network to construct the dynamic network model. Then a new topological centrality method that combines GO annotation and edge aggregation coefficient (ECC) is proposed to measure the topological char-

acteristic of PPI networks with modular local average connectivity (LAC) in dynamic networks. Furthermore, the distribution of proteins in each compartment according to subcellular localization data is obtained, and the role of components in identifying essential proteins is analyzed.

Finally, EPFOA is designed to identify essential proteins. To assess the performance of our method, EPFOA is compared with some existing methods including DC, EC, IC, SC, NC, LAC, PeC and UDoNC, and the experimental results indicate that our method significantly outperforms with the existing methods.

2 Method

2.1 Fruit fly optimization algorithm

Fruit fly optimization algorithm (FOA) is a novel method for global optimization, which is inspired by the foraging behavior of fruit flies. In sensory perception, the fruit fly is superior to the other species, especially in olfactory and vision. The olfactory organs of fruit flies can collect all kinds of scents floating in the air, even smell the food source from 40 kilometers away. After the fruit fly gets close to the food, it can also use the sensitive vision to find food and the companyĄŹs flocking location, and fly to the direction [25]. The procedure of FOA is presented in pseudo code as follows.

Step 1. Randomly initialize the location of the fruit flies (X_{axis}, Y_{axis}).

Step 2. Give the random direction and distance for the search of food using osphresis by an individual fruit fly.

$$\begin{cases} X_i = X_{axis} + RandomValue \\ Y_i = Y_{axis} + RandomValue \end{cases} \tag{1}$$

Step 3. The distance $(Dist_i)$ to the origin is estimated, then the smell concentration judgment value (S_i) is calculated, which is the reciprocal of the distance.

$$\begin{cases} Dist_i = \sqrt{x^2 + y^2} \\ S_i = \frac{1}{Dist_i} \end{cases} \tag{2}$$

Step 4. Substitute smell concentration judgment value (S_i) into smell concentration judgment function (or called fitness function) to find the smell concentration $(Smell_i)$ of the individual location of the fruit fly.

$$Smell_i = Function(S_i) \tag{3}$$

Step 5. Find the individual with the maximal smell concentration among the fruit fly swarm according to the smell concentration value.

$$[bestSmell\ bestInedx] = max(Smell) \tag{4}$$

Step 6. Maintain the best smell concentration value x and y, where the fruit fly swarm will use vision to fly towards that location.

$$\begin{cases} Smell = bestSmell \\ X_{axis} = X(bestIndex) \\ Y_{axis} = Y(bestIndex) \end{cases} \tag{5}$$

Step 7. Repeat steps 2-5 until the smell concentration is superior to the previous smell concentration; otherwise, go to step 6.

2.2 Dynamic PPI network model construction

Gene expression data is valuable for revealing the dynamic properties of proteins and PPI. We integrate gene expression data with high-throughput PPI data to construct a dynamic PPI network. Note that protein does not always become active at a cell cycle, a protein is active at the highest gene expression level. In order to mark the active time of each gene, the active threshold of each gene should be calculated, and the gene is active if its expression value is greater than the active threshold. The calculation of active threshold is proceeded on the 3-sigma model [45].

$$AT(p) = \mu(p) + 3 \times \sigma(p) \times (1 - \frac{1}{1 + \sigma(p)^2}), \tag{6}$$

where $\mu(i)$ is the mean gene expression value of protein i and $\sigma(i)$ is the algorithm standard deviation of the expression values over time 1 to T for protein i. Since the gene expression data has three cycles and each cycle has 12 times tamps, the final gene expression at each time point is the average of the three cycles, which is defined as follows [16]:

$$FT(i) = \frac{T(i) + T(i+12) + T(i+24)}{3}, (i \in [1, 12]), \tag{7}$$

where $T(i)$ denotes the gene expression value at time point i. At a certain times tamp, if both proteins are active with an interaction, the interaction of the two proteins is also active. Eventually the entire PPI network was divided into 12 sub-networks, the dynamic PPI network was constructed.

2.3 Topological characteristics of dynamic networks

A PPI network is not only an important biological network but also a typical complex network, which meets the topological characteristics of complex network, such as small-world [49], scale-free [51], and modularity [10]. In this part, the role of the topological characteristics in the process of essential proteins identification is investigated, and a new topological centrality method based on the ECC and GO annotation is proposed. Furthermore, the modularity of the network that applied LAC is also considered.

Dynamic network topology centrality strategy

A PPI network can usually be expressed as an undirected graph $G = (V, E)$, where the set of vertices V represents protein, and E represents all of interactions between pairs of proteins. In order to assess the centrality of dynamic network topology, we introduce the GO annotation (since the ECC cannot fully reflect the characteristics). GO annotation provides valuable information and a convenient method to study the gene function similarity, some researches have shown that the adoption of GO semantic similarity term can improve the prediction accuracy of protein complexes gene and disease [36, 56, 57].

Weighting the networks via ECC

In order to measure the tightness of the two nodes, we use the ECC [41], which is defined as follows:

$$ECC(u, v) = \frac{|N_u \cap N_v| + 1}{min\{d_u, d_v\}}, \tag{8}$$

where N_u (or N_v) refers to the set of neighbours of node u (or v) in PPI networks, $|N_u \cap N_v|$ is the number of common neighbor nodes of u and v, which is consistent with the number of triangles which edge (u, v) belongs to d_u (or d_v) indicates the degrees of node u (or v).

Weighting the networks using the Gene Ontology

The GO information consists of three sub-ontologies: Biological Process (BP), Cellular Component (CC) and Molecular function (MF) [6]. In order to measure the semantic similarity between the GO terms to protein annotations in an interaction network, we applied the method developed by Wang et al. [47]:

$$GO\_sim(u,v) = \frac{\sum_{t \in T_u \bigcap T_v}(S_u(t) + S_v(t))}{\sum_{t \in T_u} S_u(t) + \sum_{t \in T_v} S_v(t)}, \tag{9}$$

where where T_u and T_v are the annotations of protein u and v; $S_u(t)$ is the S-value of GO term t related to term u and $S_v(t)$ is the S-value of GO term t related to term v.

Generating new weighted networks

Based on the definition of the ECC and gene functional similarity, a new centrality measure, named EG, is proposed. For a protein u, the essentiality $EG(u)$ is defined as the probability between the ECC and GO information:

$$EG(u) = \sum_{v \in N_u} ECC(u,v) \times GO\_sim(u,v), \tag{10}$$

where $N(u)$ denotes the set of all neighbors of node u. When computing dynamic $EG(u)$, we should consider the number of times that each node appears in a dynamic PPI network, since some nodes are not included in all the time networks. Dynamic $EG(u)$ can be defined as the follows:

$$D_{EG}(u) = \frac{\sum_{i=1}^{N} EG^i(u)}{tim(u)}, \tag{11}$$

where N is the number of temporal networks in the dynamic network, $EG^i(u)$ the EG of node u in the ith time point, $tim(u)$ the number of time networks that contain node u. If node u does not appear at time point i, $EG^i(u)$ is equal to zero.

Dynamic local average connectivity

The LAC of a node indicates its closeness [49], and the LAC of a node v is defined as:

$$LAC(u) = \frac{\sum_{v \in N_u} deg^{C_u}(v)}{|N_u|}, \tag{12}$$

where N_u is the neighbors of node v, $|N_u|$ the number of nodes in N_u, and C_u the subgraph induced by N_u. For a node u in C_v, its local connectivity in C_u is represented as $deg^{(C_u)}(v)$. Similar to $D_{EG}(u)$, we define Dynamic LAC as follows [30]:

$$D_{LAC}(u) = \frac{\sum_{i=1}^{N} LAC^i(u)}{tim(u)}, \tag{13}$$

where N is the number of temporal networks in the dynamic network, $LAC^i(u)$ the LAC of node v in the ith time point, and $tim(u)$ the number of time networks that contain node u.

2.4 Subcellular localization score

Subcellular location is divided into different compartments, different compartments play different roles in cell activities. In order to understand the relationship between subcellular localization and essential proteins, we analyze the number of essential proteins in each subcellular location and propose a method to evaluate subcellular localization in previous research. Assume

that in the Nucleus, the wider the distribution of the proteins is, the greater the possibility of essential protein becomes [17].

Let C_{max} denote the protein with the largest number of times appearing in subcellular localization of the nucleus, $|u|$ represents the number of times of the protein u appearing in the nucleus. The importance of protein u, denoted as $NSL(u)$, is calculated by the ratio of its size to the largest size of the nucleus. The value of $NSL(u)$ is in the range of $(0, 1]$.

$$NSL(u) = \frac{|u|}{|C_{max}|} \tag{14}$$

2.5 EPFOA algorithm

In order to make up for the shortcomings of traditional identification of essential proteins one by one, we propose the algorithm EPFOA. The algorithm can identify p candidate essential proteins simultaneously, which greatly improves the recognition efficiency. In what follows, we introduce the algorithm EPFOA. First, initialize the position of fruit fly and set the rules of location updating. Then find p candidate essential proteins according to the characteristic of FOA. Finally, the identified p essential proteins are compared with known essential proteins to verify the number of essential proteins identified correctly.

The initialization and update of the location of fruit flies

The initialization and location update rules of fruit fly play an important role in the performance of EPFOA. The position of the fruit fly is encoded as an integer set of p-dimensional set $X_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, $(i = 1, 2, \ldots, n)$ which denotes a candidate essential protein set. Each element x_{ij} $(x_{ij} \leq |N|, j = 1, 2, \ldots, p)$ in X_i is the sequence number of a protein. First we randomly selected p proteins to initialize a fruit fly position X_i. Then we compare the selected p proteins with the known essential proteins and keep the proteins that are successfully matched. After that the remaining positions that represent proteins are updated. In order to speed up the convergence of the proposed EPFOA algorithm, we sort all the proteins based on degree except selected p proteins. A random value is assigned to the individual that is not essential protein in the X_i and update the position in a sequence that is ranked by degree.

Encoding and decoding of EPFOA

The framework of EPFOA is shown in Fig.2. We set every fruit fly as essential protein candidate set, and the location of fruit fly is the serial number of the candidate proteins. For the purpose of evaluating the topological characteristics of the network comprehensively, we combine LAC that represents the network modularity with the new network centrality. Thus, when a fruit fly is in a certain position, we suppose its smell concentration judgment value $S(i)$ can be calculated as following equation:

$$S(i) = \sum_{j=1}^{p} (D_{LAC}(\mu_j) + D_{EG}(\mu_j)), \tag{15}$$

where $D_{LAC}(u_j)$ denotes the dynamic local average connectivity of the jth protein among the p candidate essential proteins and $D_{EG}(u_j)$ denotes dynamic network topology centrality of the jth protein among the p candidate essential proteins.

The topological characteristics and biological data are both indispensable in the process of identifying essential proteins and subcellular localization data plays an important role in essential proteins identification. We set the following smell concentration judgement function to measure the possibility of essential proteins represented by a fruit fly individual:

$$Fit(i) = \alpha \times S(i) + (1 - \alpha) \times \sum_{j=1}^{p} NSL(\mu_j), \tag{16}$$

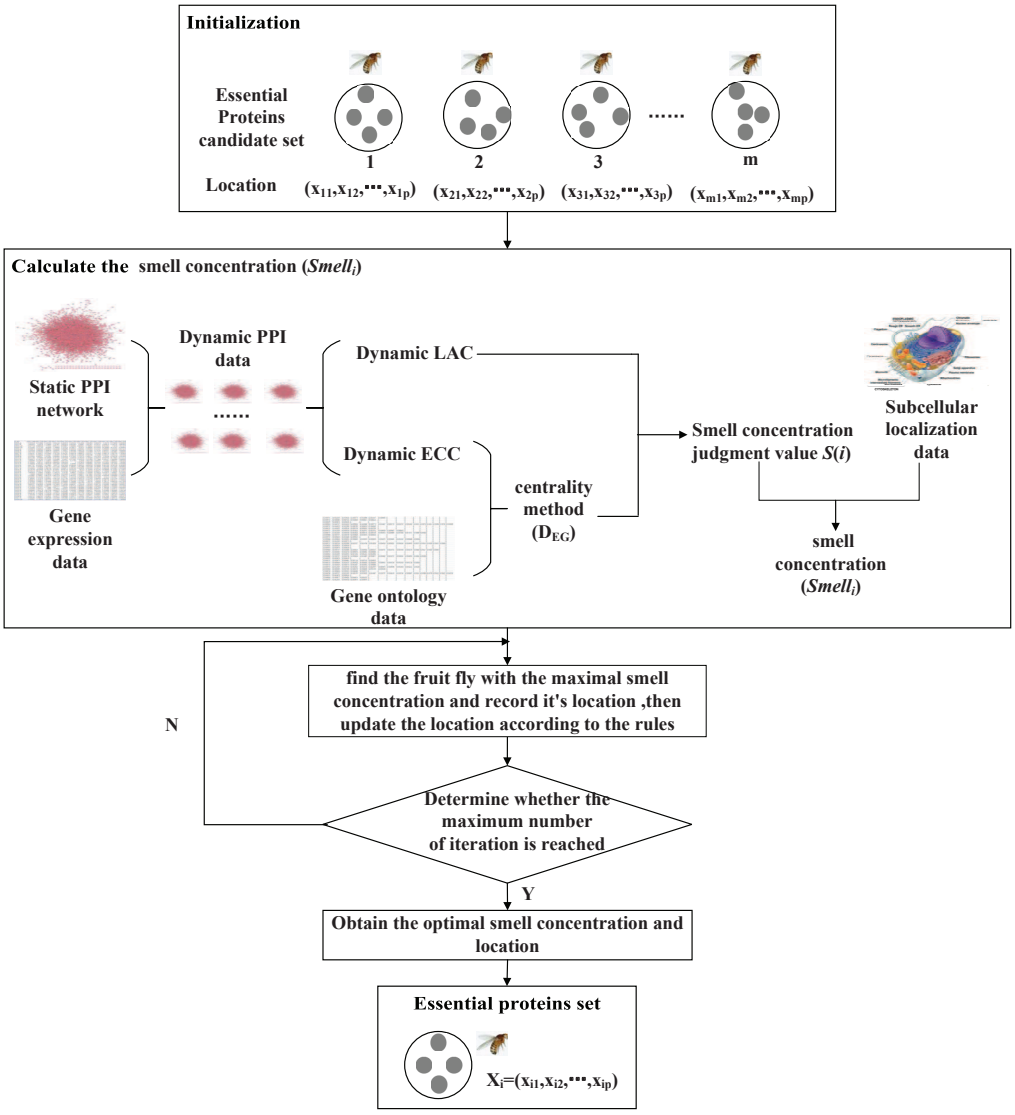


Figure 1: The framework of the algorithm EPFOA.

where $NSL(u_j)$ denotes subcellular localization score of the jth protein among the p candidate essential proteins and $\alpha \in [0, 1]$, α is a parameter that regulates the proportion of the network topology and biological information in the process of identifying essential proteins. If $\alpha = 0$, only subcellular location information works; else if $\alpha = 1$, only network topology works.

Pseudo code of EPFOA

The process of EPFOA can be divided into two steps. The first step calculates the topological and biological characteristics of protein nodes. The second step applies the process of FOA algorithm to seek the optimal to find the essential proteins. The pseudo code of EPFOA is shown in Algorithm 1.

Algorithm 1 The pseudo code of EPFOA

Ensure: $G = (V, E)$ (the PPI network), Gene expression data, Gene Ontology GO, Subcellular location data.
Require: Essential protein set.
Construct the dynamic PPI network
for each interacting protein pair (a, b) in PPI **do**
    Calculate ECC /*The closeness of the two nodes*/
    Calculate GO /*The functional similarity of the two nodes based on GO annotation*/
end for
for each node in G **do**
    Update the centrality $D_{EG}(u)$
    Calculate $D_{LAC}(u)$
    Calculate subcellular location score $NSL(u)$
end for
for fruit fly i **do**
    Initialize location $x(i)$ and its best location $b\_x(i)$
    Calculate the smell concentration smell(i)= Fit($S(i)$)
end for
for m in $[1, maxiter]$ **do**
    for fruit fly i **do**
        Update location $X(i) = X(i) + random$
        if smell(i)<Fit(S(i)) **then**
            $b\_x(i) = X(i)$
        end if
    end for
end for

3 Results and discussion

In this section, we first introduce the experimental data. Then we analyze the parameter A towards the performance of EPFOA. Next, in order to evaluate the performance of EPFOA more synthetically, we not only compare EPFOA with some topology-based centrality methods (DC, EC, IC, SC, NC, LAC) but also with some methods that integrate their topological properties with their biological properties (PeC and UDoNC). In order to assess the essentiality of proteins in PPI networks, these methods are ranked in descending order based on their ranking scores including eight existing centrality methods (DC, EC, IC, SC, NC, LAC, PeC and UDoNC). After

that, top 1%,5%, 10%, 15%, 20% and 25% of the ranked proteins are selected as candidates for essential proteins. In this paper, the size of the set of essential proteins candidate is 1274. Taking into account of the random optimization process of FOA, we conduct ten experiments and then use the average of ten experiments as the final result to to analyze the parameter towards the performance of EPFOA. The ten experiments are listed in the attachment 1. To further evaluate the EPFOA performance, we randomly choose a candidate essential proteins from ten experiments to compare with other methods. The performance is presented in the form of histograms of the number of essential proteins predicted by each algorithm and also use six statistical measures to evaluate them. And precision-recall curves and jackknife curves are also used to evaluate the performance of the proposed EPFOA method and the other eight methods. Finally,we analysis the modularity of identified essential proteins.

3.1 Experimental data

To evaluate the performance of our proposed algorithm EPFOA, we adopt PPI networks of S.cerevisiae which has been well characterized by knockout experiments and widely used in the evaluation of methods for essential proteins discovery. The PPI data of S.cerevisiae was downloaded from DIP database [58], which contains 5093 proteins and 24743 interactions after removing the repeated interactions and the self-interactions. The known essential proteins data of S.cerevisiae contains 1285 essential proteins among which 1167 essential proteins present in the DIP network, which are collected from four databases: MIPS [23], SGD [4], DEG [55], and SGDP (http://www-sequence.stanford.edu/group). The gene expression data of S.cerevisiae are downloaded from GEO database [44] that contains 7074 gene expression products. The Gene ontology annotation data of S.cerevisiae is obtained from GO Consortium [5]. Subcellular localization dataset of S. cerevisiae is downloaded from knowledge channel of COMPARTMENTS database [1], which includes 5095 yeast proteins and 206,831 subcellular localization records.

3.2 The effect of parameter α on performance

In our proposed algorithm EPFOA, evaluation function of proteins is changed with different values of α. To study the effect of parameter α on performance of EPFOA, we evaluate the prediction accuracy by setting different values of α, ranging from 0 to 1. The detailed results are listed in Table 1. As shown in Table 1, the results are similar with α, ranging from 0.4 to 1. Synthetically, we consider the optimal values to be $\alpha = 0.1$.

Table 1: Effect of parameter α on the performance of EPFOA

α / TOP	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
1%	37	**45**	42	40	40	40	39	40	40	39	39
5%	181	183	183	184	184	**185**	177	178	177	176	178
10%	**350**	341	334	317	304	295	287	289	288	284	284
15%	444	**451**	445	429	423	417	415	409	409	400	415
20%	**563**	542	537	532	531	528	529	530	530	529	544
25%	610	**628**	624	621	618	612	616	617	617	616	616

3.3 Comparison with other prediction measures

In order to demonstrate the advantage of our proposed EPFOA, we compare EPFOA with eight existing methods including DC, EC, IC, SC, NC, LAC, PeC and UDoNC. The essential proteins candidate population size p is set to 1274 (5093*25%=1274). The top 1, 5, 10, 15, 20 and 25% proteins are selected as candidate essential proteins, respectively. Then the prediction results are compared with the known essential proteins, and the experimental results are shown in Fig. 3. It can be observed that the percentage of essential proteins predicted by EPFOA is consistently higher than that achieved by the eight compared methods. Taking top 1% (top 51) predicted essential proteins as an example, 46 essential proteins are correctly identified by EPFOA while SC and EC have correctly predicted 24 essential proteins.

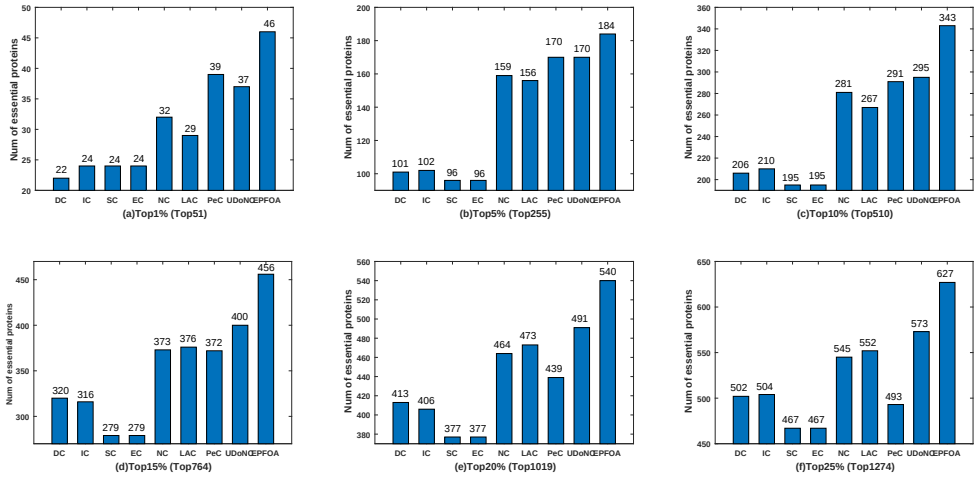


Figure 2: EPFOA compared with several existing methods.(a) Top 1% (Top 51), (b) Top 5% (Top 255), (c) Top 10% (Top 510), (d) Top 15% (Top 764), (e) Top 20% (Top 1019), (f) Top 25% (Top 1274).

3.4 Validation using six statistical measures

In order to evaluate the performance of EPFOA, we compare EPFOA with the other methods using six statistical measures: sensitivity (SN), specificity (SP), positive predictive value (PPV), negative predictive value (NPV), F-measure, and accuracy (ACC). Each statistical measure is defined as follows:

$$SN = \frac{TP}{TP + FN}, \tag{17}$$

$$SP = \frac{TN}{TN + FP}, \tag{18}$$

$$PPV = \frac{TP}{TP + FP}, \tag{19}$$

$$NPV = \frac{TN}{TN + FN}, \tag{20}$$

$$F - measure = \frac{2 \times SN \times PPV}{SN + PPV}, \tag{21}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}, \tag{22}$$

where TP is the number of essential proteins correctly identified as essential proteins, FP is the number of nonessential proteins mistakenly identified as essential proteins, TN is the number of nonessential proteins correctly identified as nonessential proteins, and FN is the number of essential proteins mistakenly identified as nonessential proteins. The comparison results between EPFOA and the other predicted essential proteins methods by six statistical measures performed on DIP are shown in Table 2. Obviously, we can see that EPFOA significantly outperforms all the compared methods.

Table 2: Comparison of EPFOA and the other methods in terms of SN, SP, PPV, NPV, F-measure, and ACC on the PPI networks.

Method	SN	SP	PPV	NPV	F-measure	ACC
DC	0.4302	0.8033	0.394	0.8258	0.4113	0.7178
EC	0.4002	0.7944	0.3666	0.8167	0.3826	0.704
IC	0.4319	0.8038	0.3956	0.8263	0.4129	0.7186
SC	0.4002	0.7944	0.3666	0.8167	0.3826	0.704
NC	0.467	0.8143	0.4278	0.8371	0.4465	0.7347
LAC	0.473	0.8161	0.4333	0.8389	0.4523	0.7374
PeC	0.4225	0.801	0.387	0.8235	0.4039	0.7143
UDoNC	0.491	0.8214	0.4498	0.8444	0.4695	0.7457
EPFOA	**0.5373**	**0.8352**	**0.4922**	**0.8586**	**0.5137**	**0.7669**

3.5 Comparison of the experimental results based on precision-recall curves

To further validate the performance of EPFOA, we study the Precision-Recall (PR) of EPFOA on the PPI networks and compare with the other methods. The precision and recall of the top n ranked proteins are defined as follow:

$$Precision(n) = \frac{TP(n)}{TP(n) + FP(n)}, \tag{23}$$

$$Recall(n) = \frac{TP(n)}{P}, \tag{24}$$

where $TP(n)$ is the number of true predicted essential proteins among the top n ranked proteins, $FP(n)$ is the number of false predicted essential proteins among the top n ranked proteins, P is the total number of essential proteins under consideration. Fig. 4 shows the PR curves of EPFOA and the other eight methods on the PPI networks. Obviously, EPFOA obtains the best performance, which demonstrates that the algorithm EPFOA works well in identifying essential proteins.

3.6 Validation using jackknife curves

A more general comparison between the proposed algorithm EPFOA and the eight previously proposed methods is tested by using a jackknife curves. The experimental results validated by Jackknife curves are shown in Fig. 5 the X-axis represents the proteins ranked in descending order from left to right according to the values computed using the corresponding methods, and the Y-axis represents the number of true essential proteins among the top n proteins, where n is the number along the X-axis. The area under the curve is always used to measure the generality of a method. As shown in Fig. 5, EPFOA clearly performs better than the other methods.

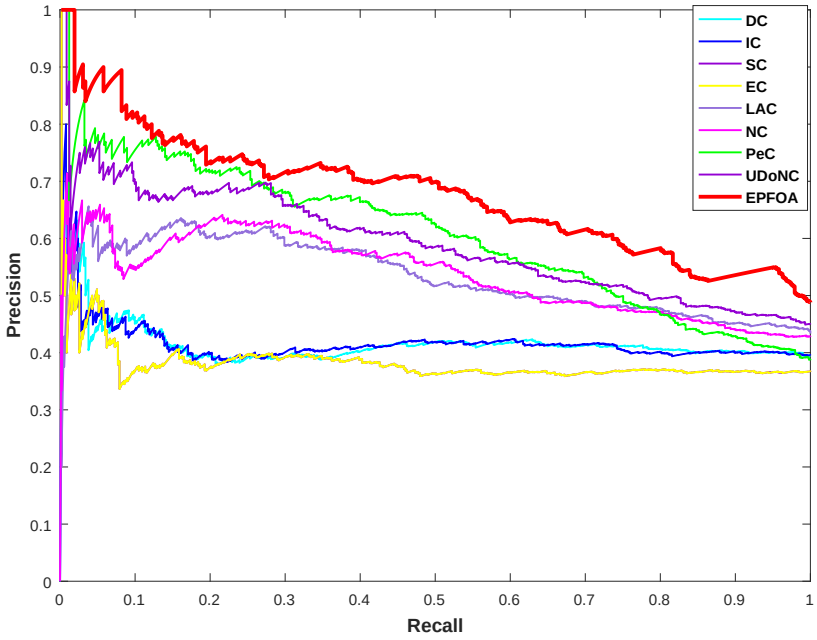


Figure 3: The PR curves of GSP and that of other methods.

3.7 The modularity of essential proteins predicted by EPFOA

Proteins usually perform tasks in biological system with protein complexes or functional modules and rarely act alone. Therefore, protein modularity may be an appropriate measurement to evaluate the significance of essential proteins identified by EPFOA. In order to examine the modularity of essential proteins identified by EPFOA, we compare EPFOA with DC that fully depend on network topology and PeC that combine network topology with biological information. We show the top 1% identified essential proteins of each method. As illustrated in Fig. 5, the number of essential proteins identified by EPFOA is higher than DC and PeC obviously. It also can be seen in Fig. 5, it is worthy to note that the essential proteins identified by EPFOA show more significant modularity than DC and PeC. It indicates that EPFOA is effective in identifying essential proteins.

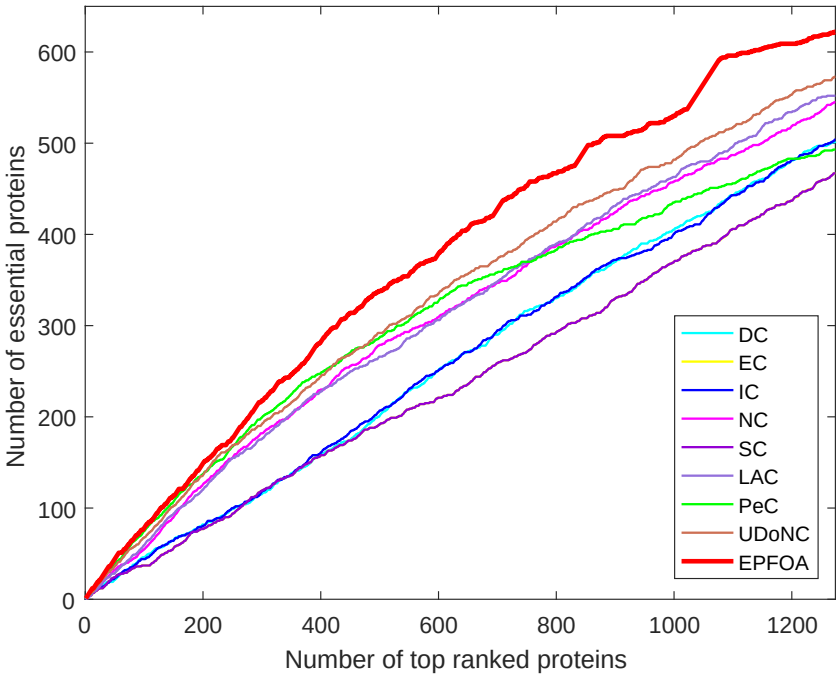


Figure 4: The jackknife curves of GSP and the other nine methods.

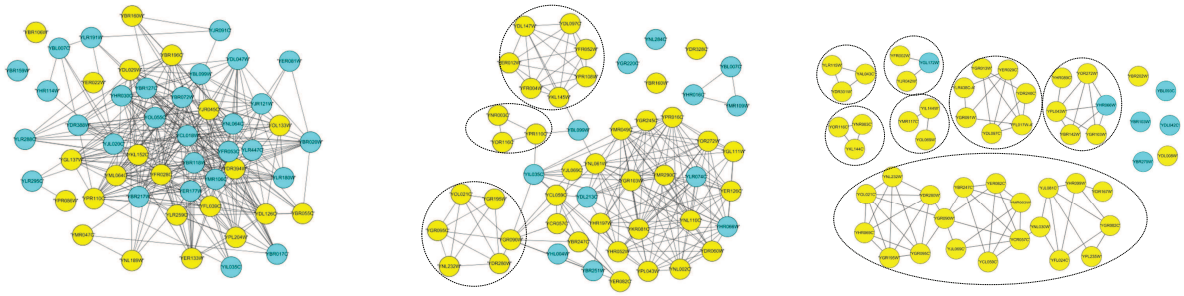


Figure 5: The modules formed by the top 1% identified essential proteins predicted by DC, PeC and EPFOA. Yellow circles are the essential proteins predicted by EPFOA, and blue circles are the non-essential proteins that incorrectly predicted.

4 Conclusion

It is believed that identification of essential proteins is very useful for understanding the minimal requirements for cellular life, and even the disease study and drug design. Although there are many methods have been proposed, it is still a challenge to improve the predicted precision. It is a strong potential way to use computational methods to identify essential proteins. In this study, we propose a novel algorithm EPFOA to boost the performance of essential proteins. We not only analyze the network topological characteristics in the dynamic PPI networks with GO annotation, but also analyze the biological characteristics with the subcellular location information. By comparing with other existing methods, FOCA can more effectively identify the essential proteins with the higher precision. As future work, it would be interesting to apply the EPFOA to other studies, such as gene and disease prediction.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (61672334, 91530320, 61502290, 61401263, and 61320106005) and the Innovation Scientists and Technicians Troop Construction Projects of Henan Province (154200510012).

Bibliography

[1] Binder, J. X., Pletscher-Frankild, S., Tsafou, K., Stolte, C., O'Donoghue, S. I., Schneider, R., Jensen, L. J. (2014); COMPARTMENTS: Unification and Visualization of Protein Subcellular Localization Evidence, *Database, bau012*, 2014.

[2] Bocu, R., Tabirca, S. (2011); The Flag-based Algorithm - A Novel Greedy Method that Optimizes Protein Communities Detection, *International Journal of Computers Communications & Control*, 6(1), 33-44, 2011.

[3] Bonacich, P. (1987); Power and Centrality: A Family of Measures, *American Journal of Sociology*, 92(5), 1170-1182, 1987.

[4] Cherry, J. M., Adler, C., Ball, C., Chervitz, S. A., Dwight, S. S., Hester, E. T., Schroeder, M. (1998); SGD: Saccharomyces Genome Database, *Nucleic Acids Research*, 26(1), 73, 1998.

[5] Consortium, G. O. (2015); Gene Ontology Consortium: Going Forward, *Nucleic Acids Research*, 43 (Database issue), 1049-1056, 2015.

[6] Consortium, G. O., Blake, J. A., Dolan, M., Drabkin, H., Hill, D. P., Li, N., Buza, T. (2013); Gene Ontology Annotations and Resources, *Nucleic Acids Research*, 41(D1), 530-535, 2013.

[7] Cullen, L. M., Arndt, G. M. (2005); Genome-Wide Screening for Gene Function Using RNAi in Mammalian Cells, *Immunology Cell Biology*, 83(3), 217-223, 2005.

[8] 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.

[9] Estrada, E., Rodriguez-Velázquez, J. A. (2005); Subgraph Centrality in Complex Networks, *Physical Review E Statistical Nonlinear Soft Matter Physics*, 71(2), 056103, 2005.

[10] Gavin, A. C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., Dampelfeld, B. (2006); Proteome Survey Reveals Modularity of The Yeast Cell Machinery, *Nature*, 440(7084), 631-636, 2006.

[11] Giaever, G., Chu, A. M., Ni, L., Connelly, C., Riles, L., Véronneau, S., André, B. (2002); Functional Profiling of the Saccharomyces Cerevisiae Genome, *Nature*, 418(6896), 387, 2002.

[12] Gill, N., Singh, S., Aseri, T. C. (2014); Computational Disease Gene Prioritization: An Appraisal, *Journal of Computational Biology A Journal of Computational Molecular Cell Biology*, 21(6), 456-465, 2014.

[13] Hsing, M., Byler, K. G.,Cherkasov, A. (2008); The Use of Gene Ontology Terms for Predicting Highly-Connected 'Hub' Nodes in Protein-Protein Interaction Networks, *BMC Systems Biology*, 2(1), 1-14, 2008.

[14] Jeong, H., Mason, S. P., Barabási, A. L., Oltvai, Z. N. (2001); Lethality and Centrality in Protein Networks, *Nature*, 411(6833), 41-42, 2001.

[15] Jimenezsanchez, G., Childs, B., Valle, D. (2001); Human Disease Genes, *Nature*, 409(6822), 853-855, 2001.

[16] Lei, X., Wang, F., Wu, F. X., Zhang, A., Pedrycz, W. (2016); Protein Complex Identification Through Markov Clustering with Firefly Algorithm on Dynamic Protein-Protein Interaction Networks, *Information Sciences*, 329(6), 303-316, 2016.

[17] Lei, X., Wang, S., Pan, L. (2017); Predicting Essential Proteins Based on Gene Expression Data, Subcellular Localization and PPI Data. *Bio-inspired Computing: Theories and Applications: 12th International Conference, Proceedings of*, 92-105, 2017.

[18] Li, M., Lu, Y., Wang, J., Wu, F. X., Pan, Y. (2015); A Topology Potential-Based Method for Identifying Essential Proteins from PPI Networks, *IEEE/ACM Transactions on Computational Biology Bioinformatics*, 12(2), 372, 2015.

[19] Li, M., Wang, J., Chen, X., Wang, H., Pan, Y. (2011); A Local Average Connectivity-Based Method for Identifying Essential Proteins from the Network Level, *Computational Biology Chemistry*, 35(3), 143-150, 2011.

[20] Li, M., Wang, J., Wang, H., Pan, Y. (2012); Identification of Essential Proteins Based on Edge Clustering Coefficient, *IEEE/ACM Transactions on Computational Biology Bioinformatics*, 9(4), 1070, 2012.

[21] Li, M., Zhang, H., Wang, J. X., Pan, Y. (2012); A New Essential Protein Discovery Method Based on the Integration of Protein-Protein Interaction and Gene Expression Data, *BMC Systems Biology*, 6(1), 15, 2012.

[22] Luo, J., Kuang, L. (2014); A New Method for Predicting Essential Proteins Based on Dynamic Network Topology and Complex Information, *Computational Biology Chemistry*, 52(C), 34, 2014.

[23] Mewes, H. W., Frishman, D., Mayer, K. F. X., Münsterkötter, M., Noubibou, O., Pagel, P., St¨šmpflen, V. (2006); MIPS: Analysis and Annotation of Proteins from Whole Genomes in 2005, *Nucleic Acids Research*, 34 (Database issue), 169-172, 2006.

[24] Newman, M. E. J. (2005); A Measure of Betweenness Centrality Based on Random Walks, *Social Networks*, 27(1), 39-54, 2005.

[25] Pan, W. T. (2012); A New Fruit Fly Optimization Algorithm: Taking the Financial Distress Model as an Example, *Knowledge-Based Systems*, 26(2), 69-74, 2012.

[26] Pan, L., Păun, Gh. (2009); Spiking Neural P Systems with Anti-Spikes. *International Journal of Computers Communications & Control*, 4(3), 273–282, 2009.

[27] Pál, C., Papp, B., Hurst, L. D. (2003); Genomic function: Rate of Evolution and Gene Dispensability, *Nature*, 421(6922), 496-497, 2003.

[28] Păun, Gh. (2000); Computing with Membranes, *Journal of Computer and System Sciences*, 61(1), 108–143, 2000.

[29] Păun, Gh. (2016); Membrane Computing and Economics: A General View, *International Journal of Computers Communications & Control*, 11(1), 105–112, 2016.

[30] Peng, W., Wang, J., Cheng, Y., Lu, Y., Wu, F., Pan, Y. (2015); UDoNC: An Algorithm for Identifying Essential Proteins Based on Protein Domains and Protein-Protein Interaction Networks, *Computational Biology Bioinformatics IEEE/ACM Transactions on*, 12(2), 276-288, 2015.

[31] Przytycka, T. M., Singh, M., Slonim, D. K. (2010); Toward the Dynamic Interactome: It's about Time, *Briefings in Bioinformatics*, 11(1), 15-29, 2010.

[32] Qin, C., Sun, Y., Dong, Y. (2017); A New Computational Strategy for Identifying Essential Proteins Based on Network Topological Properties and Biological Information, *PLoS ONE*, 12(7), e0182031, 2017.

[33] Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., Parisi, D. (2004); Defining and Identifying Communities in Networks, *Proceedings of the National Academy of Sciences of the United States of America*, 101, 2658-2663, 2004.

[34] Ren, J., Wang, J., Li, M., Wang, H., Liu, B. (2011); Prediction of Essential Proteins by Integration of PPI Network Topology and Protein Complexes. *Information Bioinformatics Research and Applications - International Symposium, Isbra 2011, Changsha, China, May 27-29, 2011. Proceedings of*, 12-24, 2011.

[35] Roemer, T., Jiang, B., Davison, J., Ketela, T., Veillette, K., Breton, A., Marta, C. (2003); Large-Scale Essential Gene Identification in Candida Albicans and Applications to Antifungal Drug Discovery, *Molecular Microbiology*, 50(1), 167-181, 2003.

[36] Schlicker, A., Lengauer, T., Albrecht, M. (2010); Improving Disease Gene Prioritization Using the Semantic Similarity of Gene Ontology Terms, *Bioinformatics*, 26(18), i561, 2010.

[37] Song, B., Pan, L., Pérez-Jiménez, M. J. (2016); Cell-Like P Systems with Channel States and Symport/Antiport Rules, *IEEE Transactions on NanoBioscience*, 15(6), 555–566, 2016.

[38] Song, B., Song, T., Pan, L. (2017); A Time-Free Uniform Solution to Subset Sum Problem by Tissue P Systems with Cell Division, *Mathematical Structures in Computer Science*, 27(1), 17–32, 2017.

[39] Song, B., Zhang, C., Pan, L. (2017); Tissue-Like P Systems with Evolutional Symport/Antiport Rules, *Information Sciences*, 378, 177–193, 2017.

[40] Stephenson, K., Zelen, M. (1989); Rethinking centrality: Methods and Examples, *Social Networks*, 11(1), 1-37, 1989.

[41] Tang, X., Wang, J., Zhong, J., Pan, Y. (2014); Predicting Essential Proteins Based on Weighted Degree Centrality, *IEEE/ACM Transactions on Computational Biology Bioinformatics*, 11(2), 407-418, 2014.

[42] Tang, X. W. (2017); Predicting Essential Proteins Using a New Method, *Intelligent Computing Theories and Application: 13th International Conference, ICIC 2017, Liverpool, UK, August 7-10, Proceedings of*, Part II, 301-308, 2017.

[43] Tang, Y., Li, M., Wang, J., Pan, Y., Wu, F. X. (2015); CytoNCA: A Cytoscape Plugin for Centrality Analysis and Evaluation of Protein Interaction Networks, *BioSystems*, 127, 67-72, 2015.

[44] Tu, B. P., Mcknight, S. L. (2005); Logic of the Yeast Metabolic Cycle: Temporal Compartmentalization of Cellular Processes, *Science*, 310(5751), 115, 2005.

[45] Wang, J., Peng, X., Li, M., Luo, Y., Pan, Y. (2011); Active Protein Interaction Network and Its Application on Protein Complex Detection, *IEEE International Conference on Bioinformatics and Biomedicine*, 37-42, 2011.

[46] Wang, J., Peng, X., Peng, W., Wu, F. X. (2014); Dynamic Protein Interaction Network Construction and Applications, *Proteomics*, 14(4-5), 338-352, 2014.

[47] Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., Chen, C. F. (2007); A New Method to Measure the Semantic Similarity of GO Terms, *Bioinformatics*, 23(10), 1274, 2007.

[48] Wang, L., Zheng, X. L., Wang, S. Y. (2013); A Novel Binary Fruit Fly Optimization Algorithm for Solving The Multidimensional Knapsack Problem, *Knowledge-Based Systems*, 48(2), 17-23, 2013.

[49] Watts, D. J., Strogatz, S. H. (1998); Collective Dynamics of 'Small-World' Networks, *Nature*, 393(6684), 440, 1998.

[50] Winzeler, E. A., Shoemaker, D. D., Astromoff, A., Liang, H., Anderson, K., Andre, B., Bussey, H. (1999); Functional Characterization of the S. cerevisiae Genome by Gene Deletion and Parallel Analysis, *Science*, 285(5429), 901-906, 1999.

[51] Wuchty, S. (2001); Scale-Free Behavior in Protein Domain Networks, *Molecular Biology Evolution*, 18(9), 1694, 2001.

[52] Wuchty, S., Stadler, P. F. (2003); Centers of Complex Networks, *Journal of Theoretical Biology*, 223(1), 45, 2003.

[53] Yan, W., Sun, H., Wei, D., Enrico, B., Gabriella, V., Ying, X., Liang, Y. (2014); Identification of Essential Proteins Based on Ranking Edge-Weights in Protein-Protein Interaction Networks, *PLoS ONE*, 9(9), e108716, 2014.

[54] Zeng, X., Lin, W., Guo, M., Zou, Q. (2017). A comprehensive overview and evaluation of circular RNA detection tools, *PLoS Computational Biology*, 13(6), e1005420, 2017.

[55] Zhang, R., Lin, Y. (2009); DEG 5.0, A Database of Essential Genes in both Prokaryotes and Eukaryotes, *Nucleic Acids Research*, 37 (Database issue), D455, 2009.

[56] Zhang, X. F., Dai, D. Q., Ouyang, L., Yan, H. (2014); Detecting Overlapping Protein Complexes Based on a Generative Model with Functional and Topological Properties, *BMC Bioinformatics*, 15(1), 186, 2014.

[57] Zhang, Y., Lin, H., Yang, Z., Wang, J. (2013); Construction of Ontology Augmented Networks for Protein Complex Prediction, *PLoS ONE*, 8(5), : e62077, 2013.

[58] Zhao, B., Wang, J., Li, M., Wu, F. X., Pan, Y. (2014); Detecting Protein Complexes Based on Uncertain Graph Model, *IEEE/ACM Transactions on Computational Biology Bioinformatics*, 11(3), 486-497, 2014.

[59] Zhu, C., Wu, C., Aronow, B. J., Jegga, A. G. (2014); Computational Approaches for Human Disease Gene Prediction and Ranking, *Advances in Experimental Medicine Biology*, 799, 69, 2014.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 383-390, June 2018.

Reduction of Conditional Factors in Causal Analysis

H. Liu, I. Dzitac, S. Guo

Haitao Liu*, Sicong Guo
1. Institute of Intelligence Engineering and Mathematics
Liaoning Technical University, Fuxin 123000, China
2. College of Science
Liaoning Technical University, Fuxin 123000, China
*Corresponding author: liuhaitao@lntu.edu.cn

Ioan Dzitac
1. Aurel Vlaicu University of Arad
310330 Arad, Elena Dragoi, 2, Romania
ioan.dzitac@uav.ro
2. Agora University of Oradea
410526 Oradea, P-ta Tineretului 8, Romania,
idzitac@univagora.ro

Abstract: Faced with a great number of conditional factors in big data causal analysis, the reduction algorithm put forward in this paper can reasonably reduce the number of conditional factors. Compared with the previous reduction methods, we take into consideration the influence of conditional factors on resulted factors, as well as the relationship among conditional factors themselves. The basic idea of the algorithm proposed in this paper is to establish the matrix of mutual deterministic degrees in between conditional factors. If a conditional factor f has a greater deterministic degree with respect to another conditional factor h, we will delete the factor h unless factor h has a greater deterministic degree with respect to f, then delete factor f in this case. With this reduction, we can ensure that the conditional factors participating in causal analysis are as irrelevant as possible. This is a reasonable requirement for causal analysis.
Keywords: factors space, causal analysis, reduction of factors, fuzzy logic.

1 Introduction

Causal analysis in factors space [18] is proposed in the paper [22], which extracts causal rules from the background distribution in between a group of factors. This is the original methodology provided by the factor space for the machine learning, classification and decision-making and so on. The paper [13] applies those causal rules to causal reasoning, and the paper [17] improves the inductive algorithm introduced in paper [22]. The paper [1] puts forward that the slip-differential algorithm, improving the precision of causal reasoning. The paper [15] gives the rule extraction with respect to multi-result factors, which connects multi-label learning theory [5]. The paper [2] presents a reasonable statement on logistic regression based on fuzzy sets and the factor space theory. The paper [14] introduces the historic background of factors space and its relationship with formal concept analysis [6]. A lot of theoretical papers about factors space can be found in the reference [3, 4, 7–12, 16, 19–21, 23, 24]. All this lays a complete foundation for the unified depiction of causal induction and reasoning in artificial intelligence. However, in the face of the impact of big data, the number of factors to be processed by causal analysis is surprisingly large. We will discuss how to simplify and merge the large number of conditional factors in this paper.

The idea of the article [22] is that the factor which has the strongest influence on the resulted factor will be used first. By using it, we can have a causal rule and delete some data. Repeating

the process, when all the data are deleted all unused conditional factors were reduced. This reduction method is determined by the deterministic degrees of conditional factors with respect to the resulted factor. This paper makes a supplement to the idea of reduction. Not only do we consider the influence of the conditional factors on the result factor, but also the relationship between the condition factors is taken into consideration. The deterministic degrees of a condition factor with respect to other factors should be considered. The conditional factors are reduced or merged according to the degree of mutual determination, and the last is a set of conditional factors that are as not related to each other as possible, which is the best condition for causal analysis.

The structure of this paper is that section 2 introduces the mutual relationship in between conditional factors, and section 3 introduces the reduction algorithm of conditional factors. Section 4 is a short conclusion. This paper is a mathematical method without specific examples.

2 Mutual relationship in between conditional factors

The factor is the quality root, each factor in command of a string of attributes. For example, the color is a factor, which commands the red, orange, yellow, green, blue, blue, purple and so on. It mathematically is defined as a mapping $f : U \to X(f)$. f is color, for example, U are a group of cars, $X(f)$=$\{$red, orange, yellow, green, green, blue, purple$\}$, which draws our attention from the group of cars to their colors. $X(f)$ is called the state space of the factor f , where the states are described by natural language words, called the qualitative states; of course, factors also can have quantitative state space, then back to the variable. The factor is the promotion of variables. Factor f is regular if there are at least two objects u and v such that $f(u) \neq f(v)$.

Considering a set of basic factors $F^* = \{f_1, \cdots, f_n\}$, we can define a synthetic factor by any subset $f = \{f_{(1)}, \cdots, f_{(k)}\}$ of F^* with the state space $X = X(f_{(1)}) \times \cdots \times X(f_{(k)})$ ($\times$ stands for Cartesian product). Denote the synthetic factor as $f = \{f_{(1)} \cup \cdots \cup f_{(k)}\}$. It is easy to prove that $P(F^*) = (P(F^*), \cup, \cap, c)$ forms a factorial Boolean algebra, where the operations $\cup$ and $\cap$ are called the synthesis and separation of factors respectively.

Denote $X_{F^*} = \{X(f)\}_{(f \in P(F^*))}$, and $\phi = (U, X_{F^*})$ is called the factor space defined on U.

A factor f defines an equivalence relation $\sim$ in the domain U: For any $u, v \in U$, $u \sim v$ if and only if $f(u) = f(v)$. Denote the subclass of U containing u as $[u]_f = \{v \in U | f(v) = f(u)\}$. We call that $H(f, U) = \{[u]_f | u \in U\}$ the division of U by f.

We call that f is more specific than h, denoted as $H(f, U) \rangle H(h, U)$, if for any u, there is an v in U such that $[v]_f \subseteq [u]_h$, and for any u, there is an v in U such that $[u]_f \subseteq [v]_h$. It is obvious that $H(f, U) \rangle H(h, U)$ if and only if $H(f \cup h, U) = H(f, U)$, in this case, for any $a \in X(h)$, there are $a_1, \cdots, a_t \in X(f)$, such that $[a]_h = [a_1]_f + \cdots + [a_t]_f$. We call that f and h are equivalent if $H(f, U) \rangle H(h, U)$ and $H(h, U) \rangle H(f, U)$. Suppose that the numbers of subclasses in the divisions of f and h are s and t respectively, and suppose that $s, t > 1$. If $H(f \cup h, U)$ is the roughest common more specific division of $H(f, U)$ and $H(h, U)$, then we call that f and h are independent in division. In this case, for any $b \in X(h)$, there are $a_1, \cdots, a_s \in X(f)$ such that $[b]_h = [a_1]_f + \cdots + [a_s]_f$; and for any $a \in X(f)$, there are $b_1, \cdots, b_t \in X(h)$ such that $[a]_f = [b_1]_h + \ + [b_t]_h$.

Given a factors space $\phi = (U, X_{F^*})$, selecting $f_1, \cdots, f_k$ and g from X_{F^*}, called a set of conditional factors and a result factor respectively, and extracting m objects from U to form a sample domain U', we obtain the combined states data of these objects with respect to the $k+1$ factors. Causal analysis aims to extract causal rules from conditions to the result based on the sample distribution of U'. One of the key concepts is the deterministic degree of factor f_i with respect to g.

Definition 1. (Wang2015) If there is an object $u \in U'$ such that $[u]_{f_i} \subseteq [u]_g$, then we say that $[u]_{f_i}$ is a deterministic class of f_i with respect to g. The ratio d of the number of objects in all deterministic classes of f_i with respect to g and the number of objects in U' is called the determination degree of f_i with respect to g.

When $f_j(u) = f_j(v)$,we have that $[u]_{f_i} = [v]_{f_i}$. To avoid repetition, denote $[u]_{f_i} = [v]_{f_i} = [a]_{f_i}$, then we have

$$d(f_j, g) = \sum\{|[a]_{f_i}||[a]_{f_i} \textit{ is a deterministic class of } f_i \textit{ on } g\}/m \tag{1}$$

where $|A|$ stands for the number of elements in A.

In this Section, we will consider the deterministic degree $d(f, h)$ of a conditional factor f with another conditional factor h. The whole theory is applied on a sampling $U' \subseteq U$.

Theorem 2. *Let f, h be two conditional factors on sample U'. Factor f is more specific than h on U' if and only if $d(f, h) = 1$.*

Proof: Suppose that f is more specific than h on U'. For any $u \in U'$, there is $v \in U'$ such that $[v]_f \subseteq [u]_h$, it means that $[v]_f$ is a deterministic subclass of f with respect to h. Therefore, all the elements of U' is covered by deterministic subclasses of f with respect to h, then, we have that $d(f, h) = 1$.

Inversely, suppose that $d(f, h) = 1$. For any $a \in X(h)$, let $[a]$ be the subclass that has the state a under h, there must be an element $u \in U'$ such that $h(u) = a$, and then $[a] = [u]_h$. Since $d(f, h) = 1$, we have that $[u]_f \subseteq [u]_h$, i.e., $[u]_f \subseteq [a]$; For any $a \in X(f)$, let $[a]$ be the subclass has the state a under f, there must be an element $u \in U'$ such that $f(u) = a$. Since $d(f, h) = 1$, we have that $[u]_f \subseteq [u]_h$, i.e., $[u] \subseteq [u]_f$. Therefore, the factor f is more specific than h. □

Theorem 3. *If f is more specific than h on U', and the two factors h and f are not equivalent, then $d(h, f) < 1$.*

Proof: Suppose that $d(h, f) = 1$. According to proposition 2, h is more specific than f, and then f and h are equivalent U'. This is a contradiction. □

Theorem 4. *If f is more specific than h on U', then $d(f, g) \geq d(h, g)$.*

Proof: Suppose that $[a]$ is a deterministic subclass of h ($a \in X(h)$). There is $u \in [a]_h$ such that $h(u) = a$. Since that f is more specific than h on U', we have that $H(f \cup h, U) = H(f, U)$, and then $[a]_h = [a_1]_f + \cdots + [a_t]_f$, where $[a_1]_f, \cdots, [a_t]_f$ are deterministic degrees of f on U' with respect to g both. It is obvious that $d(f, g) \geq d(h, g)$. □

Theorem 5. *If f and h are two regular factors mutual independent in division on U', then $d(f, h) = d(h, f) = 0$.*

Proof: Since f and h are two regular factors mutual independent in division on U' for any $b \in X(h)$, there are $a_1, \cdots, a_s \in X(f)$ such that $[b]_h = [a_1]_f + \cdots + [a_s]_f$; and for any $a \in X(f)$, there are $b_1, \cdots, b_t \in X(h)$ such that $[a]_f = [b_1]_h + \cdots + [a_t]_h$, and it ensures that $[a]_f \setminus [b]_h \neq \Phi$ and $[b]_h \setminus [a]_f \neq \Phi$ hold for any $a \in X(f)$ and any $b \in X(h)$. then $d(f, h) = d(h, f) = 0$. □

There are three kinds of relationship between conditional factors: 1. $d(f, h)$ is rather larger and $d(h, f)$ is rather smaller; 2. $d(f, h)$ and $d(h, f)$ are rather larger both; 3. $d(f, h)$ and $d(h, f)$ are rather smaller both. According to the statements mentioned above, in case 1, if $d(f, g)$ is larger, then we need not the factor h when f is taken part in with respect to the result g; in case 2, factors f, h are related to each other closely, and they are not suitable to be conditional simultaneously, and need to do reduction; in the case 3, factors f, h are rather independent, so they don't need to be deleted provided they have important influence to the resulted factor.

Table 1: Conditional Factors

factor	*name*	*state space*
f_1	Age	$X(f_1)$={Old, Middle, Young}
f_2	Income	$X(f_2)$={High, Average, Low}
f_3	Student	$X(f_3)$={Y, N}
f_4	Credit	$X(f_4)$={Very-good, Good, Un-recorded}
f_5	Education	$X(f_5)$={Primary, Junior, University, Graduated}
f_6	Civil	$X(f_6)$={Civil, Private}
f_7	Housing	$X(f_7)$={Rent, Narrow, Mansion}
f_8	Car	$X(f_8)$={Car, Bike}
f_9	Health	$X(f_9)$={Healthy, Sickness}
f_{10}	Residence	$X(f_{10})$={Town, Rural}

Table 2: Causal Data

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}
f_1	O	O	M	O	O	M	M	M	M	Y	Y	Y	Y	Y
f_2	L	A	H	A	H	L	A	H	L	A	H	A	L	L
f_3	N	N	N	N	N	N	N	N	N	Y	N	Y	Y	Y
f_4	U	U	V	G	V	U	G	V	G	V	V	V	G	U
f_5	P	J	G	U	U	J	U	J	P	G	U	U	U	G
f_6	C	C	C	C	C	P	C	P	P	P	C	P	P	P
f_7	N	N	M	M	M	N	N	M	N	N	M	R	R	R
f_8	B	B	C	C	B	B	C	C	B	C	C	B	B	B
f_9	H	S	H	H	S	H	S	S	H	H	H	S	H	S
f_{10}	R	T	T	T	T	R	T	R	R	T	T	T	T	T
g	0	1	2	1	2	0	1	2	0	2	2	2	1	0

3 Reduction of conditional factors

Causal analysis aims to extract the rules form conditional to resulted factors; the more independent the better the conditional factors. The reduction of conditions factors obeys such a principle: For a pair of the factors with higher deterministic degrees with respect to the resulted factor g both, delete one of them except their mutual deterministic degrees are smaller both (i.e., in the case 3). This principle aims to take conditional factors into causal analysis as independent as possible.

Algorithm

Step 1. Rank the conditional factors according to their deterministic degrees with respect to the resulted factor g from high to low;

Step 2. Write the matrix of deterministic degree between conditional factors;

Step 3. For any i and j, if ($d(f_i, f_j) > 0.5$ or $d(f_j, f_i) > 0.5$) and $d(f_i, f_j) > d(f_j, f_i)$, then delete f_j.

The remaining factor sequence is the conditional sequence that is required by the reduction. If causal analysis is performed according to this sequence, the sequence will be terminated when the causal tree is formed, and all unused conditional factors are deleted at all.

Example. In customer analysis, the goal is to open the market. The utility factor is the purchasing power of the customer, and the form factor is the information of the customers. Take the form factors as the conditionals; the benefit factor should be the result to do the causal analysis. The conditional factors considered are listed in Table 1.

Selecting 14 customers to form a sampling universe

$U' = \{u_1\ u_2\ u_3\ u_4\ u_5\ u_6\ u_7\ u_8\ u_9\ u_{10}\ u_{11}\ u_{12}\ u_{13}\ u_{14}\}$,

Table 2 presents causal data type.

Table 3: Frequencies of factors

f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
0	4	0	6	2	0	0	0	0	0

Table 4: The matrix of mutual deterministic degrees between conditional factors

	f_4	f_2	f_5	f_1	f_3	f_6	f_7	f_8	f_9	f_{10}
f_4		0	0	0	0	0	0	4	0	0
f_2	4		0	0	5	0	4	**9**	0	5
f_5	**2**	2		0	2	0	2	4	4	**8**
f_1	0	0	0		**9**	4	0	0	0	5
f_3	**0**	2	0	2		4	0	0	0	4
f_6	0	0	0	0	0		0	0	0	0
f_7	0	0	0	3	3	3		3	0	0
f_8	0	**0**	0	0	0	0	0		0	0
f_9	0	0	0	0	0	0	0	0		0
f_{10}	0	0	0	0	4	0	0	0	0	

The steps of reduction of conditional factors are shown as follows:

Step 1. Reordering conditional factors according to their deterministic degrees with respect to g. Remember that m=14, to be more simple, we write all frequencies by 14 times. The results are given in Tables 3.

The new order is shown as: f_4, f_2, f_5, f_1, f_3, f_6, f_7, f_8, f_9, f_{10}.

Step 2. The matrix of mutual deterministic degrees between conditional factors is listed in Table 4.

Step 3.

When $i = 2$, $j = 8$, $d(f_i, f_j) = 9/14 > 0.5$ and $d(f_i, f_j) > d(f_j, f_i) = 0$, delete f_8;

When $i = 5$, $j = 10$, $d(f_i, f_j) = 8/14 > 0.5$ and $d(f_i, f_j) > d(f_j, f_i) = 0$, delete f_{10};

When $i = 1$, $j = 3$, $d(f_i, f_j) = 9/14 > 0.5$ and $d(f_i, f_j) > d(f_j, f_i) = 0$, delte f_3.

After deleting the three conditional factors, the new causal analysis data style is presented in Table 5

According to the causal analysis [22], do rule extraction by f_4 to get that

Rule 1: If Credit is very good, then the purchasing power is #2

Taking out those customers having very good credit from U', Table 6 presents newer causal analysis data style.

Do rule extraction by f_4 and f_2 to get that

Rule 2: If Credit is unrecorded and Income is low, then the purchasing power is #0;

Rule 3: If Credit is unrecorded and Income is average, then the purchasing power is #1;

Rule 4: If Credit is good and Income is average, then the purchasing power is #1;

Table 5: New Causal Data

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_{11}	u_{12}	u_{13}	u_{14}
f_4	U	U	V	G	V	U	G	V	G	V	V	V	G	U
f_2	L	A	H	A	H	L	A	H	L	A	H	A	L	L
f_5	P	J	G	U	U	J	U	J	P	G	U	U	U	G
f_1	O	O	M	O	O	M	M	M	M	Y	Y	Y	Y	Y
f_6	C	C	C	C	C	P	C	P	P	P	C	P	P	P
f_7	N	N	M	M	M	N	N	M	N	N	M	R	R	R
f_9	H	S	H	H	S	H	S	S	H	H	H	S	H	S
g	0	1	2	1	2	0	1	2	0	2	2	2	1	0

Table 6: New Causal Data (with 8 factors)

	u_1	u_2	u_4	u_6	u_7	u_9	u_{13}	u_{14}
f_4	U	U	G	U	G	G	G	U
f_2	L	A	A	L	A	L	L	L
f_5	P	J	U	J	U	P	U	G
f_1	O	O	O	M	M	M	Y	Y
f_6	C	C	C	P	C	P	P	P
f_7	N	N	M	N	N	N	R	R
f_9	H	S	H	H	S	H	H	S
g	0	1	1	0	1	0	1	0

Table 7: New Causal Data (with 2 factors)

U'	u_9	u_{13}
f_4	G	G
f_2	L	L
f_5	P	U
f_1	M	Y
f_6	P	P
f_7	N	R
f_9	H	H
g	0	1

Taking out those customers having been contributed to rule extraction from U', the newer causal analysis data style is given in Table 7.

Do rule extraction by f_4, f_2 and f_5 to get that

Rule 5: If Credit is Good, Income is low, and Education is Univ., then the purchasing is #1;

Rule 6: If Credit is Good, Income is low, and Education is Prim., then the purchasing is #0;

Now, the universe U' has been empty, the rule extraction has finished. We just select three factors in use, all of others have been deleted at all. What are the relationship in between the three factors?

$d(f_4, f_2) = 0 < 0.5$, $d(f_2, f_4) = 4/14 < 0.5$,

$d(f_4, f_5) = 0 < 0.5$, $d(f_5, f_4) = 2/14 < 0.5$,

$d(f_2, f_5) = 0 < 0.5$, $d(f_5, f_2) = 2/14 < 0.5$.

All of mutual deterministic degrees between them are small, which satisfy the requirement of causal analysis.

4 Conclusion

In the face of the challenge of big data, the number of conditional factors in causal analysis is very large, so the reduction of conditional factors is an important task. The proposed reduction algorithm can reasonably reduce the number of conditional factors. Compared with the previous reduction methods, we take into consideration the influence of conditional factors on resulted factors, as well as the relationship among conditional factors themselves. In this paper we consider the mutual deterministic degrees in between conditional factors. Such reduction ensures the conditional factors are selected as independent as possible, Causal analysis requires such selection, and this improvement is of great importance in practice.

Acknowledgement

The authors specially thank Professor P. Z. Wang for his guidance and modification. This study was partially supported by the grants (Grant Nos. 61350003, 11401284, 70621001, 70531040) from the Natural Science Foundation of China, and the grant (Grant Nos. L2014133) from the department of education of Liaoning Province.

Bibliography

[1] Bao, Y. K.; Ru, H.Y.; Jin, S.J.(2014); A new algorithm of knowledge mining in factor space, *Journal of Liaoning Technical University (Natural Science)*, 33(8), 1141–1144, 2014.

[2] Cheng, Q.F.; Wang, T.T.; Guo S.C.; Zhang, D. Y.; Jing K.; Feng, L.; Wang P.Z. (2017); The Logistic Regression from the Viewpoint of the Factor Space Theory, *International Journal of Computers Communications & Control*, 12(4), 492–502, 2017.

[3] Dzitac I. (2015), The Fuzzification of Classical Structures: A General View, *International Journal of Computers Communications & Control*, 10(6), 772-788, 2015.

[4] 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.

[5] Furnkranz, J.; Hullermeier, E.; Mencia, E.L.; Brinker, K.(2008); Multilabel classification via calibrated label ranking, *Machine Learning*, 73(2), 133-153, 2008.

[6] Ganter, B.; Wille, R. (1996); Formal concept analysis, *Wissenschaftliche Zeitschrift-Technischen Universitat Dresden*, 45, 8–13, 1999.

[7] Kandel, A.; Peng, X.T.; Cao, Z.Q.; Wang P.Z. (1990); Representation of concepts by factor spaces, *Cybernetics and Systems: An International Journal*, 21(1), 43–57, 1990.

[8] Li, H.X.; Wang, P.Z.; Yen, V.C. (1998); Factor spaces theory and its applications to fuzzy information processing.(I). The basics of factor spaces, *Fuzzy Sets and Systems*, 95(2), 147–160, 1998.

[9] Li, H.X., Yen, V.C.; Lee, E.S. (2000); Factor space theory in fuzzy information processing-Composition of states of factors and multifactorial decision making, *Computers & Mathematics with Applications*, 39(1), 245–265, 2000.

[10] Li, H.X.; Yen, V.C.; Lee, E.S. (2000); Models of neurons based on factor space, *Computers & Mathematics with Applications*, 39(12), 91–100, 2000.

[11] Li, H. X.; Chen, C.P.; Yen, V.C., Lee, E.S. (2000); Factor spaces theory and its applications to fuzzy information processing: Two kinds of factor space canes, *Computers & Mathematics with Applications*, 40(6-7), 835–843, 2000.

[12] Li, H.X.; Chen, C.P., Lee, E.S. (2000); Factor space theory and fuzzy information processing-Fuzzy decision making based on the concepts of feedback extension, *Computers & Mathematics with Applications*, 40(6-7), 845–864, 2000.

[13] Liu, H.T.; Guo, S.C. (2015); Inference model of causality analysis, *Journal of Liaoning Technical University(Natural Science)*, 34(1), 124–128, 2015.

[14] Liu, H.T.; Dzitac, I.; Guo, S.C. (2018); Reduction of conditional factors in causal analysis, *International Journal of Computers Communications & Control*, 13(1), 83–98, 2018.

[15] Qu, W.H.; Liu, H.T.; Guo, S.Z.(2017); Multi-target causality analysis in factor space, *Fuzzy Systems & Mathematics*, 31(6), 74–81, 2017.

[16] Vesselenyi, T.; Dzitac, I.; Dzitac, S.; Vaida, V. (2008); Surface roughness image analysis using quasi-fractal characteristics and fuzzy clustering methods, *International Journal of Computers Communications & Control*, 3(3), 304–316, 2008.

[17] Wang, H.D.; Wang, P.Z.; Shi, Y.; Liu, H.T. (2014); Improved factorial analysis algorithm in factor spaces, *International Conference on Informatics*, 201–204, 2014.

[18] Wang, P.Z.; Sugeno, M. (1982); The factor fields and background structure for fuzzy subsets, *Fuzzy Mathematics*, 2(2), 45–54, 1982.

[19] Wang, P.Z. (1990); A factor spaces approach to knowledge representation, *Fuzzy Sets and Systems*, 36(1), 113–124, 1990.

[20] Wang, P.Z.; Zhang, X.H.; Lui, H.C.; Zhang, H.M.; Xu, W. (1995); Mathematical theory of truth-valued flow inference, *Fuzzy Sets and Systems*, 72(2), 221–238, 1995.

[21] Wang, P.Z.; Jiang, A. (2002); Rules detecting and rules-data mutual enhancement based on factors space theory, *International Journal of Information Technology & Decision Making*, 1(01), 73–90, 2002.

[22] Wang, P.Z.; Guo, S.C.; Bao, Y.K.; Liu, H.T. (2014); Causality analysis in factor spaces, *Journal of Liaoning Technical University (Natural Science)*, 33(7), 1–6, 2015.

[23] Yuan, X.H.; Wang, P.Z.; Lee, E.S. (1992); Factor space and its algebraic representation theory, *J Math Anal Appl.*, 171(1), 256–276, 1992.

[24] Yuan, X.H.; Wang, P.Z.; Lee, E.S. (1994); Factor Rattans, Category FR (Y), and Factor Space, *Journal of Mathematical Analysis and Applications*, 186(1), 254–264, 1994.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 391-407, June 2018.

Attribute Selection Method based on Objective Data and Subjective Preferences in MCDM

X. Ma, Y. Feng, Y. Qu, Y. Yu

Xiaofei Ma, Yi Feng
Technology Planning
Dalian Commodity Exchange
Dalian City, China, 116023
maxf@dce.com.cn, fengyi@dce.com.cn

Yi Qu*, Yang Yu
Agricultural Bank of China Data Center
88 Aoni Road, Pudong New Area
Shanghai City, China, 200131
*Corresponding author: 57638907@qq.com
yuyinyang@126.com

Abstract: Decision attributes are important parameters when choosing an alternative in a multiple criteria decision-making (MCDM) problem. In order to select the optimal set of decision attributes, an analysis framework is proposed to illustrate the attribute selection problem. Then a two-step attribute selection procedure is presented based on the framework: In the first step, attributes are filtered by using correlation algorithm. In the second step, a multi-objective optimization model is constructed to screen attributes from the results of the first step. Finally, a case study is given to illustrate and verify this method. The advantage of this method is that both external attribute data and subjective decision preferences are utilized in a sequential procedure. It enhances the reliability of decision attributes and matches the actual decision-making scenarios better.
Keywords: attribute selection, multi-criteria decision-making (MCDM), multi-objective optimization, attribute correlation.

1 Introduction

Multi-criteria decision-making (MCDM) is successfully applied to help decision makers (DMs) choose optimal alternatives. During the past few decades, various MCDM methods have been proposed based upon different philosophies such as multi-attribute utility, the analytic hierarchy process (AHP), outranking methods and so on. Meanwhile, many decision support systems (DSSs) have been designed in MCDM to assist DMs in analyzing problems and making decisions more easily. MCDM deals with a general class of problems that contains multiple attributes, objectives and criteria [20]; and alternatives are determined based on many qualitative or quantitative criteria which are generally complicated and assessed by more relevant attributes. However, not all of them can be used in decision-making procedure, because they contain plenty of redundant or "noisy" attributes and it will lead to useless decision alternatives. Therefore, the effectiveness of an alternative is highly dependent on the set of decision attributes.

The concept of "optimal" can be illustrated by two aspects: (1) Elements of the set are highly related to the MCDM problem for decision purpose; (2) The set of attributes is parsimonious, and the selected alternative will be suboptimum if one of these attributes is omitted. The rationale of attribute selection is similar to feature selection in data mining field. There are a lot of feature selection methods have been proposed, but only a few of them can be applied in decision-making process directly which are mentioned in the Literature Review part. Most of these methods can't

account for both objective and subjective perspectives, so there is a low reliability of decision attributes. This study is focused on how to select the optimal set of decision attributes for a MCDM problem, external attribute data and subjective decision preferences are utilized in a sequential procedure to enhance the reliability of decision attributes.

The attribute selection method for decision problem should account for both objective and subjective perspectives, more precisely, this paper merges attribute data (objective) with DMs' requirement (subjective). Utilize the former for rational, constrained modeling and the latter for adapting specific problem issues to the decision-making process. In order to obtain the optimal set of attributes, an analysis framework for attribute selection problem and a two-step screening procedure is conducted in this paper. The specific procedure is illustrated in Figure 1.

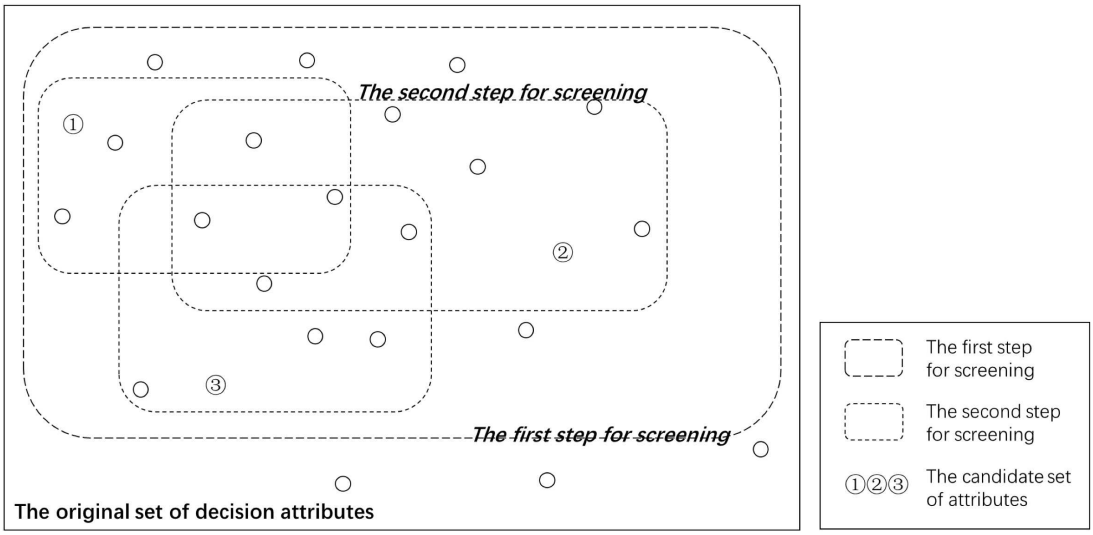


Figure 1: The illustration for the presented method

This new attribute selection method in MCDM contributes to selecting the optimal set of decision attributes and helping DMs choose optimal alternatives better. Meanwhile, this method merges objective data and subjective preferences to improve performance for actual decision scenarios.

2 Literature review

Many feature selection methods have been proposed, and they are usually classified into three classes: "filter" methods, "wrapper" methods and "embedded" methods [2,7,8,14]. Wu [19] proposed using the fuzzy and grey Delphi methods to identify a set of reliable attributes and, based on these attributes, transforming big data to a manageable scale to consider their impacts. Meinshausen et al. [11] demonstrated linear model and Gaussian model of variable selection consistency in higher dimensional case. Although they have good performance, they cannot be applied in decision-making process directly (partial principle may be accepted). Only a few papers mentioned attribute selection in decision-making problems; for instance, Chun [4] considered the "optimizing" and "satisficing" (a portmanteau of satisfy and suffice) attributes and deal with the multi-attribute decision problem with sequentially presented decision alternatives; Dai et al. [6] constructed three attribute selection approaches in context of incomplete decision systems based on information-theoretical measurement of attribute importance; in order to screen the critical factors influencing the stability of perilous rock, Meng et al. [12] used fuzzy compromise TOPSIS method to calculate the importance of attributes in the decision problem.

Wu [18] used Fuzzy Delphi method to screen out the unnecessary attributes to deal with the complex interrelationships among the aspects and attributes. Attribute selection procedure can improve the decision performance, but no papers used this procedure in MCDM problems.

3 Method

3.1 Preliminaries

Trapezoidal fuzzy numbers

Definition 1. Let X be a universe set. A fuzzy $\widetilde{a}$ in a universe of discourse X is characterized by a membership function $\mu_{\widetilde{a}}(x)$, which associates with each element x in X, a real number in the interval [0,1]. The function is termed the grade of membership of x in $\widetilde{a}$.

Definition 2. A tuple $\widetilde{A} = (a, b, c, d), a \leq b \leq c \leq d$, is called a trapezoidal fuzzy number (TFN) if its membership function is

$$\mu_{\widetilde{a}}(x) = \begin{cases} (x-a)/(b-a) & a \leq x \leq b \\ 1 & b \leq x \leq c \\ (d-x)/(d-c) & c \leq x \leq d \\ 0 & otherwise \end{cases}$$

Where *a, b, c, d* are real numbers.

Definition 3. Given two trapezoidal fuzzy numbers $\widetilde{A} = (a_1, b_1, c_1, d_1)$, $\widetilde{A} = (a_2, b_2, c_2, d_2)$, and a real number λ, the main operations can be expressed as follows:

①$\widetilde{A_1} \bigoplus \widetilde{A_2} = (a_1 + a_2, b_1 + b_2, c_1 + c_2, d_1 + d_2)$

②$\widetilde{A_1} \bigotimes \widetilde{A_2} = (a_1 a_2, b_1 b_2, c_1 c_2, d_1 d_2)$

③$\lambda \bigotimes \widetilde{A_2} = (\lambda a_1, \lambda b_1, \lambda c_1, \lambda d_1)$

Multi-objective optimization

Let χ be a vector containing n decision variables and in a universe of discourse X. Mathematically, an optimization problem with p objective functions can be expressed as (Mahdi and Seyed, 2012):

minimize $f_i(x)$ for $i = 1, 2, ..., p$

subject to: $g_j(x) \leq 0, j = 1, 2, ..., q,$

$h_j(x) = 0, j = q+1, 2, ..., m$

where $x = (x_1, x_2, ..., x_n)$, x_l is the lth decision variable; p and m are respectively the numbers of objective function and constraint.

A variety of methods can be used to solve this problem. One popular method is to combine those objectives into a single composite objective so that traditional mathematical programming methods can be applied. And other approaches are based on the Pareto optimum concept, and more specifically, let $y = (y_1, y_2, ..., y_n)$ be another vector containing **n** decision variables in X.

Definition 4. (Domination) $x \in X$ dominates $y \in X$, denoted $x \succ y$, if $\forall j \in p$, $v_j(x) \leq v_j(y)$ with at least one strict inequality.

Definition 5. (Pareto Optimal) x is a Pareto Optimal (PO) solution, if there is no $y \in X$ such that $y \succ x$. The set of all Pareto optimal solutions in X is denoted PO(X).

These methods include Multiple Objective Genetic Algorithm (MOGA), Non-dominated Sorting Genetic Algorithm (NSGA), NSGA-II, Multi-objective Multi-state Genetic Algorithm (MOMS-GA), Niched-Pareto Genetic Algorithm (NPGA) and Multi-objective Particle Swarm Optimization (MOPSO) [9, 21].

An analysis framework of attribute selection under MCDM

Multi-Criteria Decision Making (MCDM) has been widely applied as a well-known branch of decision making, and can be further divided into multi-objective decision making (MADM) and multi-attribute decision making (MODM) [5, 13]. The whole hierarchical structure of a MCDM problem is shown in Figure 2, and it can be unfolded from four hierarchies from top to the bottom, namely, objective, criteria, attribute and alternative.

(1) Objective Hierarchy: Generally contain multiple decision objectives given by DMs which reflect different decision requirement.

(2) Criteria Hierarchy: Criteria Hierarchy is to realize a fair comparison of alternatives from the various aspects of Objective Hierarchy, and it is also developed in a hierarchical fashion, starting from some general but imprecise criteria description, which is refined into more precise sub- and sub-sub criteria.

(3) Attribute Hierarchy: Attribute Hierarchy is determined by criteria Hierarchy and Objective Hierarchy. It is generally assumed that each criterion can be represented by some measurable attributes of the consequences arising from implementation of any particular decision alternative [16], as well as objective. No matter what kind of MCDM methods, decision attributes are the bottom and direct evaluative parameters to determine alternatives.

(4) Alternative Hierarchy: Consist from candidate alternatives evaluated by the set of decision attribute.

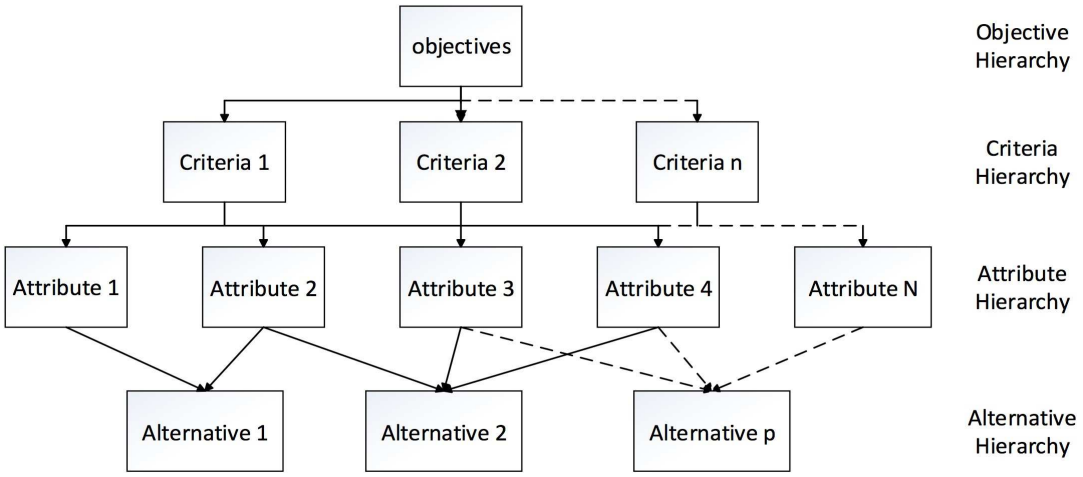


Figure 2: The hierarchical structure of a MCDM problem

This paper focus on how to obtain the optimal set of decision attributes from Attribute Hierarchy for evaluating alternatives.

Rationale of the proposed method

Let $O = \{O^1, O^2, ..., O^\gamma\}$ be a finite set of objectives, and $C = \{C_1, C_2, ..., C_n\}$ be the set of criteria. In most actual cases, objectives and criteria are always predetermined by MDs according to different knowledge background and decision experiences, thus we assume that all of objectives and criteria (or sub-criteria) are given.

We define decision attribute selection procedure as a sequential selection procedure based on objective and subjective information. It means that a series of screening steps will be applied in sequence to obtain an optimal set of attributes. A sequential screening can be described as follows [3]:

$$Scr_{h,h-1,...,1}(A_{original}) = Scr_h(Scr_{h-1}(...(Scr_1(A_{original}))...)) \tag{1}$$

where $Scr_k, k = 1, 2, ..., h$ are screening steps, and Scr_h is the final screening step in this sequential screening; $A_{original} = \{a_j | j \in N\}$ is the original set of decision attributes.

And two screening steps are conducted in proposed selection procedure, namely, $Scr_{2,1}(A) = Scr_2(Scr_1(A))$.

Scr_i is a screening step based on real-time attribute data, and the corresponding screening algorithm is established in the step. The result $Scr_i(A)$ is to obtain a relatively optimized set of decision attributes A_{Scr_1} by deleting redundant attributes.

Scr_2 is a further screening step based on the results of $Scr_2(A)$ to acquire the absolutely optimal set of decision attributes A_{Scr_2}. Decision goals that represent subjective preference are utilized in this step, and the selected set should satisfy all decision goals.

The whole procedure of attribute selection is shown by Figure 3

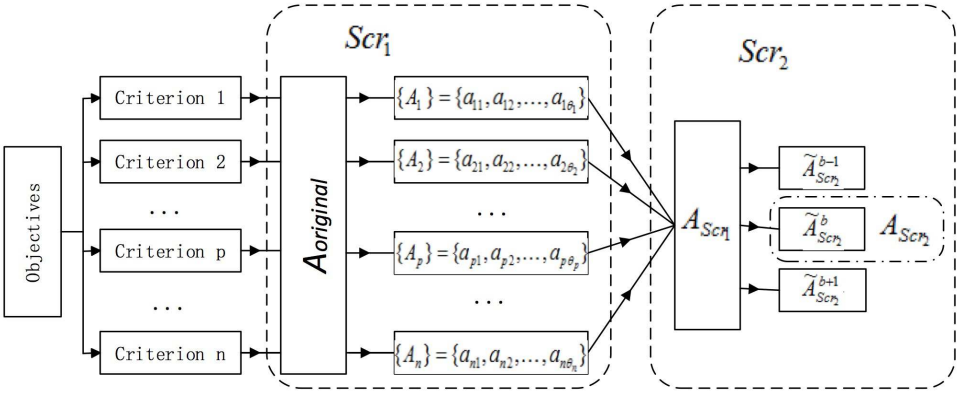


Figure 3: The procedure of the proposed method

where $A_{Scr_1} = \{\{A_1\}, \{A_1\}, ..., \{A_p\}, ..., \{A_n\}\}$, and $\widetilde{A}^b_{Scr_2}$ is candidate optimal set of Scr_2.

3.2 A new method for attribute selection

Preprocessing of the data

We consider three data types of decision attributes in this paper, namely, real numbers expressed as α^0, intervals expressed as $[\alpha^L, \alpha^U]$ and linguistic term set expressed as a linguistic term set $S = \{s_0, s_1, ..., s_K\}$. And real-time attribute data can be listed as attribute value series in Table 1.

where $c \le t_1 < t_2 < ... < t_m \le d$ and $[c, d]$ is a time interval.

Table 1: Attribute value series

	α_1	α_2	...	α_j	...
t_1	$\alpha_1(t_1)$	$\alpha_2(t_1)$	...	$\alpha_j(t_1)$	...
t_2	$\alpha_2(t_2)$	$\alpha_2(t_2)$	...	$\alpha_j(t_2)$	...
...					
t_i	$\alpha_i(t_i)$	$\alpha_2(t_i)$	...	$\alpha_j(t_i)$	...
...					
t_m	$\alpha_m(t_m)$	$\alpha_2(t_m)$	...	$\alpha_j(t_m)$	...

Both Scr_1 and Scr_2 are quantification procedures and therefore, in order to guarantee the scientificity and preciseness of the presented method, preprocessing of attribute data need to be carried out firstly (referring to our previous work [10]).

(1) Convert attribute values to TFNs

For a real number α^0, it can be denoted as a TFN $(\alpha^1, \alpha^2, \alpha^3, \alpha^4)$, where $\alpha^1 = \alpha^2 = \alpha^3 = \alpha^4 = \alpha^0$; for internal value $\lfloor \alpha^L, \alpha^U \rfloor$, the corresponding TFN can also be expressed as $(\alpha^1, \alpha^2, \alpha^3, \alpha^4)$, where $\alpha^1 = \alpha^2 = \alpha^L$ and $\alpha^3 = \alpha^4 = \alpha^U$. It is a little complex for a linguistic term set with odd cardinality $S = \{s_0, s_1, ..., s_K\}$, and the element s_K is converted to the corresponding TFN as follows:

$$(\alpha_k^1, \alpha_k^2, \alpha_k^3, \alpha_k^4) = \left(\max\left\{ \frac{2k-1}{2K+1}, 0 \right\}, \frac{2k}{2K+1}, \frac{2k+1}{2K+1}, \min\left\{ \frac{2k+2}{2K+1}, 1 \right\} \right) \tag{2}$$

where $k = 0, 1, ..., K$.

(2) Normalize values of attributes

In real-world decision scenarios, some decision attributes are cost ones, which mean the lower the values of attributes, the better they will be; the others are benefit ones, which mean the higher the values of attributes, the better they will be. Without loss of generality, we assume that the attribute values can be expressed as $\alpha_j(t_i) = (\alpha_j^1(t_i), \alpha_j^2(t_i), \alpha_j^3(t_i), \alpha_j^4(t_i))$, and normalized attribute values can be denoted as $\gamma_j(t_i) = (r_j^1(t_i), r_j^2(t_i), r_j^3(t_i), r_j^4(t_i))$. The normalized methods are given as follows:

- For cost attributes:

$$r_j^v(t_i) = \frac{\max\limits_i \left(\alpha_j^4(t_i) \right) - \alpha_j^v(t_i)}{\max\limits_i \left(\alpha_j^4(t_i) \right) - \min\limits_i \left(\alpha_j^1(t_i) \right)} \quad v = 1, 2, 3, 4. \tag{3}$$

- For benefit attributes:

$$r_j^v(t_i) = \frac{\alpha_j^v(t_i) - \min\limits_i \left(\alpha_j^1(t_i) \right)}{\max\limits_i \left(\alpha_j^4(t_i) \right) - \min\limits_i \left(\alpha_j^1(t_i) \right)} \quad v = 1, 2, 3, 4. \tag{4}$$

And the normalized attribute value series can be denoted as $\gamma_1 = \{r_1(t_1), r_1(t_2), ..., r_1(t_m)\}$;

$\gamma_2 = \{r_2(t_1), r_2(t_2), ..., r_2(t_m)\}$;
...;
$\gamma_j = \{r_j(t_1), r_j(t_2), ..., r_j(t_m)\}$;
....

The first step for screening Scr_1

In Scr_1, one "relatively best" representative attribute will be chosen by DMs for each criterion. It means that,

For $\forall C_p \in \{C_1, C_2, ..., C_n\}, 1 \leq p \leq n$,

$\exists a_{p1} \in A_{original}, st. C_p \Leftrightarrow a_{p1}$.

For example, "Economic losses" is a criterion in emergency decision, and "Amount of loss" is the "relatively best" attribute to evaluate loss degree. Thus the set $\{a_{p1}|1 \leq p \leq n\}$ is a significant by-product after criteria determination, and actually it is the input of Scr_1.

Scr_1 is used to screen decision-relevant attributes from $A_{original}$ by analyzing attribute interrelation. More precisely, we will investigate geometrical relationship between attributes based on attribute value series; select highly correlative attributes for a_{p1} from $A_{original}$ to constitute a "relatively best" representative set for each criterion C_i, and omit the rest of decision-irrelevant attributes. The step is entirely implemented on attribute value series without subjective impacts, and the outcome A_{Scr_1} is a smaller and relative optimal set which the elements are determined by evaluation criteria.

The idea of Scr_1 arises from Grey Relational Analysis (GRA) Theory which provides a valid way to quantify the interrelation of different factors using geometric methods [22]. And the detailed process of Scr_1 is given in the following.

Step 1.1: *Normalize attribute value series*

It is different from normalization of attribute values, and it guarantees comparability of different attribute value series. The specific process is

$$y_j = \{a_j(t_i)/D_j, i = 1, 2, ..., m\} \tag{5}$$

$$D_j = \frac{1}{m-1} \sum_{i=2}^{m} |a_j(t_i) - a_j(t_{i-1})| \tag{6}$$

where y_j is normalized attribute value series, and D_j is increment average of y_j.

Step 1.2: *Calculate increment series*

$$\Delta y_j = \{\Delta y_j(t_i) = y_j(t_i) - y_j(t_{i-1}), i = 2, 3, ..., m\} \tag{7}$$

Step 1.3: *Calculate correlation coefficient*

A_{Scr_1} is obtained based on the input set $\{a_{p1}|1 \leq p \leq n\}$ and thus, Step 1.3 will be stepwise conducted for each element a_{p1}. The correlation coefficient of different t_i between a_{p1} and a_j can be defined as follows:

$$\xi(t_i) = \begin{cases} sgn(\Delta y_{p1}(t_i) \cdot \Delta y_j(t_i)) \cdot \frac{\min(|\Delta y_{p1}(t_i)|, |\Delta y_j(t_i)|)}{\max(|\Delta y_{p1}(t_i)|, |\Delta y_j(t_i)|)}; & j \neq p1 \\ 0 & \Delta y_{p1}(t_i) \cdot \Delta y_j(t_i) = 0 \end{cases} \tag{8}$$

where $sgn(\cdot)$ is a sign function; and $sgn(\Delta y_{p1}(t_i) \cdot \Delta y_j(t_i)) = 1$, if $\Delta y_{p1}(t_i) \cdot \Delta y_j(t_i) > 0$; $sgn(\Delta y_{p1}(t_i) \cdot \Delta y_j(t_i)) = -1$, if $\Delta y_{p1}(t_i) \cdot \Delta y_j(t_i) < 0$.

Step 1.4: *Calculate correlation degrees*

The correlation degree between a_{p1} and a_j can be obtained by

$$\tau(a_{p1}, a_j) = \left| \frac{1}{d-c} \sum_{i=2}^{m} \Delta t_i \cdot \xi(t_i) \right| \tag{9}$$

The correlation degrees reflect the closeness of two attributes; the greater the value of $\tau(a_{p1}, a_j)$ is, the closer the two attributes are, and the better the capability of a_j is to evaluate criterion C_p.

Step 1.5: *Set threshold and screen correlation attributes for* a_{p1}

For $\forall C_p$, a threshold of correlation degree τ^* should be set by DMs, and the attribute a_j will be reserved, if $\tau(a_{p1}, a_j) > \tau^*$. And the reserved attributes for evaluating C_p can be denoted as $\{A_p\} = \{a_{p1}, a_{p2}, ..., a_{p\theta_p}\}$, where θ_p is the sum of attributes. In addition, the correlation degree $\tau(a_{p1}, a_{pq_p})$ between a_{p1} and a_{pq_p} will be abbreviated as z_{pq_p}, where $1 \leq q_p \leq \theta_p$.

Step 1.6: *Acquire the screening result set* A_{Scr_1}

Repeat Step 1.1 to Step 1.5 for each a_{p1}, where $1 \leq p \leq n$; and the result set A_{Scr_1} can be obtained, namely,

$A_{Scr_1} = \{\{A_1\}, \{A_n\}, ..., \{A_n\}\}$.

The second step based on multi-objective optimization Scr_2

Implementing decision attribute selection is aimed to assist DMs in evaluating alternatives for given decision problems and therefore, the selected attributes should be sensitive to the decision problem as well as familiar to DMs. Sensitivity reflects whether the selected attributes are accurate enough to represent the decision problem, and familiarity is used to estimate whether DMs are certain enough to make decision based on the selected attributes. For example, "fire duration" and "indoor fire load" are more important (sensitive) attributes than "carbon monoxide concentration" for evaluating fire grades; while it is more reliable for medical staff to estimate the physical condition of the wounded with "carbon monoxide concentration" than "Radiative Heat Flux".

Multi-objective optimization theory is a feasible way to quantify and synthesize these requirements, and it has solved similar problems in other areas successfully [1]. Thus three main objectives will be considered in this paper: attribute amount, attribute utility and attribute familiarity. Attribute amount should be as few as possible without reducing the validity of alternative, and it is the demand of both Scr_2 and attribute selection method. Attribute utility is used to distinguish attribute sensitivity in estimating the same decision problem, and attribute familiarity explores DMs' cognition differences for decision attributes. The outcome A_{Scr_2} drawn from Scr_2 should satisfy mentioned three goals, and now the optimal set of decision attributes have been found after Scr_1 and Scr_2, namely, $A_{Scr_2} = Scr_2(Scr_1(A_{original}))$.

A set of zero-one binary vectors $\{v^b\}$ with $T = (\theta_1 + \theta_2 + ... + \theta_n)$ dimensionality will be defined to represent different candidate outcome set $\widetilde{A}^b_{Scr_1}$, where one represents the attribute belonging to the set, otherwise it is zero. The optimal set of decision attributes is $A_{Scr_2} \in \{\widetilde{A}^b_{Scr_1} | b \in N\}$, and $\{\widetilde{A}^b_{Scr_1}\}$ contains all the subsets of A_{Scr_1}.

Three objective functions are defined firstly, and we will show that all the objective functions can be expressed as the functions with independent variable v^b.

Defining objective functions (1) Attribute utility function

Attribute utility function measures decision sensitivity of $\widetilde{A}^b_{Scr_1}$, and it can be defined as follows:

$$U(\widetilde{A}^b_{Scr_1}) = g(MU(\widetilde{A}^b_{Scr_1}), W(\widetilde{A}^b_{Scr_1})) \tag{10}$$

where

$$\textcircled{1}\ MU(\widetilde{A}^b_{Scr_1}) = v^b \cdot MU^* \tag{11}$$

MU^* represents attribute utility matrix, and it is diagonal matrix with the diagonal elements $\frac{\Delta a_{11}}{\Delta a_{11}}, ..., \frac{\Delta a_{11}}{\Delta a_{1\theta_1}}, ..., \frac{\Delta a_{p1}}{\Delta a_{pq_p}}, ..., \frac{\Delta a_{n1}}{\Delta a_{n1}}, ..., \frac{\Delta a_{n1}}{\Delta a_{n\theta_n}}$.

More precisely, MU^* can be explained by a marginal utility function $u(\cdot)$, namely,

$$u(a_{pq_p}) = \frac{\Delta a_{p1}}{\Delta a_{pq_p}} \qquad 1 \le p \le n, 1 \le q_p \le \theta_p; \tag{12}$$

where Δa_{p1} is increment of a_{p1}, and Δa_{pq_p} is the corresponding varying of a_{pq_p}. And the function $u(\cdot)$ quantifies the assessment performance on criterion C_i of the attribute a_{pq_p} according to calculate the ratio between increments of the two attribute values.

In fact, MU^* can be predetermined by domain experts, because plenty of field studies have been conducted to investigate utility relation of attributes.

$$\textcircled{2}\ W(\widetilde{A}^b_{Scr_1}) = W^* \cdot (v^b)' \tag{13}$$

W^* represents attribute weighting matrix, and it is also a diagonal matrix with diagonal elements $w_{11}, ..., w_{1\theta_1}, ..., w_{pq_p}, ..., w_{n1}, ..., w_{n\theta_n}$.

where

$$w_{pq_p} = \omega_p \cdot z_{pq_p}, \quad 1 \le p \le n, 1 \le q_p \le \theta_p; \tag{14}$$

and $\{\omega_1, \omega_2, ..., \omega_n\}$ are the know weights of $C = \{C_1, C_2, ..., C_n\}$.

$g(\cdot)$ is a dot product function and therefore, $U(\widetilde{A}^b_{Scr_1})$ can be transformed into the following function with independent variable v^b, namely,

$$U(v^b) = (v^b \cdot MU^*) \cdot (W^* \cdot (v^b)') \tag{15}$$

(2) Attribute familiarity function

Attribute familiarity function is established to quantify DMs' cognition degrees on different candidate set $\widetilde{A}^b_{Scr_1}$, and it is defined as

$$F(v^b) = \eta \cdot \sigma \cdot (v^b)' \tag{16}$$

In Eq. 16, $\eta = (\eta_1, \eta_2, ..., \eta_L)_{1\times L}$ is the weight vector of DMs, and L is the number of DMs; σ is attribute familiarity matrix with the following expression

$$\sigma = \begin{pmatrix} \sigma_1(1) & \sigma_2(1) & ... & \sigma_T(1) \\ \sigma_1(2) & \sigma_2(2) & ... & \sigma_T(2) \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \sigma_1(L) & \sigma_2(L) & ... & \sigma_T(L) \end{pmatrix}_{L\times T} \tag{17}$$

and $\sigma_h(l)$ represents the lth DM's familiarity degree with attribute a_h, where $1 \le l \le L$, and $1 \le h \le T$. Attribute familiarity is usually evaluated qualitatively by DMs, and many methods can be applied to quantify assessment results and obtain matrix σ, such as AHP and fuzzy set theory.

(3) Attribute count function

Attribute count function calculates the attribute number of a candidate set $\widetilde{A}^b_{Scr_1}$, and it can be defined by means of vector length formula,

$$Count(\widetilde{A}^b_{Scr_1}) = |v^b|^2 \tag{18}$$

where $|v^b|$ represents the length of vector v^b.

Establish multi-objective optimization model for Scr_2 After define relevant objective functions, the multi-objective optimization model of Scr_2 is established to obtain the Pareto Optimal solutions A_{Scr_2}, as well as the optimal set of decision attribute selection problem.

$$\begin{aligned} Maximize \quad & U(v^b) = (v^b \cdot MU^*) \cdot (W^* \cdot (v^b)') \\ & F(v^b) = \eta \cdot \sigma \cdot (v^b)' \end{aligned} \tag{19}$$

$$Minimize \quad Count(\widetilde{A}^b_{Scr_1}) = |v^b|^2$$

$$Subject\ to: \quad |v^b|^2 \geq n$$

The model Eq. 19 can be solved by aforementioned algorithm, such as MOGA, NSGA, MOMS-GA and traditional mathematical programming methods, and it depends the specific model drawn from given decision problems and decision requirements.

4 Empirical results

4.1 Accident scenario and computational settings

In order to validate the presented method, a fire accident scenario will be constructed and simulated. The fire happens in a factory dormitory, and two workers are asleep; some inflammable are stacked near the door which arise spontaneous combustion for some reasons. The total layout of this room is illustrated in Figure 4, containing three beds, two workers marked by fellow cuboids, fire origin and plenty of fire smoke. Now two experts need to make rescue alternative based on the collected data; one is fire expert who are engaged in fire research for many years and have in-depth study of the fire attributes, and the other is an experienced firefighter who know some medical knowledge. And they are weighted with 0.6 and 0.4 respectively. A fire simulation software FDS [17] is used to establish fire scenario and generate fire data.

The purpose of simulation is not only to provide a fire scenario, and more importantly, it can generate relevant fire attributes data for the following experiment. The data can be exported from FDS containing attribute names, physical dimensions, attribute values and attribute types. In this paper, we generate twelve fire attributes and 125 data records.

4.2 Select fire attributes by utilizing the presented method

In this fire scenario, three main criteria will be considered, i.e., fire behavior, personal security and environmental conditions, and corresponding attributes are shown in Table 5

where "Enthalpy" is an interval attribute, "Indoor fire load" and "Circuit aging degree" are linguistic attributes, and the rest are real number attributes. The bold attributes are "relatively best" representative attributes for each criterion.

(1) Data preprocessing

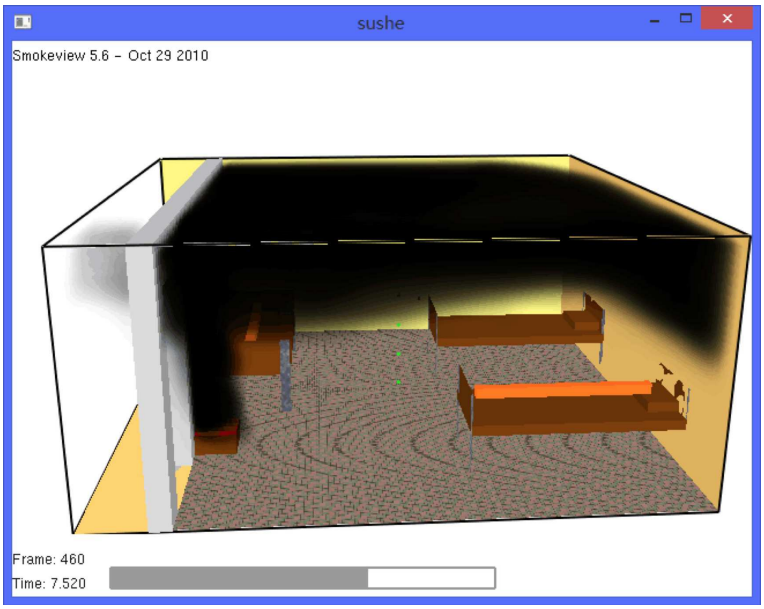


Figure 4: A simulated fire scenario

Table 2: List of criteria and corresponding attributes

Criterion	Attributes
Fire behavior	**Temperature** / Enthalpy / Heat Release Rate / Relative Humidity / Burning Rate / Pressure / Radiative Heat Flux
Personal security	**Carbon monoxide** / Carbon dioxide / oxygen
Environmental conditions	**Indoor fire load** / Circuit aging degree

As aforementioned, the real number attributes and interval attributes can be denoted by TFNs directly. The linguistic attributes "Flammable of storage items" and "Circuit aging degree" can be transformed into TFNs in Table 3.

Table 3: Linguistic attributes and corresponding TFNs

Linguistic terms	TFNs
Not dangerous	(0,0,0.111,0.222)
Medium dangerous	(0.111,0.222,0.333,0.444)
Dangerous	(0.333,0.444,0.555,0.666)
Very dangerous	(0.555,0.666,0.777,0.888)
Highly dangerous	(0.777,0.888,0.999,1)

Then we carry out normalization by Eq. 3 and Eq. 4 to obtain corresponding TFNs.

(2) The first step for screening Scr_1

Without loss of generality, the criterion "Fire behavior" will be utilized to illustrate the first screening procedure and the other two criteria "Personal security" and "Environmental conditions" can be dealt with similarly.

The criterion "Fire behavior" is represented by seven attributes, namely, "Temperature", "Enthalpy", "Heat Release Rate", "Relative Humidity", "Burning Rate", "Pressure" and "Radiative Heat Flux"; and "Temperature" is the "relatively best" representative attribute for the criterion "Fire" behavior.

Step 1.1 to *Step 1.4* are implemented to calculate correlation degrees between ""Temperature" and the remaining six attributes, and the results are given in Table 4.

Table 4: Correlation degrees between "Temperature" and the other six attributes

Attribute	Enthalpy	Heat Release Rate	Relative Humidity	Burning Rate	Pressure	Radiative Heat Flux
Correlation degree	0.53745	0.96987	0.33271	0.74787	0.47665	0.88712

Step 1.5: Set threshold of correlation degree $\tau^* = 0.6$

Thus the attributes will be reserved whose correlation degrees is greater than or equal to 0.6, i.e., "Burning Rate", "Heat Release Rate" and "Radiative Heat Flux".

Step 1.6: Repeat Step 1.1 to Step 1.5 for each criterion

For criteria "Personal security" and "Environmental conditions", the "relatively best" representative attributes are respectively "Carbon monoxide" and "Indoor fire load", and relevant correlation degrees are shown in Table 5.

Table 5: Relevant correlation degrees for the other two criteria

Criterion	Personal security		Environmental conditions
Attribute	Carbon dioxide	oxygen	Circuit aging degree
Correlation degree	0.80961	0.93428	0.76865

We reset thresholds of correlation degree for each criteria, viz., $\tau^*_{Personal} = 0.9$ and $\tau^*_{Environmental} = 0.8$.

Finally, the result of the first screening A_{Scr_1} can be obtained, namely,

A_{Scr_1} = {"Temperature", "Heat Release Rate", "Radiative Heat Flux", "Burning Rate", "Carbon monoxide", "oxygen", "Indoor fire load"}.

(3) The second step for screening Scr_2

Before conducting Scr_2, MU^*, σ and W^* should be given firstly. As aforementioned, MU^* and σ can be determined by relevant research achievements and expert experiences, and therefore, we collected and integrated evaluation results from ten fire researchers to assign MU^* and σ as follows:

$$MU^* = \begin{pmatrix} 19.5 & & & & & & \\ & 15.6 & & & & & \\ & & 7.4 & & & & \\ & & & 11.2 & & & \\ & & & & 15.4 & & \\ & & & & & 18.7 & \\ & & & & & & 18.2 \end{pmatrix} \text{ and } \sigma = \begin{pmatrix} 9 & 8 & 8 & 9 & 4 & 4 & 5 \\ 4 & 5 & 4 & 4 & 9 & 9 & 7 \end{pmatrix}$$

where the elements values of MU^* are in the interval [0, 20], and the greater the value is, the more sensitive the attribute is for evaluation. The elements values of σ are in the interval [0, 10], and the greater the value is, the more familiar the attribute is to DMs.

Moreover, W^* can be acquired with the given criteria weight vector $(\omega_1, \omega_2, \omega_3) = (0.4, 0.4, 0.2)$, namely,

$$W^* = \begin{pmatrix} 0.299 & & & & & & \\ & 0.388 & & & & & \\ & & 0.215 & & & & \\ & & & 0.354 & & & \\ & & & & 0.352 & & \\ & & & & & 0.374 & \\ & & & & & & 0.197 \end{pmatrix}$$

After determine relevant parameters MU^*, W^*, σ, and the seven-dimensional independent variable v^b, Scr_2 based on the multi-objective optimization model Eq 19 is performed in Matlab R2013b. We choose the traditional mathematical programming method to solve this model, because the research [15] has proven that the programming method has better performance than aforementioned other algorithms to solve such a problem with the zero-one binary vector variable and the small size of data.

The results are shown in Figure 5 where the curve represents all possible solutions and the Pareto Optimal solution is the intersection of the curve and the x-axis, namely, $v^b = (1, 1, 0, 1, 1, 1, 1)$. It means that the attribute "Radiative Heat Flux" represented by zero can be omitted, and the optimal set of decision attributes is:

$A_{Scr2} = \{$"Temperature", "Heat Release Rate", "Burning Rate", "Carbon monoxide", "oxygen", "Indoor fire load"$\}$.

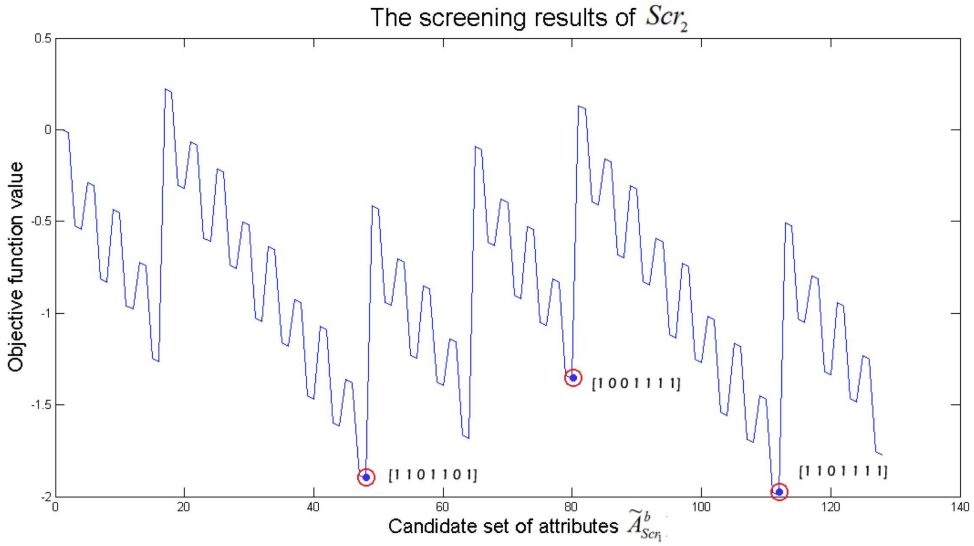


Figure 5: The results of the second step Scr_2

4.3 Validation of the presented method

In order to verify the result is the optimal set for a MCDM problem, multi-attribute utility theory will be used in the following.

A conclusion underlies the presented method: Removing decision-irrelevant attributes from the set of decision attributes will improve the effectiveness of decision alternatives. Thus it is reasonable and feasible to verify our work starting from Scr_2, because the purpose of Scr_1 is to omit decision-irrelevant attributes.

Multi-attribute utility theory (MAUT) is used to choose decision alternatives by evaluating the utility values of alternatives which is calculated based on the utility of decision attributes, and it is an effective way to quantify decision attributes' impact on decision results. More precisely, the utility evaluation model of an alternative is given as follows:

$$u(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} k_i u_i(x_i) \qquad if \ \sum_{i=1}^{n} k_i = 1 \tag{20}$$

where x_i is the ith decision attribute, and k_i is the corresponding weight; $u_i(x_i)$ is the attribute utility function of x_i, and $u(x_1, x_2, ..., x_n)$ is the utility function of an alternative evaluated by $x_1, x_2, ..., x_n$.

Thus the utility values of all possible candidate sets $\widetilde{A}^b_{Scr_1}$ are calculated based on Eq. 20, where attribute utility function $u_i(x_i)$ can be defined as Eq. 12, and the weights are acquired by Eq. 14. It should be pointed that normalization of weights need be carried out to guarantee $\sum_{i=1}^{n} k_i = 1$. And the results are given in Table 6, where S_i represents the candidate set of decision attributes, and $1 \leq i \leq 16$. The correspondence between S_i and the set of attributes are shown in Table 7.

Table 6: The utility values of attribute sets

Attribute set	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
Utility value	23.621	24.343	28.496	28.923	29.665	31.745	28.474	29.500
Attribute set	S_9	S_{10}	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}
Utility value	25.908	28.447	32.631	31.229	30.335	35.774	31.658	33.439

Table 7: The correspondence between S_i and the set of attributes

S_i	The correspondence set	S_i	The correspondence set
S_1	(1, 0, 0, 0, 1, 0, 1)	S_9	(1, 0, 0, 0, 1, 1, 1)
S_2	(1, 1, 0, 0, 1, 0, 1)	S_{10}	(1, 1, 0, 0, 1, 1, 1)
S_3	(1, 0, 1, 0, 1, 0, 1)	S_{11}	(1, 0, 1, 0, 1, 1, 1)
S_4	(1, 0, 0, 1, 1, 0, 1)	S_{12}	(1, 0, 0, 1, 1, 1, 1)
S_5	(1, 1, 1, 0, 1, 0, 1)	S_{13}	(1, 1, 1, 0, 1, 1, 1)
S_6	(1, 1, 0, 1, 1, 0, 1)	S_{14}	(1, 1, 0, 1, 1, 1, 1)
S_7	(1, 0, 1, 1, 1, 0, 1)	S_{15}	(1, 0, 1, 1, 1, 1, 1)
S_8	(1, 1, 1, 1, 1, 0, 1)	S_{16}	(1, 1, 1, 1, 1, 1, 1)

In Table 6, for $\forall i$, $1 \leq i \leq 16$, it has $u(S_{14}) > u(S_i)$, where $i \neq 14$. The set S_{14} has the maximum utility value $u(S_{14}) = 35.774$, and thus the alternative chosen based on S_{14} (the optimal set A_{Scr_2}) is the optimal alternative.

In addition, S_1 contains three attributes with minimum utility value $u(S_1) = 23.621$, and S_{16} contains all attributes with utility value $u(S_{16}) = 33.439$. It means that too few or too many

attributes are not available in a MCDM problem; too few attributes are not enough to evaluate the decision problem, while too many attributes may lead to conflict and interfere decision results.

The set of decision attributes obtained by utilizing the presented method has the maximum utility value in alternative evaluation, and it demonstrates the effectiveness of the presented method.

5 Implications

This paper utilizes attribute selection procedure in MCDM problems innovatively. MCDM problems refer to getting the optimal alternatives which are determined based on many qualitative or quantitative criteria, and these criteria are conflicting and assessed by more relevant attributes. Therefore, the set of decision attributes determines the effectiveness of an alternative. In order to improve the reliability of MCDM, the method proposed in this study can select the optimal set of decision attributes which are highly related to the MCDM problem and remove redundant or "noisy" attributes. Meanwhile, the selection of optimal set can reduce the attribute dimension and improve the efficiency of decision making while maintaining the great effect.

The method combines external attribute data with subjective decision preferences in the sequential procedure to improve decision performance. External attribute data are objective factors, which show the features of the data. These attributes contain its own regularity and all kinds of information that decision-making needs, so the first step to screen is based on objective attribute data. Subjective factors show the decision makers' preferences. Because of the different profession and background knowledge, decision makers have different recognition of attributes. Considering the subjective decision preferences contributes to improve the reliability and accuracy of decision-making. Using the former to build the method and using the constraint conditions of the latter to adjust the method can make it more in line with the actual situation.

The method combines external attribute data with subjective decision preferences in the sequential procedure to improve decision performance. External attribute data are objective factors, which show the features of the data. These attributes contain its own regularity and all kinds of information that decision-making needs, so the first step to screen is based on objective attribute data. Subjective factors show the decision makers" preferences. Because of the different profession and background knowledge, decision makers have different recognition of attributes. Considering the subjective decision preferences contributes to improve the reliability and accuracy of decision-making. Using the former to build the method and using the constraint conditions of the latter to adjust the method can make it more in line with the actual situation.

6 Conclusions

In this paper, a new attribute selection method with the context of MCDM is proposed. According to the analysis of attribute selection problem, a two-step procedure is established to reduce original set to the optimal set of decision attributes. GRA theory and multi-objective optimization method are respectively used to implement these two screening steps. And then a fire example is shown to illustrate application and validity of screening method. The attribute selection method presented in this paper is a dynamic and flexible procedure which depends on decision data resource and decision requirement. It eliminates the drawback that decision attributes are chosen completely subjectively, and it will show better performance for new decision scenarios.

Attribute selection is also a critical process for many domains, such as classification problem, clustering problem, risk assessment, credit assessment, etc. And even solution of these issues

depended on results of attribute selection. Attribute selection problem is also discussed in data mining called "Feature Engineering", and still has needs to continue thorough research and the technique problem to be solved.

Generally it exists plenty of missing data among decision information, and how to select the optimal decision attributes in this case will be our future research.

Bibliography

[1] Babaei, S., Sepehri, M. M., Babaei, E. (2015). Multi-objective portfolio optimization considering the dependence structure of asset returns. *European Journal of Operational Research*, 244(2), 525-539, 2015.

[2] Bermejo, P., Gamez, J. A., Puerta, J. M. (2014). Speeding up incremental wrapper feature subset selection with Naive Bayes classifier. *Knowledge-Based Systems*, 55, 140-147, 2014.

[3] Chen, Y., Kilgour, D. M., Hipel, K. W. (2008). Screening in multiple criteria decision analysis. *Decision Support Systems*, 45(2), 278-290, 2008.

[4] Chun, Y. H. (2015). Multi-attribute sequential decision problem with optimizing and satisficing attributes. *European Journal of Operational Research*, 243(1), 224-232, 2015.

[5] Comes, T., Hiete, M., Wijngaards, N., Schultmann, F. (2011). Decision maps: A framework for multi-criteria decision support under severe uncertainty. *Decision Support Systems*, 52(1), 108-118, 2011.

[6] Dai, J., Wang, W., Tian, H., Liu, L. (2013). Attribute selection based on a new conditional entropy for incomplete decision systems. *Knowledge-Based Systems*, 39, 207-213, 2013.

[7] Hapfelmeier, A., Ulm, K. (2014). Variable selection by Random Forests using data with missing values, *Computational Statistics & Data Analysis*, 80, 129-139, 2014.

[8] Huda, S., Abdollahian, M., Mammadov, M., Yearwood, J., Ahmed, S., Sultan, I. (2014): A hybrid wrapper-filter approach to detect the source (s) of out-of-control signals in multivariate manufacturing process, *European Journal of Operational Research*, 237(3), 857-870, 2014.

[9] Lin, Q., Li, J., Du, Z., Chen, J., Ming, Z. (2015). A novel multi-objective particle swarm optimization with multiple search strategies, *European Journal of Operational Research*, 247(3), 732-744, 2015.

[10] Ma XF, Zhong QY, Qu Y (2013) Determination method of emergency key property based on common knowledge model and Euclidean distance, *Systems Engineering*, 31(10), 93-97, 2013.

[11] Meinshausen, N., Buhlmann, P. (2010): Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4), 417-473, 2010.

[12] Meng MH, Pei XJ, Wu MQ (2015): Study on choice of factors influencing stability of perilous rock based on fuzzy multi-attribute group decision-making. *Subgrade Engineering*, 1:20-23, 2015.

[13] Montajabiha, M. (2016): An Extended PROMETHE II Multi-Criteria Group Decision Making Technique Based on Intuitionistic Fuzzy Logic for Sustainable Energy Planning. *Group Decision and Negotiation*, 25(2), 221-244, 2016.

[14] Robin, G., Jean-Michel P., Christine T. (2010): Variable selection using random forests, *Pattern Recognition Letters*, 31, 2225-2236, 2010.

[15] Shen HP, Zhang YP, Wang YK (2014): Research on regular Chinese fragments reassembly based on 0-1 programming model, *Electronic Science and Technology*, 6:13-16, 2014.

[16] Stewart, T. J. (1992): A critical survey on the status of multiple criteria decision making theory and practice, *Omega*, 20(5-6), 569-586, 1992.

[17] Wahlqvist, J., Van Hees, P. (2013): Validation of FDS for large-scale well-confined mechanically ventilated fire scenarios with emphasis on predicting ventilation system behavior, *Fire Safety Journal*, 62, 102-114, 2013.

[18] Wu, K. J., Tseng, M. L., Chiu, A. S., Lim, M. K. (2016): Achieving competitive advantage through supply chain agility under uncertainty: A novel multi-criteria decision-making structure, *International Journal of Production Economics*, article in press, 2016.

[19] Wu, K. J., Liao, C. J., Tseng, M. L., Lim, M. K., Hu, J., Tan, K. (2017): Toward sustainability: using big data to explore the decisive attributes of supply chain risks and uncertainties, *Journal of Cleaner Production*, 142, 663-676, 2017.

[20] Zeleny, M., Cochrane, J. L. (1973): *Multiple criteria decision making*, University of South Carolina Press, 1973.

[21] Zhang, Y., Gong, D., Cheng, J. (2017): Multi-Objective Particle Swarm Optimization Approach for Cost-based Feature Selection in Classification, *IEEe/ACM Transactions on Computational Biology and Bioinformatics*, 14, 64-75, 2017.

[22] Zhu, J., Zhang, S., Chen, Y., Zhang, L. (2016): A hierarchical clustering approach based on three- dimensional gray relational analysis for clustering a large group of decision makers with double information, *Group Decision and Negotiation*, 25(2), 325-354, 2016.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 408-428, June 2018.

Latent Semantic Analysis using a Dennis Coefficient for English Sentiment Classification in a Parallel System

V.N. Phu, V.T.N. Tran

Vo Ngoc Phu*
Institute of Research and Development,
Duy Tan University-DTU
Da Nang, Vietnam
*Corresponding author: vongocphu03hca@gmail.com; vongocphu@dtu.edu.vn

Vo Thi Ngoc Tran
School of Industrial Management (SIM),
Ho Chi Minh City University of Technology - HCMUT,
Vietnam National University
Ho Chi Minh City, Vietnam
vtntran@hcmut.edu.vn

Abstract: We have already survey many significant approaches for many years because there are many crucial contributions of the sentiment classification which can be applied in everyday life, such as in political activities, commodity production, and commercial activities. We have proposed a novel model using a Latent Semantic Analysis (LSA) and a Dennis Coefficient (DNC) for big data sentiment classification in English. Many LSA vectors (LSAV) have successfully been reformed by using the DNC. We use the DNC and the LSAVs to classify 11,000,000 documents of our testing data set to 5,000,000 documents of our training data set in English. This novel model uses many sentiment lexicons of our basis English sentiment dictionary (bESD). We have tested the proposed model in both a sequential environment and a distributed network system. The results of the sequential system are not as good as that of the parallel environment. We have achieved 88.76% accuracy of the testing data set, and this is better than the accuracies of many previous models of the semantic analysis. Besides, we have also compared the novel model with the previous models, and the experiments and the results of our proposed model are better than that of the previous model. Many different fields can widely use the results of the novel model in many commercial applications and surveys of the sentiment classification.
Keywords: English sentiment classification; parallel system; Cloudera; Hadoop Map and Hadoop Reduce; Dennis Measure; Latent Semantic Analysis

1 Introduction

In this survey, our novel model is performed as follows: Firstly, we use the Dennis Coefficient (DNC) to identify the valences and polarities of the sentiment lexicons of the basis English sentiment dictionary (bESD). Then, the Latent Semantic Analysis (LSA) is improved by using the sentiment lexicons. All the positive documents of the training data set are transferred into one LSAV, called the positive LSAV group. All the negative documents of the training data set are transferred into one LSAV, called the negative LSAV group. Each document in the documents of the testing data set is transferred into one LSAV. We use the DNC to cluster this LSAV into either the positive LSAV group or the negative LSAV group. One similarity measure between the LSAV and the positive LSAV group is calculated certainly, called Measure_1 and one similarity measure between the LSAV and the negative LSAV group is calculated certainly, called Measure_2. If Measure_1 is greater than Measure_2, it means that the LSAV being close to the positive LSAV group is greater than the LSAV being close to the negative LSAV group, so

the LSAV (corresponding to the document of the testing data set) is clustered into the positive. If Measure_1 is less than Measure_2, it means that the LSAV being close to the positive LSAV group is less than the LSAV being close to the negative LSAV group, so the LSAV (corresponding to the document of the testing data set) is clustered into the negative. If Measure_1 is as equal as Measure_2, it means that the LSAV being close to the positive LSAV group is as equal as the LSAV being close to the negative LSAV group, so the LSAV (corresponding to the document of the testing data set) is not clustered into both the positive and the negative. The LSAV is clustered into the neutral polarity. Therefore, the sentiment classification of this document is identified successfully. Finally, the sentiment classification of all the document of the testing data set is identified fully.

We firstly implement all the above things in the sequential system, and then, we perform all the above things in the parallel network environment to shorten the execution times of the proposed model.

Our proposed model has the crucial contributions to many areas of research as well as commercial applications as follows: (1) Many surveys and commercial applications can use the results of this work in a significant way; (2) The algorithms are built in the proposed model; (3) This survey can certainly be applied to other languages easily; (4) The algorithm of data mining is applicable to semantic analysis of natural language processing; (5) Millions of English documents are successfully processed for emotional analysis; (6) Our proposed model can be applied to many different parallel network environments such as a Cloudera system; (7) This study can be applied to many different distributed functions such as Hadoop Map (M) and Hadoop Reduce (R); (8) The LSA − related algorithms are proposed in this survey; (9) The DNC − related algorithms are built in this work.

2 Related work

We summarize many researches which are related to our research. The surveys related the similarity coefficients to calculate the valences of words are in [1, 13–18]. In the research [1], the authors generate several Norwegian sentiment lexicons by extracting sentiment information from two different types of Norwegian text corpus, namely, news corpus and discussion forums. The methodology is based on the Point wise Mutual Information (PMI), etc.

The English dictionaries are [19–21] and there are more than 55,000 English words (including English nouns, English adjectives, English verbs, etc.) from them.
There are the works related to the Dennis Coefficient (DNC) in [2, 3, 5]. The authors in [3] collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique, etc.

There are the researches related the Latent Semantic Analysis (LSA) in [4, 6, 7]. The study in [4] presents a novel statistical method for factor analysis of binary and count data which is closely related to a technique known as Latent Semantic Analysis, etc.

The latest researches of the sentiment classification are [8–12]. In the research [9], the authors have explored different methods of improving the accuracy of sentiment classification. The authors' proposed method based on the combination of TermCounting method and Enhanced Contextual Valence Shifters method has improved the accuracy of sentiment classification, etc.

3 Methodology

Our methodology comprises 3 sub-sections as follows: (1) First sub-section: Creating the sentiment lexicons of the bESD; (2) Second sub-section: Improving the LSA according to the

sentiment lexicons of the bESD a sequential environment and a distributed network system; (3) Third sub-section: Using the LSA and a DNC to cluster the documents of the testing data set into either the positive or the negative in both a sequential environment and a parallel distributed system.

We built our the testing data set including the 11,000,000 documents in the movie field, which contains the 5,500,000 positive and 5,500,000 negative in English. We also built our the training data set including the 5,000,000 documents in the movie field, which contains the 2,500,000 positive and 2,500,000 negative in English. All the English documents in our testing data set and training data set are automatically extracted from millions of the documents of English Facebook, English websites and social networks; then we labeled positive and negative for them.

3.1 Creating the sentiment lexicons of the bESD

Calculating a valence of one word (or one phrase) in English

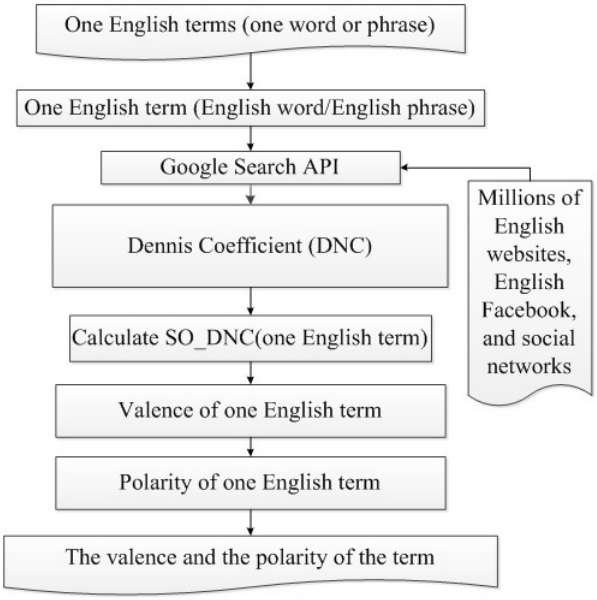


Figure 1: Overview of identifying the valence and the polarity of one term in English using a Dennis coefficient (DNC)

In this part, we calculated the valence and the polarity of one English word (or phrase) by using the DNC through a Google search engine with AND operator and OR operator, as the following diagram in Figure 1 shows.
According to [1, 13–18], Pointwise Mutual Information (PMI) between two words wi and wj has the equation

$$PMI(w_i, w_j) = \log_2 \left[\frac{P(w_i, w_j)}{P(w_i) \times P(w_j)} \right] \tag{1}$$

and SO (sentiment orientation) of word wi has the equation

$$SO(w_i) = PMI(w_i, positive) - PMI(w_i, negative) \tag{2}$$

In the research [1], the authors generate several Norwegian sentiment lexicons by extracting sentiment information from two different types of Norwegian text corpus, namely, news corpus and discussion forums. The methodology is based on the Point wise Mutual Information (PMI),

etc. The authors in [13] used the Ochiai Measure through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in Vietnamese. The authors in [14] used the Consine Measure through the Google search engine with AND operator and OR operator to identify the sentiment scores of the words in English. The authors in [15] used the Sorensen Coefficient through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in English. The authors in [16] used the Jaccard Measure through the Google search engine with AND operator and OR operator to calculate the sentiment values of the words in Vietnamese. The authors in [17] used the Tanimoto Coefficient through the Google search engine with AND operator and OR operator to identify the sentiment scores of the words in English.

With the above proofs, we had the information about the measures as follows: PMI was used with AltaVista in English, Chinese, and Japanese with the Google in English; Jaccard was used with the Google in English, Chinese, and Vietnamese. The Ochiai was used with the Google in Vietnamese. The Consine and Sorensen were used with the Google in English.

According to [1,13–18], PMI, Jaccard, Consine , Ochiai, Sorensen, Tanimoto, and DNC were the similarity measures between two words, and they can perform the same functions and with the same chracteristics; so DNC was used in calculating the valence of the words. In addition, we proved that DNC can be used in identifying the valence of the English word through the Google search with the AND operator and OR operator.

With the Dennis coefficient (DNC) in [2, 3, 5], we had the equation of the DNC:

$$DNC(a,b) = \frac{[(a \cap b) \times [(\neg a) \cap (\neg b)] - [(\neg a) \cap b] \times [a \cap (\neg b)]]}{\sqrt{n \times [(a \cap b) + [(\neg a) \cap b]] \times [(a \cap b) + [a \cap (\neg b)]]}} \tag{3}$$

with a and b are the vectors.

In this study, we chose n=1. Therefore, we had eq. (4) as follows:

$$DNC(a,b) = \frac{[(a \cap b) \times [(\neg a) \cap (\neg b)] - [(\neg a) \cap b] \times [a \cap (\neg b)]]}{\sqrt{[(a \cap b) + [(\neg a) \cap b]] \times [(a \cap b) + [a \cap (\neg b)]]}} \tag{4}$$

From the eq. (1), (2), (3), we proposed many new equations of the DNC to calculate the valence and the polarity of the English words (or the English phrases) through the Google search engine as the following equations below. In eq. (3), when a had only one element, a is a word. When b had only one element, b is a word. In eq. (3), a was replaced by w1 and b was replaced by w2.

$$DennisMeasure(w_1, w_2) = DennisCoefficient(w_1, w_2) = DNC(w_1, w_2) = \frac{B}{\sqrt{A}} \tag{5}$$

with
(1) B=P(w_1,w_2)× P($\neg w_1$,$\neg w_2$)-P($\neg w1$,w_2) × P(w_1,$\neg w_2$);
(2) A=[P(w_1,w_2)+P($\neg w_1$,w_2)]× [P(w_1,w_2)+P(w_1,$\neg w_2$)].

Eq. (5) was similar to eq. (1). In eq. (2), eq. (1) was replaced by eq. (4). We had eq. (6) as follows:

$$Valence(w) = SO\_DNC(w) = DNC(w, positive\_query) - DNC(w, negative\_query) \tag{6}$$

In eq. (5), w_1 was replaced by w and w_2 was replaced by position_query. We had eq. (7). Eq. (7) was as follows:

$$DNC(w, positive\_query) = \frac{B7}{\sqrt{A7}} \tag{7}$$

with

(1) B7=P(w,positive_query) × P($\neg w$, $\neg positive\_query$)-P($\neg w$,positive_query) × P(w,$\neg positive\_query$);

(2) A7=[P(w,positive_query)+P($\neg w$,positive_query)] × [P(w,positive_query)+P(w,$\neg positive\_query$)].

In eq. (5), w_1 was replaced by w and w_2 was replaced by negative_query. We had eq. (8). Eq. (8) was as follows:

$$DNC(w, negative\_query) = \frac{B8}{\sqrt{A8}} \tag{8}$$

with:

(1) B8=P(w,negative_query) × P($\neg w$,$\neg negative\_query$)-P($\neg w$,negative_query) × P(w,$\neg negative\_query$);

(2) A8=[P(w,negative_query)+P($\neg w$,negative_query)] × [P(w,negative_query)+P(w,$\neg negative\_query$)].

We had the information about w, w_1, w_2, and etc. as follows:

(1) w, w_1, w_2 : were the English words (or the English phrases);

(2) P(w_1, w_2): number of returned results in Google search by keyword (w_1 and w_2). We use the Google Search API to get the number of returned results in search online Google by keyword (w_1 and w_2);

(3) P(w_1): number of returned results in Google search by keyword w_1. We use the Google Search API to get the number of returned results in search online Google by keyword w_1;

(4) P(w_2): number of returned results in Google search by keyword w_2. We use the Google Search API to get the number of returned results in search online Google by keyword w_2;

(5) Valence(W) = SO_DNC(w): valence of English word (or English phrase) w; is SO of word (or phrase) by using the Dennis coefficient (DNC);

(6) positive_query: active or good or positive or beautiful or strong or nice or excellent or fortunate or correct or superior with the positive_query is the a group of the positive English words;

(7) negative_query: passive or bad or negative or ugly or week or nasty or poor or unfortunate or wrong or inferior with the negative_query is the a group of the negative English words;

(8) P(w, positive_query): number of returned results in Google search by keyword (positive_query and w). We used the Google Search API to get the number of returned results in search online Google by keyword (positive_query and w);

(9) P(w, negative_query): number of returned results in Google search by keyword (negative_query and w). We used the Google Search API to get the number of returned results in search online Google by keyword (negative_query and w);

(10) P(w): number of returned results in Google search by keyword w. We used the Google Search API to get the number of returned results in search online Google by keyword w;

(11) P($\neg w$,positive_query): number of returned results in Google search by keyword ($\neg w$ and positive_query). We used the Google Search API to get the number of returned results in search online Google by keyword ($\neg w$ and positive_query);

(12) P(w, $\neg positive\_query$): number of returned results in the Google search by keyword (w and ($\neg positive\_query$)). We used the Google Search API to get the number of returned results in search online Google by keyword (w and ($\neg positive\_query$));

(13) P($\neg w$, $\neg positive\_query$): number of returned results in the Google search by keyword ($\neg w$ and ($\neg positive\_query$)). We used the Google Search API to get the number of returned results in search online Google by keyword (($\neg w$) and ($\neg positive\_query$));

(14) P($\neg w$,negative_query): number of returned results in Google search by keyword ($\neg w$ and negative_query). We used the Google Search API to get the number of returned results in search online Google by keyword ($\neg w$ and negative_query);

(15) P(w, $\neg negative\_query$): number of returned results in the Google search by keyword (w and ($\neg negative\_query$)). We used the Google Search API to get the number of returned results in search online Google by keyword (w and ($\neg negative\_query$));

(16) P($\neg w$,$\neg negative\_query$): number of returned results in the Google search by keyword ($\neg w$ and ($\neg negative\_query$)). We used the Google Search API to get the number of returned results in search online Google by keyword ($\neg w$ and ($\neg negative\_query$)).

We have the information about the DNC as follows: (1) DNC(w, positive_query) $\geq$ 0 and DNC(w, positive_query) $\leq$ 1. (2) DNC(w, negative_query) $\geq$ 0 and DNC (w, negative_query) $\leq$ 1. (3) If DNC (w, positive_query) = 0 and DNC (w, negative_query) = 0 then SO_DNC (w) = 0. (4) If DNC (w, positive_query) =1 and DNC (w, negative_query) = 0 then SO_DNC (w) = 0. (5) If DNC (w, positive_query) = 0 and DNC (w, negative_query) = 1 then SO_DNC (w) = -1. (6) If DNC (w, positive_query) =1 and DNC (w, negative_query) = 1 then SO_DNC(w) = 0.

So, SO_DNC (w) $\geq$ -1 and SO_DNC (w) $\leq$ 1.

The polarity of the English word w is positive polarity If SO_DNC (w) $<$ 0. The polarity of the English word w is negative polarity if SO_DNC (w) $<$ 0. The polarity of the English word w is neutral polarity if SO_DNC (w) = 0. In addition, the semantic value of the English word w is SO_DNC (w). The result of calculating the valence w (English word) is similar to the result of calculating valence w by using AltaVista. However, AltaVista is no longer.

In summary, by using eq. (6), eq. (7), and eq. (8), we identified the valence and the polarity of one word (or one phrase) in English by using the DNC through the Google search engine with AND operator and OR operator.

Creating a basis English sentiment dictionary (bESD) in a sequential environment

In this part, we calculated the valence and the polarity of the English words or phrases for our bESD by using the DNC in a sequential system in the algorithm 1. According to [19–21], we had at least 55,000 English terms, including nouns, verbs, adjectives, etc.

Algorithm 1 Performing a bESD in a sequential environment

1: Input: the 55,000 English terms; the Google search engine
2: Output: a bESD.
3: **for all** Each term in the 55,000 terms **do**
4: By using eq. (5), eq. (6), eq. (7) and eq. (8) of the calculating a valence of one word (or one phrase) in English in the sub-section [Overview of identifying the valence and the polarity of one term in English using a DNC], the sentiment score and the polarity of this term were identified. The valence and the polarity were calculated by using the DNC through the Google search engine with AND operator and OR operator.
5: Add this term into the basis English sentiment dictionary (bESD)
6: **end for**
7: Return bESD

Our basis English sentiment dictionary (bESD) had more 55,000 English words (or English phrases) and bESD was stored in Microsoft SQL Server 2008 R2.

Creating a basis English sentiment dictionary (bESD) in a distributed system

In this part, we calculated the valence and the polarity of the English words or phrases for our bESD by using the DNC in a parallel network environment in the algorithm 2 and the algorithm 3. According to [19–21], we had at least 55,000 English terms, including nouns, verbs, adjectives, etc. This section included two phases: the Hadoop Map (M) phase and the Hadoop Reduce (R) phase. The input of the Hadoop Map phase was the 55,000 terms in English in [19–21]. The output of the Hadoop Map phase was one term which the sentiment score and the polarity are identified. The output of the Hadoop Map phase was the input of the Hadoop Reduce phase. Thus, the input of the Hadoop Reduce phase was one term which the sentiment score and the polarity are identified. The output of the Hadoop Reduce phase was the basis English sentiment dictionary (bESD).

Algorithm 2 Performing the Hadoop Map phase

1: Input: the 55,000 English terms; the Google search engine.
2: Output: one term which the sentiment score and the polarity are identified.
3: **for all** Each term in the 55,000 terms **do**
4: By using eq. (5), eq. (6), eq. (7) and eq. (8) of the calculating a valence of one word (or one phrase) in English in the sub-section [Overview of identifying the valence and the polarity of one term in English using a DNC], the sentiment score and the polarity of this term were identified. The valence and the polarity were calculated by using the DNC through the Google search engine with AND operator and OR operator.
5: Return this term
6: **end for**
7: Return this term

Algorithm 3 Implementing the Hadoop Reduce phase

1: Input: one term which the sentiment score and the polarity are identified − The output of M.
2: Output: a bESD.
3: Add this term into the basis English sentiment dictionary (bESD);
4: Return bESD

Our bESD had more 55,000 English words (or English phrases) and bESD was stored in Microsoft SQL Server 2008 R2.

3.2 Improving the LSA according to the sentiment lexicons of the bESD a sequential environment and a distributed network system

Reforming the LSA based on the sentiment lexicons of the bESD

According to the Latent semantic analysis (LSA) [4,6,7], it is a technique in natural language processing, that provides a theory and method for extracting and representing the contextual-usage and meaning of words by statistical computations applied to a large corpus of text. It closely approximates many aspects of human language learning and understanding. LSA produces a set of concepts (themes) exposed by the document analysis, which are based on the terms contained in the documents. It assumes that words that are similar in meaning occur in similar pieces of texts. We have the basic step involved in LSA based on the LSA [4, 6, 7] as follows: Computing Term-Passage Matrix -This is the document-term matrix where each row represents

Table 1: One Latent semantic analysis vector - LSAV

Documents	Latent	Semantic	Analysis	is	very	useful	slow
d1	1	1	1	1	2	1	0
D2	1	1	1	1	0	0	0

a document and the columns represent the terms (occurrence) in the document. Typically, given two documents d1and d2 with d1= "Latent Semantic Analysis is very very useful" and d2= "Latent Semantic Analysis is slow". Then, the document-term matrix is as shown below (called one Latent semantic analysis vector - LSAV) in Table 1. ⇒ The Latent semantic analysis vector:

$$LSAV = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

As known, in one sentence, there are sometimes many terms (meaningful words or meaningful phrases) bearing the neutral sentiment, the positive sentiment, or the negative sentiment. For example, we assume that in the bESD, "useful" is the positive sentiment and its valence is +2.3. "very" is the positive sentiment and its sentiment score is +0.3. "slow" is the negative sentiment and its valence is -1.1.

We see that the neutral terms are not the important role in a sentence. Thus, if we still use them in calculating the sentiment of the sentence, there are many noises for this calculating. We also see that the negative terms make many noises for calculating the sentiment of the positive polarity and the positive terms make many noises for calculating the sentiment of the negative polarity. We use the valences of the terms combined with their frequencies to remove many noises of identifying the sentiment classification of one sentence. We apply the valences of the sentiment lexicons of the bESD into the Vd1 and Vd2 as follows: According to the bESD, it is assumed that "Latent" is 0 of its valence; "Semantic" is 0 of its sentiment value; "Analysis" is 0 of its sentiment score; "useful" of +2.3 of its valence; "is" is 0 of its sentiment score; "document" is 0 of its valence; "classification" is 0 of its sentiment value; "an" is 0 of its valence; "very" is +0.3 of its sentiment score; the sentiment value of "slow" is −1.1. Therefore, we have d1 and d2 in Table 2.

⇒ d1 = (0, 0, 0, 0, 0.6, 2.3, 0) and d2 = (0, 0, 0, 0, 0, 0, 0)
⇒ emphasizing on the positive terms and the negative terms in one sentence. ⇒ The Latent semantic analysis vector:

$$LSAV = \begin{pmatrix} 1 & 1 & 1 & 1 & 2 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$$

In one LSA vector (LSAV), the value of each element is (its valences) × (its frequency).

Table 2: The value of d1 and the value of d2

Documents	Latent	Semantic	Analysis	is	very	useful	slow
d1	1 × (0)	1 × (0)	1 × (0)	1 × (0)	2 × (+0.3)	1 × (+2.3)	0 × (-1.1)
D2	1 × (0)	1 × (0)	1 × (0)	1 × (0)	0 × (+0.3)	0 × (+2.3)	0 × (-1.1)

Transferring all the documents of the testing data set and the training data set into the LSAVs in a sequential environmnent

In this section, we proposed the algorithm 4, the algorithm 5, the algorithm 6, and the algorithm 7 in the sequential system as follows: All the positive documents of the training data

set were transferred into one LSAV, called the positive LSAV group. All the negative documents of the training data set were transferred into one LSAV, called the negative LSAV group. Each document in the documents of the testing data set was transferred into one LSAV.

We implemented the algorithm 4 to create an order list of the LSAV which comprises all the meaningful terms of both the testing data set and the training data set in the sequential system. We proposed the algorithm 5 to transfer one document of the testing data set into one LSAV in the sequential system.
We built the algorithm 6 to transfer the positive documents of the training data set into one positive LSAV in the sequential system, called the positive LSAV group.

We proposed the algorithm 7 to transfer the negative documents of the training data set into one negative LSAV in the sequential system, called the negative LSAV group.

Algorithm 4 Creating an order list of the LSAV

```
1: Input: the documents of the training data set and the testing data set
2: Output: an order list of the LSAV − AnOrderListOfTheLSAV
3: Set AnOrderListOfTheLSAV ← ∅
4: for all Each document into the documents of the training data set and the testing data set
   do
5:     Split this document into the sentences
6:     for all Each sentence in the sentences do
7:         Split this sentence into the meaningful terms based on the sentiment lexicons of the
   bESD;
8:         for all Each term in the meaningful terms do
9:             if checking this term in AnOrderListOfTheLSAV is false then
10:                Add this term into AnOrderListOfTheLSAV
11:            end if
12:        end for
13:    end for
14: end for
15: Return AnOrderListOfTheLSAV
```

Transferring all the documents of the testing data set and the training data set into the the LSAVs in a distributed network system

In this section, we implemented many algorithms in the distributed network system as follows: All the positive documents of the training data set were transferred into one LSAV, called the positive LSAV group. All the negative documents of the training data set were transferred into one LSAV, called the negative LSAV group. Each document in the documents of the testing data set was transferred into one LSAV.

This section comprises the algorithm 8, the algorithm 9, the algorithm 10, the algorithm 11, the algorithm 12, the algorithm 13, the algorithm 14, and the algorithm 15.

We created an order list of the LSAV which comprises all the meaningful terms of both the testing data set and the training data set in the distributed network system in the algorithm 8 and the algorithm 9. This stage included two phases: the Hadoop Map phase (M) and the Hadoop Reduce phase (R). The input of M was the documents of the testing data set and the training data set. The output of R is one term. The input of R was the output of M, thus, the input of R was one term. The output of R was an order list of the LSAV − AnOrderListOfTheLSAV.

We transferred one document of the testing data set into one LSAV in the parallel system in the algorithm 10 and the algorithm 11. This stage included two phases: the Hadoop Map phase

Algorithm 5 Transferring one document of the testing data set into one LSAV in the sequential system

```
Input: one document in English and an order list of the LSA − AnOrderListOfTheLSAV;
Output: one LSAV;
Set Columns ← the length of AnOrderListOfTheLSAV
Set Rows ← the positive documents of the training data set
Set LSAV ← with its column is Columns and its rows is Rows
Set i ← 0
for all Each term in AnOrderListOfTheLSAV do
    Number := Count this term in this document
    Valence := Get the valence of this term based on the sentiment lexicons of the bESD
    Set LSAV[0][i] ← Number × Valence
    Set i ← i + 1
end for
for j:=1; j < Rows; j++ do
    for i:=0; i < Columns; i++ do
        Set LSAV[j][i] ← 0
    end for
end for
Return LSAV
```

(M) and the Hadoop Reduce phase (R). The input of M was one document in English and an order list of the LSA − AnOrderListOfTheLSAV. The output of M was one row of LSAV. The input of R was the output of M, thus, the input of R was one row of LSAV. The output of R was one LSAV of this document.

We transferred the positive documents of the training data set into one positive LSAV in the distributed system, called the positive LSAV group in the algorithm 12 and the algorithm 13. This stage included two phases: the Hadoop Map phase (M) and the Hadoop Reduce phase (R). The input of M was the positive documents in English and an order list of the LSA − AnOrderListOfTheLSAV. The output of M was one row of PositiveLSAV. The input of R was the output of M, thus, the input of R was one row of PositiveLSAV. The output of R was PositiveLSAV.

We transferred the negative documents of the training data set into one negative LSAV in the parallel system, called the negative LSAV group in the algorithm 14 and the algorithm 15. This stage included two phases: the Hadoop Map phase (M) and the Hadoop Reduce phase (R). The input of the Hadoop Map phase was the negative documents in English and an order list of the LSA − AnOrderListOfTheLSAV. The output of the Hadoop Map phase was one row of NegativeLSAV. The input of the Hadoop Reduce phase was the output of the Hadoop Map, thus, the input of the Hadoop Reduce phase was one row of NegativeLSAV. The output of the Hadoop Reduce phase was NegativeLSAV.

Algorithm 6 Transferring the positive documents of the training data set into one positive LSAV in the sequential system

```
Input: the positive documents of the training data set and an order list of the LSA − AnOrderListOfTheLSAV
Output: one positive LSAV - the positive LSAV group
Set Columns ← the length of AnOrderListOfTheLSAV
Set Rows ← the positive documents of the training data set
Set PositiveLSAV ← with its column is Columns and its rows is Rows
for j := 0; j < Rows; j++ do
    Set i ← 0
    for all Each term in AnOrderListOfTheLSAV do
        Number := Count this term in the document (j) of the positive documents of the training data set
        Valence := Get the valence of this term based on the sentiment lexicons of the bESD
        Set PositiveLSAV[j][i] ← Number × Valence
        Set i ← i + 1
    end for
end for
Return PositiveLSAV
```

3.3 Using the LSA and a DNC to cluster the documents of the testing data set into either the positive or the negative in both a sequential environment and a parallel distributed system

Using the LSA and a DNC to cluster the documents of the testing data set into either the positive or the negative in a sequential environment

This sub-section has the algorithm 16, and the algorithm 17. In this section, we used the LSA and a DNC to cluster the documents of the testing data set into either the positive or the negative in a sequential environment.

We built the algorithm 16 to cluster one LSAV (corresponding one document of the testing data set) into either positive polarity or the negative polarity in the sequential environment.

We proposed the algorithm 17 to cluster all the documents of the testing data set into either the positive or the negative in the sequential system by using the LSA and the DNC.

Using the LSA and a DNC to cluster the documents of the testing data set into either the positive or the negative in a distributed system

This part includes the algorithm 18, the algorithm 19, the algorithm 20, and the algorithm 21.

In this part, we used the LSA and a DNC to cluster the documents of the testing data set into either the positive or the negative in a distributed system.

We clustered one LSAV (corresponding one document of the testing data set) into either positive polarity or the negative polarity in the distributed environment in the algorithm 18 and the algorithm 19. This stage comprised two phases: the Hadoop Map phase (M) and the Hadoop Reduce phase (R). The input of M was one LSAV (corresponding one document of the testing data set); the positive LSAV group and the negative LSAV group. The output of M was the result of the sentiment classification of one document. The input of R was the output of M, thus, the input of R was the result of the sentiment classification of one document. The output of R

Algorithm 7 Transferring the negative documents of the training data set into one negative LSAV in the sequential system

```
Input: the negative documents of the training data set and an order list of the LSA − AnOrderListOfTheLSAV
Output: one negative LSAV − the positive LSAV group
Set Columns ← the length of AnOrderListOfTheLSAV
Set Rows ← the negative documents of the training data set
Set NegativeLSAV ← with its column is Columns and its rows is Rows
for j := 0; j < Rows; j++ do
    Set i ← 0
    for all Each term in AnOrderListOfTheLSAV do
        Number := Count this term in the document (j) of the positive documents of the training data set
        Valence := Get the valence of this term based on the sentiment lexicons of the bESD
        Set NegativeLSAV[j][i] ← Number × Valence
        Set i ← i + 1
    end for
end for
Return NegativeLSAV
```

was the result of the sentiment classification of this document of the testing data set.

We used the LSA and the DNC to cluster the documents of the testing data set into either the positive or the negative in the distributed system in the algorithm 20 and the algorithm 21. This stage comprised two phases: the Hadoop Map phase (M) and the Hadoop Reduce phase (R). The input of M was the documents of the testing data set. The output of M was the result of the sentiment classification of one document. The input of R was the output of M, thus, the input of R is the result of the sentiment classification of one document. The output of R was the results of the sentiment classification of the documents of the testing data set.

4 Experiment

To implement the proposed model, we have already used Java programming language to save the 11,000,000 documents of the testing data set and the 5,000,000 documents of the training data set, and to save the results of emotion classification. The proposed model was implemented in both the sequential system and the distributed network environment.

The configuration of one server is: IntelÂŽ Server Board S1200V3RPS, IntelÂŽ PentiumÂŽ Processor G3220 (3M Cache, 3.00 GHz), 2GB CC3-10600 ECC 1333 MHz LP Unbuffered DIMMs; and the operating system of the server is: Cloudera.

Our novel model related to the Latent Semantic Analysis and a Dennis Coefficient is implemented in the sequential environment with the configuration as follows: The sequential environment in this research includes 1 node (1 server).

The proposed model related to the Latent Semantic Analysis and a Dennis Coefficient is performed in the Cloudera parallel network environment with the configuration as follows: This Cloudera system includes 9 nodes (9 servers). All 9 nodes have the same configuration information

In Table 3, we show the accuracy and the results of our novel model in the testing data set.

The average time of the classification of our new model for the English documents in testing data set are displayed in Table 4.

Algorithm 8 Performing the Hadoop Map phase

```
Input: the documents of the testing data set and the document of the training data set
Output: one term;//the output of M.
Input the documents of the testing data set and the document of the training data set into M in the Cloudra system
for all Each document into the documents of the training data set do
    Split this document into the sentences
    for all Each sentence in the sentences do
        Split this sentence into the meaningful terms based on the sentiment lexicons of the bESD
        for all Each term in the meaningful terms do
            Return this term;//the output of the Hadoop Map.
            for all Each document in the documents of the testing data set do
                Split this document into the sentences
                for all Each sentence into the sentences do
                    Split this sentence into the meaningful terms according to the sentiment lexicons of the bESD
                    for all Each term in the meaningful terms do
                        Return this term;//the output of M
                    end for
                end for
            end for
        end for
    end for
end for
```

Table 3: The accuracy and The results of our novel model in the testing data set.

	Testing Dataset	Correct Classification	Incorrect Classification	Accuracy
Negative	5,500,000	4,884,751	615,249	
Positive	5,500,000	4,878,849	621,151	88.76%
Summary	11,000,000	9,763,600	1,236,400	

Algorithm 9 Implementing the Hadoop Reduce phase

```
Input: one term from M
Output: an order list of the LSAV − AnOrderListOfTheLSAV
Receive on term from M;
if checking this term in AnOrderListOfTheLSAV is false then
    Add this term into AnOrderListOfTheLSAV
end if
Return AnOrderListOfTheLSAV
```

Algorithm 10 Performing the Hadoop Map phase

Input: one document in English and an order list of the LSA − AnOrderListOfTheLSAV
Output: one row of LSAV; //the output of M.
Input one document in English and an order list of the LSA − AnOrderListOfTheLSAV into the Hadoop Map in the Cloudera
Set Columns ← the length of AnOrderListOfTheLSAV
Set Rows ← the positive documents of the training data set
Set OneRowOfLSAV ← with its columns is Columns
Set i ← 0
for all Each term in AnOrderListOfTheLSAV **do**
    Number := Count this term in this document
    Valence := Get the valence of this term based on the sentiment lexicons of the bESD
    Set OneRowOfLSAV[i] ← Number × Valence
    Set i ← i + 1
end for
Return OneRowOfLSAV

Algorithm 11 Implementing the Hadoop Reduce phase

Input: one row of LSAV; //the output of M.
Output: one LSAV
Receive one row of LSAV
Add this row into LSAV
for j:= 1; j < Rows; j++ **do**
    for i := 0; i < Columns; i++ **do**
        Set LSAV[j][i] ← 0
    end for
end for
Return LSAV

Algorithm 12 Performing the Hadoop Map phase

Input: the positive documents in English and an order list of the LSA − AnOrderListOfTheLSAV
Output: one row of LSAV; //the output of M.
Set Columns ← the length of AnOrderListOfTheLSAV
Set Rows ← the positive documents of the training data set
for j := 0; j < Rows; j++ **do**
    Set OneRowOfPositiveLSAV ← with its column is Columns
    Set i ← 0
    for all Each term in AnOrderListOfTheLSAV **do**
        Number := Count this term in the document (j) of the positive documents of the training data set
        Valence := Get the valence of this term based on the sentiment lexicons of the bESD
        Set OneRowOfPositiveLSAV[i] ← Number × Valence
    end for
    Set i ← i + 1
end for
Return OneRowOfPositiveLSAV; //the output of M

Algorithm 13 Implementing the Hadoop Reduce phase

Input: one row of LSAV; //the output of M.
Output: PositiveLSAV
Receive one row of PositiveLSAV
Add this row into PositiveLSAV
Return PositiveLSAV

Algorithm 14 Performing the Hadoop Map phase

Input: the negative documents in English and an order list of the LSA − AnOrderListOfTheLSAV
Output: one row of LSAV; //the output of M.
Set Columns ← the length of AnOrderListOfTheLSAV
Set Rows ← the negative documents of the training data set
for j := 0; j < Rows; j++ **do**
    Set OneRowOfNegativeLSAV ← with its column is Columns
    Set i ← 0
    for all Each term in AnOrderListOfTheLSAV **do**
        Number := Count this term in the document (j) of the negative documents of the training data set
        Valence := Get the valence of this term based on the sentiment lexicons of the bESD
        Set OneRowOfNegativeLSAV[i] ← Number × Valence
        Set i ← i + 1
    end for
end for
Return OneRowOfPositiveLSAV; //the output of M

Algorithm 15 Implementing the Hadoop Reduce phase

Input: one row of LSAV; //the output of M.
Output: NegativeLSAV
Receive one row of NegativeLSAV
Add this row into NegativeLSAV
Return NegativeLSAV

Table 4: Average time of the classification of our new model for the English documents in testing data set

	Average time of the classification 11,000,000 English documents.
The Latent Semantic Analysis and a Dennis Coefficient in the sequential environment	43,460,194 seconds
The Latent Semantic Analysis and a Dennis Coefficient in the Cloudera distributed system with 3 nodes	13,120,064 seconds
The Latent Semantic Analysis and a Dennis Coefficient in the Cloudera distributed system with 6 nodes	7,360,032 seconds
Latent Semantic Analysis and a Dennis Coefficient in the Cloudera distributed system with 9 nodes	4,851,132 seconds

Algorithm 16 Clustering one LSAV (corresponding one document of the testing data set) into either positive polarity or the negative polarity in the sequential environment

Input: one LSAV (corresponding one document of the testing data set); the positive LSAV group and the negative LSAV group.
Output: positive, negative, neutral;
Measure_1 := Similarity measure between this LSAV and the positive LSAV group by using the eq. (3) of the calculating a valence of one word (or one phrase) in English in the sub-section [Overview of identifying the valence and the polarity of one term in English using a DNC].
Measure_2 := Similarity measure between this LSAV and the negative LSAV group by using the eq. (3) of the calculating a valence of one word (or one phrase) in English in the sub-section [Overview of identifying the valence and the polarity of one term in English using a DNC].
if Measure_1 is greater than Measure_2 **then**
    Return positive
elseMeasure_1 is less than Measure_2
    Return negative
end if
Return neutral

Algorithm 17 Clustering all the documents of the testing data set into either the positive or the negative in the sequential system by using the LSA and the DNC

Input: the documents of the testing data set and the training data set.
Output: positive, negative, neutral;
the creating a bESD in a sequential environment in the sub-section [Creating a basis English sentiment dictionary (bESD) in a sequential environment].
the algorithm 4 to create an order list of the LSAV which comprises all the meaningful terms of both the testing data set and the training data set in the sequential system.
the algorithm 6 to transfer the positive documents of the training data set into one positive LSAV in the sequential system, called the positive LSAV group.
the algorithm 7 to transfer the negative documents of the training data set into one negative LSAV in the sequential system, called the negative LSAV group.
Results := null;
for all Each document in the documents of the testing data set **do**
    One LSAV := the algorithm 5 to transfer one document of the testing data set into one LSAV in the sequential system.
    OneResult := the algorithm 16 to cluster one LSAV (corresponding one document of the testing data set) into either positive polarity or the negative polarity in the sequential environment.
    Add OneResult into Results;
end for
Return Results;

Algorithm 18 Implementing the Hadoop Map phase

```
Input: one LSAV (corresponding one document of the testing data set); the positive LSAV group and the negative LSAV group.
Output: the result of the clustering − OneResult; //the output of M.
Measure_1 := Similarity measure between this LSAV and the positive LSAV group by using the eq. (3) of the calculating a valence of one word (or one phrase) in English in the subsection [Overview of identifying the valence and the polarity of one term in English using a DNC].
Measure_2 := Similarity measure between this LSAV and the negative LSAV group by using the eq. (3) of the calculating a valence of one word (or one phrase) in English in the subsection [Overview of identifying the valence and the polarity of one term in English using a DNC]
if Measure_1 is greater than Measure_2 then
    OneResult := positive;
else
    if Measure_1 is less than Measure_2 then
        OneResult := negative;
    else
        OneResult := neutral;
    end if
end if
Return OneResult; //the output of M
```

Algorithm 19 Performing the Hadoop Reduce phase

```
Input: OneResult; //the output of M.
Output: positive, negative, neutral;
Receive OneResult;
Return OneResult;
```

Algorithm 20 Implementing the Hadoop Map phase

Input: the documents of the testing data set and the training data set.
Output: positive, negative, neutral;
the creating a bESD in a distributed system in the sub-section [Creating a basis English sentiment dictionary (bESD) in a distributed system].
the algorithm 4 to create an order list of the LSAV which comprises all the meaningful terms of both the testing data set and the training data set in the sequential system.
creating an order list of the LSAV which comprises all the meaningful terms of both the testing data set and the training data set in the distributed network system in the algorithm 8 and the algorithm 9.
transferring the positive documents of the training data set into one positive LSAV in the sequential system, called the positive LSAV group in the algorithm 12 and the algorithm 13.
transferring the negative documents of the training data set into one negative LSAV in the sequential system, called the negative LSAV group in the algorithm 14 and the algorithm 15.
Input the documents of the testing data set, the positive LSAV group and the negative LSAV group into the Hadoop Map in the Cloudera system;
for all Each document in the documents of the testing data set **do**
    One LSAV := transferring one document of the testing data set into one LSAV in the parallel system in the algorithm 10 and the algorithm 11.
    OneResult :=clustering one LSAV (corresponding one document of the testing data set) into either positive polarity or the negative polarity in the distributed environment in the algorithm 18 and the algorithm 19.
    Return OneResult;//the output of M
end for
Return OneResult;//the output of M

Algorithm 21 Performing the Hadoop Reduce phase

Input: OneResult − the result of the sentiment classification of one document (the input of R is the output of M).
Output: the results of the sentiment classification of the documents of the testing data set;
Receive OneResult − the result of the sentiment classification of one document.
Add this OneResult into the results of the sentiment classification of the documents of the testing data set;
Return the results of the sentiment classification of the documents of the testing data set;

5 Conclusion

In this survey, a new model has been proposed to classify sentiment of many documents in English using the Latent Semantic Analysis and a Dennis Coefficient with Hadoop Map (M) /Reduce (R) in the Cloudera parallel network environment. Based on our proposed new model, we have achieved 88.76% accuracy of the testing data set in Table 3, and this is better than the accuracies of many previous models of the semantic analysis. Besides, we have also compared the novel model with the previous models, and the experiments and the results of our proposed model are better than that of the previous model. Until now, not many studies have shown that the clustering methods can be used to classify data. According to Table 4, the average time of the sentiment classification of using the Latent Semantic Analysis and a Dennis Coefficient in the sequential environment is 43,460,194 seconds / 11,000,000 English documents and it is greater than the average time of the sentiment classification of using the Latent Semantic Analysis and a Dennis Coefficient in the Cloudera parallel network environment with 3 nodes which is 13,120,064 seconds / 11,000,000 English documents. The average time of the sentiment classification of using the Latent Semantic Analysis and a Dennis Coefficient in the Cloudera parallel network environment with 9 nodes is 4,851,132 seconds / 11,000,000 English documents, and It is the shortest time in the table. Besides, the average time of the sentiment classification of using the Latent Semantic Analysis and a Dennis Coefficient in the Cloudera parallel network environment with 6 nodes is 7,360,032 seconds / 11,000,000 English documents
The accuracy of the proposed model is dependent on many factors as follows: (1) The LSA − related algorithms. (2) The testing data set. (3) The documents of the testing data set must be standardized carefully. (4) Transferring one document into one LSAV.

Table 5: Comparisons of our model′s positives and negatives the surveys related to the Latent Semantic Analysis (LSA)

Studies	Approach	Positives	Negatives
[4]	Unsupervised Learning by Probabilistic Latent Semantic Analysis	The survey presents perplexity results for different types of text and linguistic data collections and discusses an application in automated document indexing. The experiments indicate substantial and consistent improvements of the probabilistic method over standard Latent Semantic Analysis.	No mention
[6]	The latent semantic analysis theory of acquisition, induction, and representation of knowledge.	A new general theory of acquired similarity and knowledge representation, latent semantic analysis (LSA), is presented and used to successfully simulate such learning and several other psycholinguistic phenomena.	No mention
Our work	LSA using A DNC	The positives and negatives of the proposed model are given in the Conclusion section.	

The execution time of the proposed model is dependent on many factors as follows: (1) The parallel network environment such as the Cloudera system. (2) The distributed functions such as Hadoop Map (M) and Hadoop Reduce (R). (3) The LSA − related algorithms. (4) The performance of the distributed network system. (5) The number of nodes of the parallel network environment. (6) The performance of each node (each server) of the distributed environment. (7) The sizes of the training data set and the testing data set. (8) Transferring one document into one LSAV.

The proposed model has many advantages and disadvantages. Its positives are as follows: It uses the Latent Semantic Analysis and a Dennis Coefficient to classify semantics of English documents based on sentences. The proposed model can process millions of documents in the shortest time. This study can be performed in distributed systems to shorten the execution time of the proposed model. It can be applied to other languages. Its negatives are as follows: It has a low rate of accuracy. It costs too much and takes too much time to implement this proposed model.
To understand the scientific values of this research, we have compared our model's results with many studies in the tables below. Our novel model has more benefits than the studies in the tables, and the results of this model are better than that of the works in the tables.
In Table 5, we present the comparisons of our model′s positives and negatives the surveys related to the Latent Semantic Analysis (LSA).
The comparisons of our model′s benefits and drawbacks with the studies related to the DENNIS coefficient (DNC) are shown in Table 6.

Table 6: Comparisons of our model′s benefits and drawbacks with the studies related to the DENNIS coefficient (DNC)

Studies	Approach	Benefits	Drawbacks
[5]	Analysis of Macromolecular Polydispersity in Intensity Correlation Spectroscopy: The Method of Cumulants	A method is described by which the distribution function of the decay rates (and thus the extent of polydispersity) can be characterized, in a light scattering experiment, by calculation of the moments or cumulants.	No mention
[3]	A Survey of Binary Similarity and Distance Measures	The authors collected 76 binary similarity and distance measures used over the last century and reveal their correlations through the hierarchical clustering technique	No mention
Our work	LSA using A DNC	The advantages and disadvantages of this survey are shown in the Conclusion section.	

Conflict of interests

The authors declare that there is no conflict of interests.

Bibliography

[1] Bai, A.; Hammer, H.; Yazidi, A.; Engelstad, P. (2014); Constructing sentiment lexicons in Norwegian from a large text corpus, *2014 IEEE 17th International Conference on Computational Science and Engineering*, 231-237, 2014.

[2] Baldocchi, D.D.; Hincks, B.B.; Meyers, T.P.(1988); Measuring Biosphere-Atmosphere Exchanges of Biologically Related Gases with Micrometeorological Methods, *Ecology society of America*, 59(5), 1331-1340, 1988.

[3] Choi, S.-S; Cha, S.-H.; Tappert, C.C. (2010); A Survey Of Binary Similarity And Distance Measures, *Systemics, Cybernetics And Informatics*, 8(1), 43-48, 2010.

[4] Hofmann, T. (2001); Unsupervised Learning by Probabilistic Latent Semantic Analysis, *Machine Learning*, 42(1-2), 177-196, 2001.

[5] Koppel, D.E. (1972); Analysis of Macromolecular Polydispersity in Intensity Correlation Spectroscopy: The Method of Cumulants, *The Journal of Chemical Physics*, 57(11), 4814, 1972.

[6] Landauer, T.K.; Dumais, S. T. (1997); A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge, *Psychological Review*, 104(2), 211-240, 1997.

[7] Landauer, T.K.; Foltz, P. W.; Laham, D. (2009); An introduction to latent semantic analysis, *Discourse Processes*, 25(2-3), 259-284, 2009.

[8] Ngoc, P.V.; Ngoc, C.V.T.; Ngoc, T.V.T. et al. (2017); A C4.5 algorithm for english emotional classification, *Evolving Systems*, 1-27, 2017.

[9] Phu, V.N. ; Tuoi, P.T. (2014); Sentiment classification using Enhanced Contextual Valence Shifters, *International Conference on Asian Language Processing (IALP)*, 224-229, 2014.

[10] Phu, V.N.; Dat, N.D.; Tran, D.T.N.; Chau, V.T.N.; Nguyen, T.A.(2017); Fuzzy C-Means for English Sentiment Classification in a Distributed System, *International Journal of Applied Intelligence*, 45(3), 717-738 2017.

[11] Phu, V.N.; Chau, V.T.N.; Tran, D.T.N. (2017); SVM for English Semantic Classification in Parallel Environment, *International Journal of Speech Technology*, 20(3), 487-508, 2017.

[12] Phu, V.N.; Tran, V.T.N.; Chau, V.T.N. et al. (2017); A Decision Tree using ID3 Algorithm for English Semantic Analysis, *International Journal of Speech Technology*, 20(3), 593-613, 2017.

[13] Phu, V.N.; Chau, V.T.N.; Tran, V.T.N. et al. (2017); A Vietnamese adjective emotion dictionary based on exploitation of Vietnamese language characteristics, *International Journal of Artificial Intelligence Review (AIR)*, 1-67, 2017

[14] Phu, V.N., Chau, V.T.N., Dat, N.D. et al. (2017); A Valences-Totaling Model for English Sentiment Classification, *International Journal of Knowledge and Information Systems*, 53(3), 579-636, 2017.

[15] Phu, V.N.; Chau, V.T.N.; Tran, V.T.N(2017); Shifting Semantic Values of English Phrases for Classification, *International Journal of Speech Technology*, 20(3), 579-636, 2017.

[16] Phu, V.N., Chau, V.T.N., Tran, V.T.N. et al. (2017); A Valence-Totaling Model for Vietnamese Sentiment Classification, *International Journal of Evolving Systems*, 1-47, 2017.

[17] Phu, V.N., Tran, V.T.N., Chau, V.T.N. et al. (2017); Semantic Lexicons of English Nouns for Classification, *International Journal of Evolving Systems*, 1-69, 2017.

[18] Turney, D. P.; Littman, M.L. (2002); Unsupervised Learning of Semantic Orientation from a Hundred-Billion-Word Corpus, *arXiv:cs/0212012, Learning*, 2002.

[19] Cambridge English Dictionary (2017); *http://dictionary.cambridge.org/*

[20] Longman English Dictionary (2017); *http://www.ldoceonline.com/*

[21] Oxford English Dictionary (2017); *http://www.oxforddictionaries.com/*

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 429-439, June 2018.

Autopilot Design for Unmanned Surface Vehicle based on CNN and ACO

D. Zhao, T. Yang, W. Ou, H. Zhou

Dongming Zhao, Tiantian Yang, Wen Ou, Hao Zhou*
School of Automation
Wuhan University of Technology
Wuhan 430070, Hubei, China
dmzhao@whut.edu.cn, ttyang@whut.edu.cn, wenou@mail.hust.edu.cn
*Corresponding author; 18601157377@163.com

Abstract: There is a growing concern to design intelligent controllers for autopiloting unmanned surface vehicles as solution for many naval and civilian requirements. Traditional autopilot's performance declines due to the uncertainties in hydrodynamics as a result of harsh sailing conditions and sea states. This paper reports the design of a novel nonlinear model predictive controller (NMPC) based on convolutional neural network (CNN) and ant colony optimizer (ACO) which is superior to a linear proportional integral-derivative counterpart. This combination helps the control system to deal with model uncertainties with robustness. The results of simulation and experiment demonstrate the proposed method is more efficient and more capable to guide the vehicle through LOS waypoints particularly in the presence of large disturbances.

Keywords: USV, autopilot, predictive control, Convolution Neural Network (CNN), Ant Colony Optimization (ACO), rolling optimization.

1 Introduction

In recent years, The application of USVs is ever increasing in the fields of oceanography [13] [3], meteorology [7], military and commercial applications. Autopilot is a device for controlling the heading of USV in a truly autonomous mode, which plays a key role in course-keeping for USV [24]. When USV is working under harsh sea condition and high-risk environment, the performance of traditional autopilot declines in the presence of time-varying ocean disturbances, and measurement noises. A lot of autopilot designs for USV have been proposed to solve this problem. Gao *et al.* proposed a fuzzy neural network controller based on chaos neural network forecast model for the USV in complex sea condition [9, 15, 17]. An optimizing sliding mode cascade control structure is proposed to determine the optimal sliding surface parameters for sliding mode control of underactuated USV systems [18]. Further, nonlinear controller of USV is designed by backstepping method [1, 10]. The fuzzy control approach is also applied to the control of USV [8, 11, 19]. Li *et al.* proposed an adaptive radial basis function based on neural network controller for the nonlinear control of USV which contains modeling errors and unknown bounded environment disturbances [4]. However, it's still an open problem that how to develop effective method for robustness and high-precision steering of USV in extreme sea condition.

This paper reports a kind of autopilot based on convolutional neural network(CNN) and ant colony optimizer. A NMPC based on CNN is used to compensate this predicted disturbance and the ant colony algorithm is used to predict the disturbances. This approach helps the autopilot to deal with model uncertainties and so on. The simulation and experiment results verified the efficacy of the proposed method.

The rest paper is organized as follows: Section 2 states the specifications of the USV and the problem formulation. Section 3 elaborates the autopilot design based on CNN and ACO. Section 4 provides the simulation results and experiment results. Section 5 gives the conclusion of this paper.

Figure 1: WH-01 USV

2 Problem formulation

Consider a robotic marine surface vehicle, called WH- 01 (as shown in Figure 1), whose specifications are listed in Table 1.The vehicle, designed in the control laboratory at WHUT(Wuhan University of Technology), has a hydraulic jet propeller and one rudder driven by servo motors for controlling its surge speed and heading. The maximum angular detection for rudder is $\pm 35°$. It is equipped with navigation sensors including Beidou, 3-axis magnetometer, 3-axis accelerometer, and 3-axis rate gyro. The magnetometer and gyro are used to measure its yaw and yaw rate during trails, respectively. In addition, a 4G communication module is installed to control and monitor the vehicle from remote computer.

Table 1: SPECIFICATIONS OF WH-01 USV

Parameter	Value
Length $/m$	8.075
Vertical length $/m$	8
draft depth $/m$	0.6
moulded depth$/m$	1.15
Full loaded displacement$/T$	3.2
Maximum speed $/kn$	12
Cruising speed $/kn$	6
Propulsion mode	water jet propulsion

Because of the interference of wind, waves and currents, the steering model of USV is expressed as follows:

$$T_1T_2\ddot{\Psi} + (T_1 + T_2)\dot{\Psi} + \Psi + a\Psi^3 = K\delta + KT_3\dot{\delta} + f_a + f_\omega + f_l \tag{1}$$

$$T_1T_2 = (m + \lambda_{22})(I_z + \lambda_{66})/C \tag{2}$$

$$T = T_1 + T_2 - T_3 \tag{3}$$

where Ψ is the heading angle of the USV, δ is the rudder angle of the USV , T is the turning lag index, $T_1T_2T_3$ is the second-order turning lag index, K is the turning ability index, m is the mass of the USV, f_a is equivalent disturbance rudder angle of the wind, f_ω is equivalent disturbance rudder angle of the disturbances due to waves, f_l is equivalent disturbance rudder angle of the disturbances due to currents.

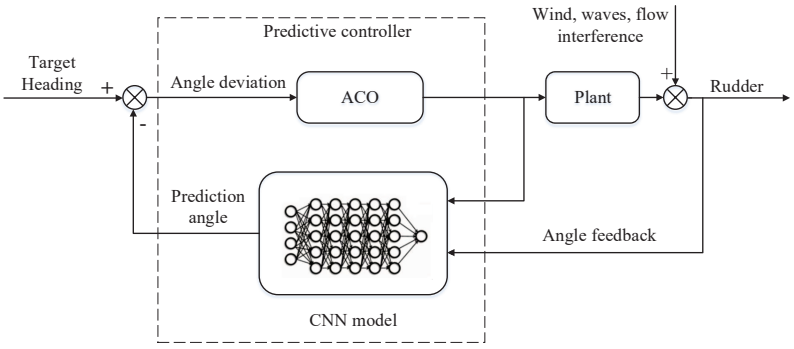


Figure 2: A block diagram of the autopilot based on CNN and ACO for USV

According to the calculation method in the literature [6] and the Zigzag experimental data of $\Psi/\delta = 15°/15°$ for the draft state in the Taihu Lake, it can be obtained that:

$$K = 2.9097, T = 55.8855 \tag{4}$$

Zigzag experimental result is shown in table 2.

Table 2: ZIGZAG EXPERIMENTAL DATA OF USV

Time /s	Rudder angle /$\circ$	Heading angle /$\circ$	Remark
$1'82$	15	15	-
$5'82$	15	28.2	Maximum of forward heading angle
$9'44$	-15	0	-
$12'68$	-15	-15	Maximum of backward heading angle
$15'62$	15	-24.7	-
$23'06$	15	15	Maximum of forward heading angle
$26'56$	-15	28.2	-

3 Robust adaptive autopilot

The autopilot design of USV is based on convolutional neural network(CNN) and ant colony optimizer. Closed-loop state prediction and rolling horizon optimization are included in the predictive controller. Particularly, The future dynamic trend is predicted by the NMPC based on historical data of input and output. An ant colony algorithm acting as nonlinear optimizer is used to improve the rolling horizon. The predictive value is corrected by negative feedback and then compared with reference input. The difference between the predicted value and the expected value is minimized in the next period of time. The proposed autopilot design is shown in Figure 2.

3.1 CNN model

The architecture of CNN based model is shown in Figure 3. The delay lines with taps are represented by TDL(Tapped Delay Line). The CNN is in parallel with the process of system.

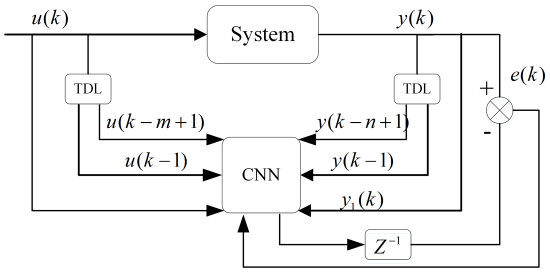


Figure 3: Architecture of CNN based model

The predictive error is used as training signal of the network. The single-step CNN's prediction model is defined as:

$$y_m(k+1) = \hat{f}[y(k), y(k-1), \cdots, y(k-n+1); u(k), u(k-1), \cdots, u(k-m+1)] \tag{5}$$

where $y(k)$ is the output of the heading angle at sampling time k, $u(k)$ is the input of the heading angle at sampling time k, m is the system input order, n is the system output order, $\hat{f}(\cdot)$ is the input and output mapping function of the CNN model, $y_m(k+1)$ is the predictive value of the heading angle.

Because the single-step neural network model has less predictive information, the multi-step prediction method is used to improve the anti-interference and robustness of the autopilot. The prediction model of single-step neural network is used to infer the multi-step one, which is expressed as:

$$y_m(k+p) = \hat{f}[y(k+p-1), \cdots, y(k+p-n); u(k+p-1), \cdots, u(k+p-m)] \tag{6}$$

At sampling time k, $y(k+p-1), \cdots, y(k+1)$ is the actual output data of future, which can't be measured. So, the predictive value is used to replace the output of future. In addition, the predicted data before the kth moment can be replaced by historical data. So, the p step predictive model is formulated as:

$$y_m(k+p) = \hat{f}[y_m(k+p-1), \cdots, y_m(k+p-n); u(k+p-1), \cdots, u(k+p-m)] \tag{7}$$

The CNN model structure of multi-step prediction is shown in Figure 4.

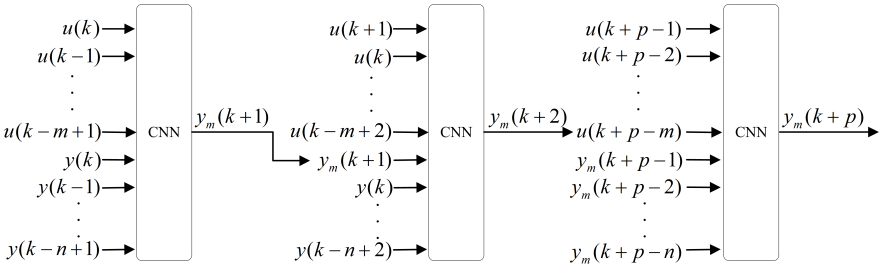


Figure 4: Structure of multi-step neural network prediction

3.2 Feedback correction

The reference trajectory function of the autopilot is based on the reference trajectory provided by the model control algorithm, which is expressed as:

$$y_\gamma(k+1) = ay(k) + (1-a)\omega \tag{8}$$

Because the predictive controller has the model mismatch problem, a large deviation is produced between prediction model and actual output of the object. The predictive output should be corrected to reduce the predictive error.

At sampling time k, we must calculate the error between the actual output value $y(k)$ and the output value predicted by the model($y_m(k)$) at first. Then, the controller adds the error to the model predictive output($y_m(k+1)$) to get the closed-loop predictive output. The future closed-loop predictive output of p step is as follows:

$$y_{pj}(k+j) = y_m(k+j) + r_j e_j(k) + \beta_j[e_j(k) - e_j(k-1)] \tag{9}$$

where r_j is the error correction coefficient, β_j is the change rate of correction coefficient, $e_j(k) = y(k) - y_m(k)$ is the difference between actual output and the expected output, $j = 1, 2, \cdots, p$ is the number of predictive steps, $y_{pj}(k+j)$ is the predictive value of the output component after the $(k+j)$ moment, $y_{pj}(k+j)$ is the model predictor of the output component at sampling time $(k+j)$.

3.3 Rolling optimization

Usually, the difference between the predictive value of heading and the expected value of heading is supposed to be zero. The range of the control increment should not be too large. For this reason, the optimal law of predictive controller is determined by the performance index. The secondary performance indicator is used as the objective function. The optimal control objective function is introduced by:

$$\begin{aligned} J =& \frac{1}{2}\{\sum_{j=1}^{s}\sum_{i=1}^{P} q_{ij}[y_{pj}(k+i) - y_{\gamma j}(k+j)]^2 \\ &+ \sum_{j=1}^{\gamma}\sum_{i=1}^{M} \lambda_{ij}[u_j(k+i-1) - u_j(k+i-2)]^2\} \end{aligned} \tag{10}$$

where P is the length of predictive time domain, M is the length of time domain, $M \leq P$, q_j is the error weighting factor, λ_j is the control incremental weighting factor, $y_j(k+j)$ is the reference trajectory of the output at $k+j$ moment.

The ant colony optimizer(ACO) is used to maximize the objective function of the controller within the range of rudder angle, which has good robustness and can be implemented in parallel. The implementation steps of the ACO are enumerated below:

1. Initialize the ant pheromone distribution, set the number of iterations;

2. 10 ants are initialized in the neighborhood, moving according to the following transfer probability. For each ant i, the objective function J_i is defined, the transfer probability for ant i at time k is as follows:

$$p_{ij}(k) = \frac{[\tau_j(k)]^{1.2}[\triangle J_{ij}(k)]^2}{\sum_r [\tau_r(k)]^{1.2}[\triangle J_{ir}(k)]^2} \tag{11}$$

   where $\tau_j(k)$ is the j-neighborhood attracting intensity of the ants at k time. $\triangle J_{ij} = J_i - J_j$ is the difference value of the target function.

3. Calculate the objective function value of each ant, and record the optimal control sequence in the current ant colony;

4. Revise the intensity of pheromone according to the pheromone update equation, following the next equation:

$$\begin{cases} \tau_j(k+1) = 0.7\tau_j(k) + \sum \triangle\tau_j \\ \triangle\tau_j = 1/J_j \end{cases} \tag{12}$$

5. Number of iterations $N = N + 1$

6. If the number of iterations does not reach the ending iteration number 100, the second step is returned, otherwise, the loop is terminated and the optimal control sequence is outputted.

4 Results and Discussion

4.1 Simulation results

The disturbance of wind and wave on USV is simulated using white noise and a second-order wave transfer function [26]. The disturbance of ocean current is treated as a constant value.

$$y(s) = K_c\omega(s)\frac{K_\omega s}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{13}$$

where $\omega(s)$ is a Gaussian white noise with a mean of zero, the value of power spectrum density is 0.1, and $K_c = 5$ is a constant disturbance coefficient, $\zeta = 0.3$ is a damp coefficient, $K_\omega = 0.42$ is the gain constant, $\omega_0 = 0.606$ is the frequency of the dominant wave.

In the case of wind, wave and current interference, when the setting heading angle is 15°, the effect of traditional controller and predictive controller simulated using matlab is shown in the following figure.

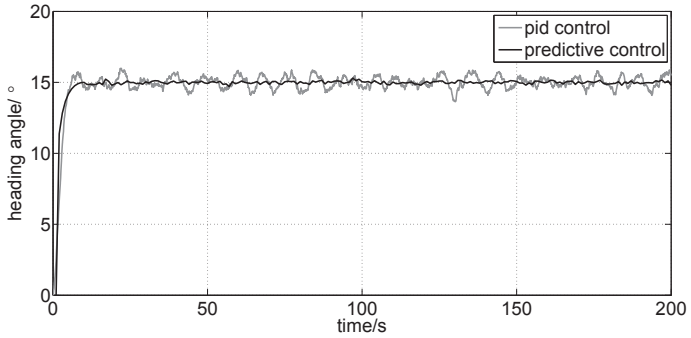


Figure 5: Heading using PID(Proportion Integration Differentiation) and the MPC

It is shown from Fig.5, Fig.6 that the predictive controller based on CNN has good control accuracy, robustness and fault tolerance. Compared with the traditional model predictive control, this controller's stability has become stronger with external disturbances, which can realize the stable path tracking with the characteristics of strong anti-interference. Finally, the problem of model mismatch in conventional generalized predictive control is also solved by this proposed method.

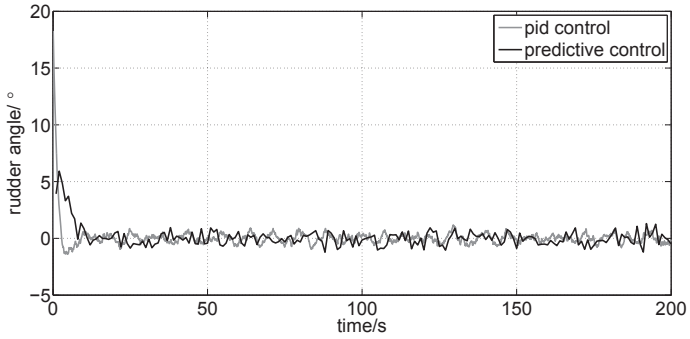


Figure 6: Rudder angle using PID and the MPC

Table 3: Latitude and longitude of the target point

target number	Coordinate	
	longitude	latitude
1	$120°08'03.7''E$	$31°26'31.4''N$
2	$120°08'07.7''E$	$31°26'27.9''N$
3	$120°08'06.5''E$	$31°26'23.1''N$
4	$120°08'02.3''E$	$31°26'21.0''N$
5	$120°07'57.4''E$	$31°26'24.3''N$
6	$120°07'59.0''E$	$31°26'29.4^{q}''N$

4.2 Experimental results

The predictive controller is applied to the WH-01, in which the course control experiment was carried out in Taihu Lake, Wuxi, Jiangsu province. Six target points are selected for the typical trajectory experiment in the lake, of which the latitude and longitude are shown in Table 2. The USV traveled at the average speed of 12 knot during the experiment.This simulation data is derived from the average of three experimental data.

Fig.7 and Fig.8 show the waypoints follow by the USV by the predictive controller and the PID scheme. Fig.9 and Fig.10 illustrate the distance of the vehicle from next waypoint and total time taken to operate under the predictive controller and the PID scheme. Fig.11 and Fig.12 demonstrate mean average deviation of the waypoints by the predictive controller and the PID scheme. It can be shown from Fig.7, Fig.8 and Table 4 that the USV can also pass through the target points when disturbed by wind and waves. It is observed that the predictive controller is

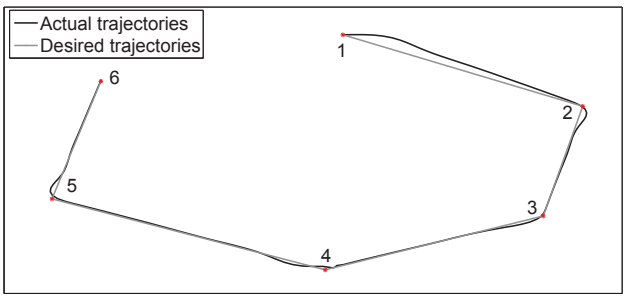


Figure 7: Waypoints followed by the Predictive controller systems

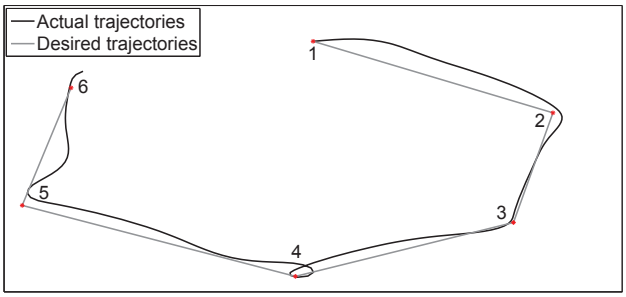


Figure 8: Waypoints followed by the PID GC systems

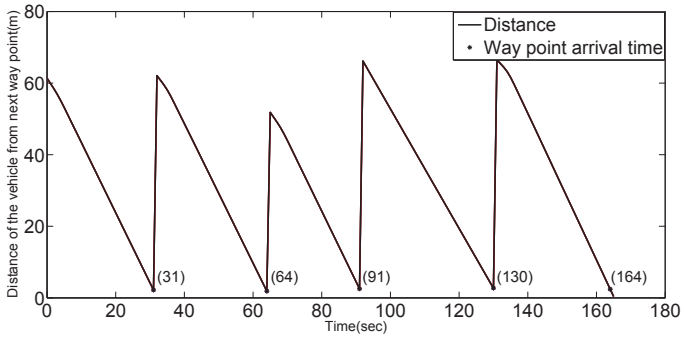


Figure 9: Distance of the vehicle from next waypoint and total time taken to operate by the Predictive controller systems

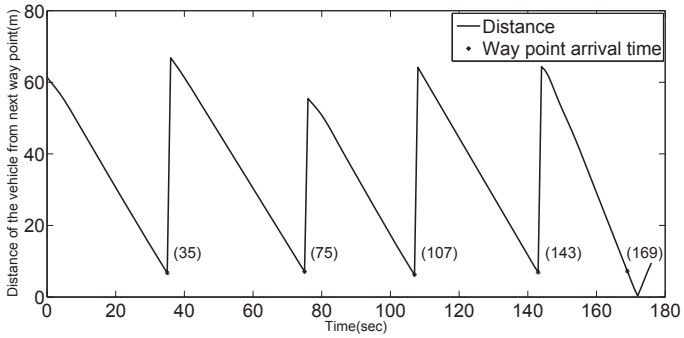


Figure 10: Distance of the vehicle from next waypoint and total time taken to operate by the PID GC systems(guidance and control systems)

Table 4: Average performance measures of predictive control

Case	Number of target points arriving	Total distance(m)	$\overline{rd}$ (m/s)
PID($K_p = 8, T_i = 8, T_d = 0$)	6	393.57	8.01
Predictive controller	6	347.98	4.23

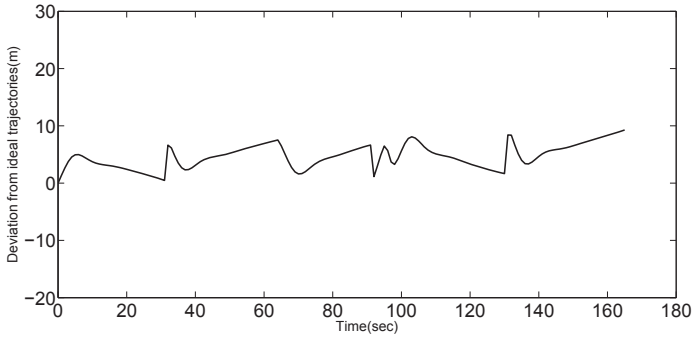


Figure 11: Mean average deviation of the waypoints by the Predictive controller systems

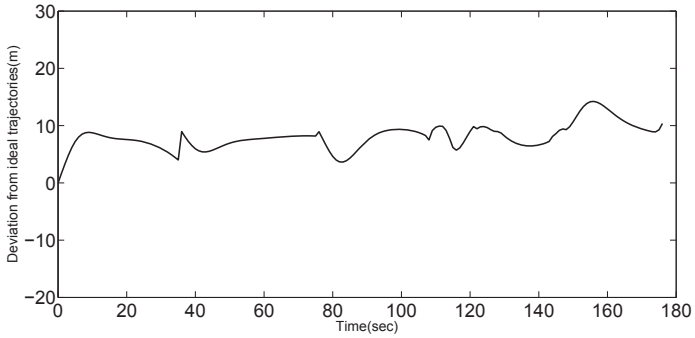


Figure 12: Mean average deviation of the waypoints by the PID GC systems

more efficient and capable to control the USV tracking the target points especially in the case of disturbances.

5 Conclusion

A predictive controller based on convolutional neural network(CNN) is designed for autopiloting an USV. It is shown from the experimental results that the predictive controller has good robustness and anti-jamming. In extreme sea condition, the problem of control overshooting and poor anti-interference is solved by the algorithm. In addition, the difficulty in hydrodynamic modeling can be solved by the proposed approach. Finally, the feasibility of the method is verified by numerical simulations and experiments on lake.

However, the steering mathematical model based on the CNN needs to be improved in the future. Particularly, the training samples accumulated from the model experiment, the actual ship test and expert knowledge need to be analyzed. In these conditions, it is difficult to build the nonlinear steering law, which is worthy of further research.

The next work will be related to research the optimal strategy and design a new predictive controller with good accuracy, adaptability and robustness. The influence of the designed parameters on the stability and other control performance can be analyzed in the system. The robustness of the predictive control system can be researched when there has modeling error and disturbance. Then, the different optimization strategies can be used to derive the different controller structure. Therefore, a new rolling optimization strategy needs to be researched.

Acknowledgement

The work was supported by National High-Tech Research and Development Program 2015AA015904.

Bibliography

[1] Abdolmalaki, R.Y.; Mahjoob, M.J.; Abbasi,E. (2013); Fuzzy LQR Controller for Heading Control of an Unmanned Surface Vessel, *International Workshops in Electrical-Electronics Engineering*, 2013.

[2] Annamalai, Andy S.K. (2014); *An adaptive autopilot design for an uninhabited surface vehicle*, PhD thesis, University of Plymouth, 2014.

[3] Brando, V.; Lovell, J.; King, E.; Boadle, D.; Scott, R.; Schroeder, T. (2016); The potential of autonomous ship-borne hyperspectral radiometers for the validation of ocean color radiometry data, *Remote Sensing*, 8(2), 150, 2016.

[4] Dou, C.X. (2003); Design of Fuzzy Neural Network Controller Based on Chaos Neural Network Forecast Model and Application, *Systems Engineering-theory & Practice*, 23(8), 48–52, 2003.

[5] 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.

[6] Fan, S.Y. (1988); *Ship maneuverability*, National Defense Industry Press, 1988.

[7] Fitzpatrick, P.J.; Lau, Y.;Moorhead, R.; Skarke, A.; Merritt, D.; Kreider, K.; Brown, C.; Carlon, R.; Hine, G.; Lampoudi, T.; Leonardi, A.P. (2007); A review of the 2014 Gulf of Mexico Wave Glider field program, *Marine Technology Society Journal*, 49(3), 64–71, 2007.

[8] Gao, F. (2012); *Design and Research of Key Technologies for a New AUV in Complex Sea Conditions*, PhD thesis, National University of Defense Technology, 2012.

[9] Jiang, L.; Mu, D.; Fan, Y.; Wang, G.; Zhao, Y. (2016); Study on USV model Identification and nonlinear course control, *Computer Measurement & Control*, 24, 133–136, 2016.

[10] Larrazabal, J.M.; Penas, M.S. (2016); Intelligent rudder control of an unmanned surface vessel, *Expert Systems with Applications*, 55, 106–117, 2016.

[11] Li, C.; Zhao, Y.; Wang, G. (2016); Adaptive RBF neural network control for USVcourse tracking, *International Conference on Information Science & Technology*, 285–290, 2016.

[12] Li, R. (2012); *Research and application on Generalized predictive control based on particle swarm optimization algorithm*, MSc thesis, LanZhou JiaoTong University, 2012.

[13] Liu, C.; Chu, X.; Wu, Q.; Wang, G. (2014); USV development status and prospects, *China Shipbuilding*, 194–205, 2014.

[14] Mcninch, L.C.; Muske, K.R.; Ashrafiuon, H. (2008); Model-based predictive control of an unmanned surface vessel, *IASTED International Conference on Intelligent Systems and Control*, 385–390, 2008.

[15] Mcninch, L.C.; Ashrafiuon, H. (2011); Predictive and sliding mode cascade control for Unmanned Surface Vessels, *American Control Conference*, 145(2), 184–189, 2011.

[16] Moe, S.; Pettersen, K.Y. (2016); Set-based Line-of-Sight (LOS) path following with collision avoidance for underactuated unmanned surface vessel, In *24th Mediterranean Conference Control and Automation*, 402–409, 2016.

[17] Mu, D.; Zhao, Y.; Wang, G.; Fan, Y.; Bai, Y. (2016); USV model identification and course control, *Sixth International Conference on Information Science and Technology*, 263–267, 2016.

[18] Mu, D.; Zhao, Y.; Wang, G.; Fan, Y.; Bai, Y. (2016); Course control of USV based on fuzzy adaptive guide control, *Control and Decision Conference*, 6433–6437, 2016.

[19] Ni, H. (2006); *Research on Predictive Control Method Based on Neural Network and Its Application*, MSc thesis, Central South University, 2006.

[20] Pan, L.; He, C.; Tian, Y.; Su, Y.; Zhang, X. (2017); A region division based diversity maintaining approach for many-objective optimization, *Integrated Computer-Aided Engineering*, 24(3), 1–18, 2017.

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

[22] Păun, G. (2016); Membrane Computing and Economics: A General View, *International Journal of Computers Communications & Control*, 11(1), 105-112, 2016.

[23] Shen, Y.; Cheng, Y.; Ji, Z. (2006); Controller Design for Asynchronism Motor Based on Multi-step Predictive Neural Network, *Small & Special Electrical Machines*, 34(12), 34–36, 2006.

[24] Sonnenburg, C.; Woolsey, C.A. (2012); An experimental comparison of two USV trajectory tracking control laws, *Oceans*, 1–10, 2012.

[25] Wang, C.S.; Xiao, H.R.; Han, Y.Z. (2013); Applications of ADRC in Unmanned Surface Vessel Course Tracking, *Applied Mechanics & Materials*, 427–429:897–900, 2013.

[26] Wang, Y.D. (2014); *Based on auto disturbance rejection control algorithm for course autopilot of unmanned surface vessel design*, MSc thesis, Dalian Maritime University, 2014.

[27] Wu, G.; Jin, Z.; Lei, W.; Qin, Z. (2009). Design of the Intelligence Motion Control System for the High-Speed USV, *Intelligent Computation Technology and Automation*, 3, 50–53, 2009.

[28] Yang, J.F. (2007); *Ant colony algorithm and its application research*, PhD thesis, Zhejiang University, 2007.

[29] Yang, L. (2013); *Analysis and Design of Simplified Dual Neural Network Based Model Predictive Controller*, MSc thesis, Shanghai Jiao Tong University, 2013.

INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 13(3), 440-455, June 2018.

Evidential Identification of New Target based on Residual

L. Zheng, Z. Zhang, Y. Deng

Lei Zheng, Zhiguo Zhang
College of Infomation Science and Technology
Jinan University, Guangzhou, China
LeiZhen@@stu2015.jnu.edu.cn, zhangzhiguo@stu2015.jnu.edu.cn

Yong Deng*
1. Big Data Decision Institute,
Jinan University
Tianhe, Guangzhou 510632, China
2. Institute of Fundamental and Frontier Science,
University of Electronic Science and Technology of China
Chengdu, 610054, China
*Corresponding author: prof.deng@hotmail.com

Abstract: Both incompleteness of frame of discernment and interference of data will lead to conflict evidence and wrong fusion. However how to identify new target that is out of frame of discernment is important but difficult when it is possible that data are interfered. In this paper, evidential identification based on residual is proposed to identify new target that is out of frame of discernment when it is possible that data are interfered. Through finding the numerical relation in different attributes, regress equations are established among various attributes in frame of discernment. And then collected data will be adjusted according to three mean value. Finally according to weighted residual it is able to decide whether the target requested to identify is new target. Numerical examples are used to verify this method.
Keywords: evidence theory, identification of new target, linear regression, residual.

1 Introduction

How to deal with uncertainty in real life is still an open issue [13, 29, 30, 54, 62, 66, 76]. Many math tools, such as fuzzy set [3, 25, 58, 64, 67, 71, 73], rough sets [19, 21, 36, 40, 41, 52], entropy function [1, 2, 49, 49, 70] and D numbers [4, 8, 33, 39, 75], are presented to address this issue [14, 16, 37, 65, 71, 74]. Among these efficient tools, Dempster Shafer evidence theory [5, 44] has been paid greatly attention recently. Since Dempster-Shafer theory (DS theory) is proposed [5, 44], it has been widely used in information fusion [9, 28, 32, 51, 60, 61], control systems [10, 31], uncertainty modelling [7, 19, 38, 46, 47, 59] decision making [26, 27, 35, 73], risk and reliability analysis [15, 22, 56, 63, 74] and other fields [6, 50, 68, 69]. In DS theory , a basic probability assignment(BPA) is distributed to power sets of the frame of discernment and the sum of BPA is always one when supposed that elements in the frame of discernment are mutually exclusive and exhaustive. However, in fact, the frame of discernment that we assume or have known is not complete, that is to say probably there are some unknown species in a world, which is called an open world [20, 45].It should be mentioned that an open issue in evidence theory is the conflict management [48, 53, 55, 72], which is also partially caused by open world.

And in past decades, a large amount of research has been conducted on this issue. Generalized evidential theory (GET) is proposed [11], which defines a novel concept called generalized basic probability assignment (GBPA) to model uncertain information, and provides a generalized combination rule (GCR) for the combination of GBPAs, and builds a generalized conflict model to measure conflict among evidences in an open world. Not only that, in [23], the conflict is

explored in a closed world and the result of evidence fusion is able to converge to correct answer when using the proposed method.

In addition, the identification of frame of discernment also is an open problem. Allocating a non-null value to the mass function on the empty is able to express and judge the incompleteness of frame of discernment in certain condition. As considering the case of potentially heterogeneous sources, Johan Schubert proposed a novel way to construct and evaluate alternative frames of discernment [43]. And dynamic estimation of the discernment frame in belief function theory is proposed by Wafa Rekik *et al.* [42] which using Cartesian product whose axes correspond to elementary discernment frames dealing with the relevance of each potential hypothesis to judge and update the frame of discernment.

However all of these works assume the collected data is correct and they are not interfered or just are interfered a little. In other words, only if the collected data are not interfered or just are interfered a little, the incompleteness of frame of discernment will be identified. In fact the collected data is very probably interfered. And the known species will be recognized as new target using existing method when the collected data is interfered. Therefore considering incompleteness identification under data interference condition have utilitarian value.

In this paper, evidential identification based on residual is proposed to identify incompleteness of frame of discernment. First assume that the frame of discernment is complete and there is a closed world. Then the method proposed in [11] is used to fuse data, which can get correct result even in evidence conflict in an closed world. Secondly through finding the numerical relation in different attributes, regress equations are established for different species and residual vector is calculated. The third step is that adjust collected data according to offset degree. The forth step is to plug the adjusted data into regress equation to calculate residual vector and weighted residual. Finally new target or perturbation of data is identified according to weighted residual. Numerical examples are used to introduce this method.

The rest of the paper is organized as follows. In Section 2, Dempster-Shafer theory, generalized evidence theory and evidence distance are briefly introduced. In Section 3, a new evidential identification based on residual is proposed. In section4 numerical examples are used to illustrate the behaviour of the new evidential identification based on residual. Section 5 concludes the main contribution of the paper.

2 Preliminaries

In this section, some preliminaries are briefly introduced below.

2.1 Dempster-Shafer theory of evidence [5, 44]

In Dempster-shafer theory, basic probability is distributed to power sets of the frame of discernment whose elements are mutually exclusive and exhaustive. Some terminology and notions are defined below to explain theory better.

Let Θ be a set of N mutually exclusive and exhaustive elements, which means the problem has N possible values. The following set is called the frame of discernment.

$$\Theta = \{H_1, H_2, ..., H_N\}. \tag{1}$$

$P(\theta)$ is the power set composed of 2^N elements A of θ, representing the object is in A.

$$\begin{aligned} P(\Theta) =& \{\phi, H_1, \cdots, H_N, (H_1, H_2), (H_1, H_3), ..., (H_{N-1}, H_N), \cdots, \\ & (H_1, H_2, H_3), \cdots, \Theta\}. \end{aligned} \tag{2}$$

A basic probability assignment (BPA) is a function from $P(\theta)$ to $[0,1]$ defined by:

$$m: \qquad P(\Theta) \to [0,1] \tag{3}$$

and which satisfies the following conditions:

$$\sum_{A \in P(\Theta)} m(A) = 1, \tag{4}$$

$$m(\phi) = 0. \tag{5}$$

where $m(A)$ represents the belief to A.

2.2 Generalized evidence theory

Generalized evidence theory is proposed by Deng [11], which generalize DS theory of evidence. When the frame of discernment is probably incomplete, generalized evidence theory is used to replace classical evidence theory, which defines a novel concept called generalized basic probability assignment (GBPA) to model uncertain information, and provides a generalized combination rule (GCR) for the combination of GBPAs, and builds a generalized conflict model to measure conflict among evidences in an open world [11]. But if the frame of discernment is complete, generalized evidence theory degrades into classical evidence theory.Some terminology and notions are defined below to explain theory better.

Let U be a frame of discernment in an open world, Which consists of N mutually exclusive elements.

$$U = \{H_1, H_2, ..., H_N\}. \tag{6}$$

And its power set 2_G^U is consisted of 2^U propositions, which contains empty set. For $\forall A \in U$, If the function $m: 2_G^U \to [0,1]$ satisfies

$$\sum_{A \in U} m_G(A) = 1 \tag{7}$$

the function m is called Generalized Basic Probability Assignment(GBPA). In this paper $m(\Phi)$ is not restricted to zero, which means the basic probability assignment to the proposition out of discernment. In other words, $m(\phi)$ represents the probability that the target is out of discernment.

2.3 Existing evidence distance

To measure the distance between two bodies of evidence, Jousselme defined a function from vector made up of BPAs to real number [24]. Let m_1 and m_2 be two BPAs on the same frame of discernment Θ, containing N mutually exclusive and exhaustive hypotheses. The distance between m_1 and m_2 is:

$$d_{BPA}(m_1, m_2) = \sqrt{\frac{1}{2}(\overrightarrow{m_1} - \overrightarrow{m_2})^T \underline{\underline{D}} (\overrightarrow{m_1} - \overrightarrow{m_2})}, \tag{8}$$

where $\overrightarrow{m_1}$ and $\overrightarrow{m_2}$ are the associated vectors of BPAs m_1 and m_2 and $\underline{\underline{D}}$ is a $2^N \times 2^N$ matrix whose elements are

$$D(A,B) = \frac{|A \cap B|}{|A \cup B|},$$
$$A, B \in P(\Theta).$$

3 The proposed evidential identification of new target based on residual

In this section evidential identification of new target based on residual is proposed. Like many ways of identifying the frame of discernment, at first assume the frame of discernment is compete and the world is closed. And then based on conflicts management in close world in [11], BPAs are produced and fusion result is obtained. Secondly through finding the numerical relation in different attributes, regress equations are established for different species and residual vector is calculated. The third step is that adjust collected data according to offset degree. The forth step is to plug the adjusted data into regress equation to calculate residual vector and weighted residual. Finally new target or perturbation of data is identified according to weighted residual. And the flow diagram is shown in Figure 1.

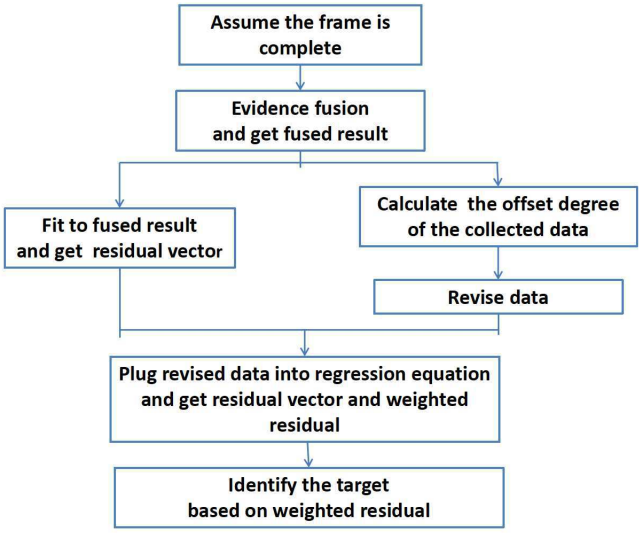


Figure 1: The flow diagram of new method

3.1 Evidence fusion

First suppose the frame of discernment is complete, conflicts management in close world. The method proposed in [11] is used to produce and combine BPAs. The reason why this method is chose is that this method can converge to correct answer even in strongly conflict condition when the frame of discernment is complete.

Furthermore when supposing that the frame of discernment is complete, many methods can be used to produce BPAs and fuse BPAs ,which will converge to correct target and show a great result.

3.2 Linear regression

In most cases, there is strong correlation among different attributes in the same species. Furthermore the correlation changes with species. For example, the relation expressions for heights and weights change with sex. Therefore the relation expression for various attributes of different species is a good tool to distinguish species. In this paper, relation expression will be established among different attributes for fused result. And it is well known that there are

many ways to represent the correlation for different attributes, such as covariance matrix, linear regression, nonlinear regression.

For simplifying linear regression is elected to represent the correlation among attributes which is able to take every attribute into account at the same time. In other words, the following equation is used to represent one species.

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = a \tag{9}$$

Where x_i represents the value of ith attribute in certain species, a_i represents the weight of ith attribute or is called ith regression coefficient in relation expression which can be calculated by least square method and a is arbitrary constant which is an artificial constant. In addition, the value of a just affects regression coefficient increasing or decreasing manifold at the same time.

In the process of linear regression, residual vector can be calculated. According to the vector, histogram and probability contribution function for residual are obtained.

Example 1. Take Iris data (http://archive.ics.uci.edu/ml/datasets/Iris) as an example. Suppose that *Setosa* is one of elements in frame of discernment and twenty pieces of sample data are known. There are four attributes in one sample which are sepal length, sepal width, petal length and petal width. The chosen data are shown in Figure 2

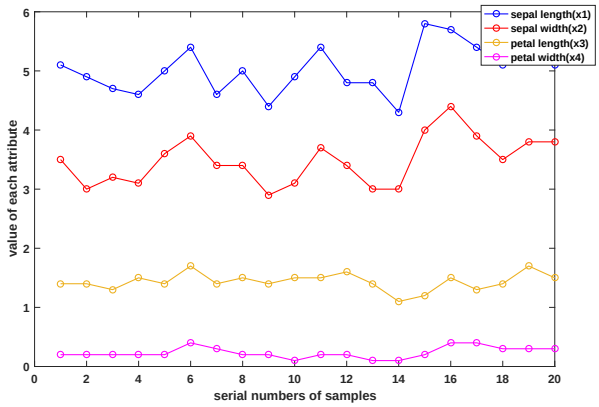


Figure 2: Twenty pieces of sample data which are used in liner regression

Let X_1 is the vector of value of sepal length, X_2 is the vector of value of sepal width, X_3 is the vector of value of petal length and X_4 is the vector of value of petal width, constant a is 100. Using least square method, the value of a_1, a_2, a_3, a_4 is calculated and linear regression of *Setosa* is obtained.

$$10.98x_1 + 4.827x_2 + 26.98x_3 - 47.89x_4 = 100. \tag{10}$$

If setting constant a is 1000 the linear regression of *Setosa* is

$$109.8x_1 + 48.27x_2 + 169.8x_3 - 478.9x_n = 1000. \tag{11}$$

The regression coefficient of the second equation is just ten times of the first equation when the second constant a is ten times of the first constant. And in the process of linear regression, residual vector can be calculated. And histogram and probability contribution function for residual are shown in Figure 3.

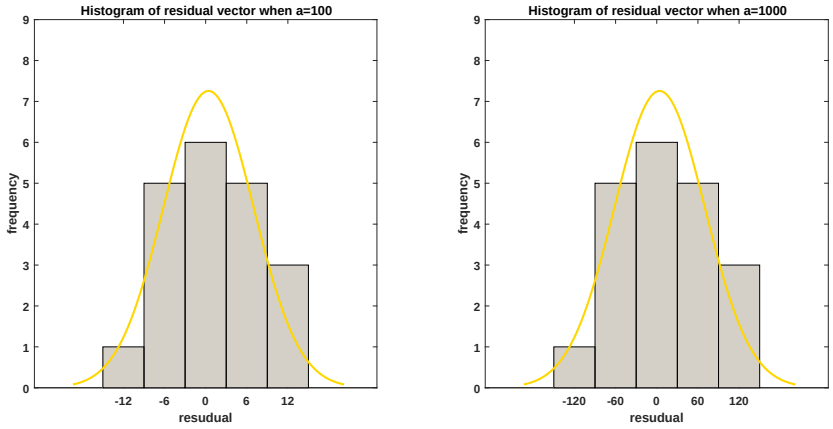


Figure 3: Histogram of residual vector when a=100 and a=1000

3.3 Data processing

If the target requested to identify is one of the elements of frame discernment and the data are not disturbed, the collected data will satisfy Eq.(9) equation or the residual will be small. Not only that, when revising noisy data and take it into Eq.(9), the revised data will satisfy Eq.(9) or the residual will be small. At the same time, if the target requested to identify is out of frame, after the same data revising, the revised data will not satisfy Eq.(9) or the residual will be large.

But how to revise data to satisfy these conditions mentioned above?

First offset degree is defined as follows. Let frame of discernment

$$\theta = \{H_1, H_2, \cdots, H_n\},$$

the target which is requested to identify is S. S_j represents jth attribute for S, and H_{ij} represents jth attribute for element H_i.

$$t_j = \frac{threemean(S_j)}{threemean(H_{ij})} \tag{12}$$

$$threemean(A) = \frac{1}{4}Q_1 + \frac{1}{2}M + \frac{1}{4}Q_3 \tag{13}$$

where t_j is defined as jth attribute offset degree for target requested to identify S. A is a vector, Q_1 is upper quartile for vector A, M is mean for vector A, and Q_3 is lower quartile for vector A.

Because threemean have disturbance rejection for extremum and abnormal data have no effect on offset degree, it is appropriate to use threemean to calculate offset degree.

And then revised data is defined as follows.

$$\widehat{S_j} = \frac{1}{t_j} \times S_j \tag{14}$$

Because when taking the fusion method in [11], if the target is one of element in frame of discernment, it can diverge to correct solution. It is enough to calculate offset degree between target requested to identify and fused result and revise collected data according to fused result. It is no need to calculate offset degree between target requested to identify and every elements in frame of discernment.

Next plug these revised data into linear regression Eq.(9) and obtain residual r.

$$r = a - (a_1 H_1 + a_2 H_2 + \cdots + a_n H_n) \tag{15}$$

3.4 Residual vector processing

Let R represent residual vector of collected data of target requested to identify, and the ith element of R , $R(i)$, represents the residual between the ith collected sample data and linear regression of fusion result calculated in (9). If there is extremum in collected data, the part of component of vector will be large. To avoid the effect of extremum, weighted residual is chose. In general condition, the more similar the samples data are, the more confidence will be given, the same as the process of weighted fusion. For simplifying, the weight proposed in [12] consist of evidence distance serves as weight to produce weighted residual.

As we know the less weighted residual is, the more probably the target requested to identify is the convergence. When setting α as rejection probability, the section that locate at both ends in probability density function will be rejected as well as Figure 4.

Therefore according to given rejection probability, residual probability density function or

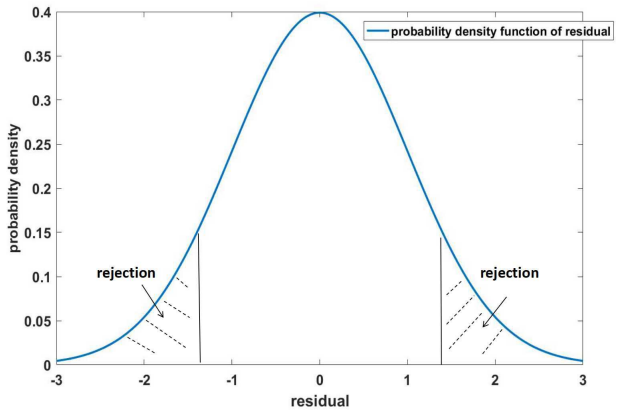


Figure 4: Rejection method according to probability density function

diagram and weighted residual, whether the target requested to identify is a new target is able to identify.

4 Numerical example

Real world exist many uncertainty [17, 34, 57]. In this section numerical examples are used to illustrate the validity of evidential identification of new target based on residual.

Take Iris data (http://archive.ics.uci.edu/ml/datasets/Iris) as an example. There are 150 samples in Iris data and each sample has four attributes such as sepal length, sepal width, petal length and petal width. In addition, 150 samples are divided into three classes which are *Setosa*, *Versicolour* and *Virginica* and there are 50 samples in each class.

Let frame of discernment consist of *Setosa* and *Virginica* $\theta = \{Setosa, Virginica\}$, and *Versicolour* is out of frame. To certify the practical of this method, two different conditions are considered. First condition is that the target requested to identify is *Versicolour* which is out of frame. Second condition is that the target requested to identify is *Virginica* whose collected data are interfered.

first condition, select 10 samples in *Versicolour* at random as collected data

Step 1. evidence fusion

According to method proposed in [11] and collected data, 40 pieces BPA are reduced shown in Figure 5 and evidence distance is obtained.

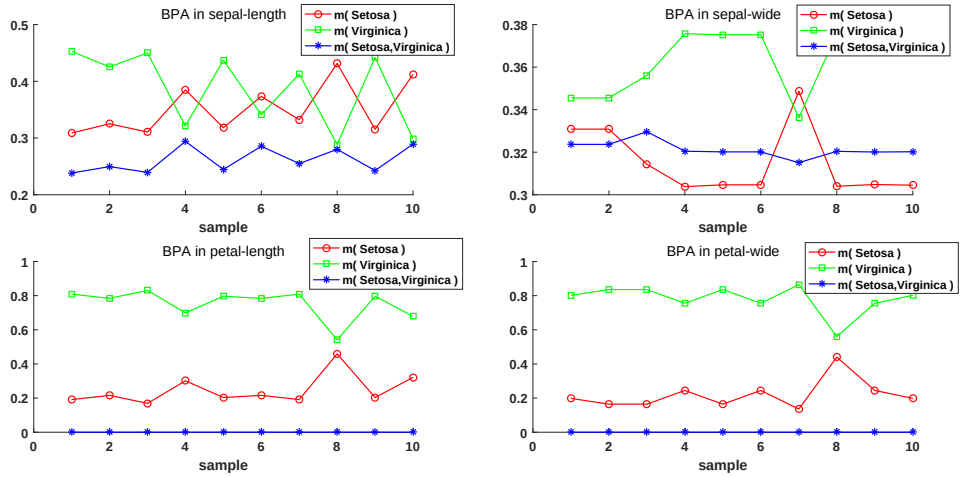


Figure 5: BPAs of collected data

And then fuse these BPAs and result converges to *Virginica*.

Step 2. linear regression

The Eq. (9) is chose to reflect the relation among four attributes of *Virginica*. where x_i represent the value of ith attribute in *Virginica* and a_i represent the weight of ith attribute or is called ith regression coefficient .

Based least square method, parameters in linear regression are calculated and the relational expression is shown below.

$$5.15x_1 + 9.25x_2 + 4.36x_3 + 6.75x_4 = 100 \tag{16}$$

At same time the residuals of each sample can be obtained and represented by histogram shown in Figure 6. According to residual vector, the average of absolute residuals is 6.73.

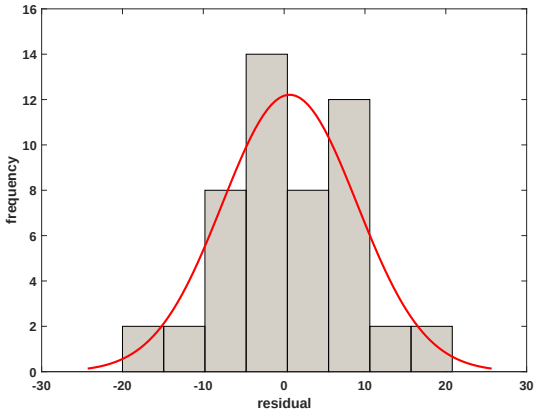


Figure 6: Residual distribution histogram

And when the absolute residual is larger than 7, there is seventy percent that the target requested to identify is not *Virginica*.

Step 3. data processing

Offset degree is calculated as below. Where S_i represent the vector of ith attribute for target, H_i represent the vector of ith attribute for *Virginica*.

$$t_1 = \frac{threemean(S_1)}{threemean(H_1)} = 0.95$$
$$t_2 = \frac{threemean(S_2)}{threemean(H_2)} = 0.96$$
$$t_3 = \frac{threemean(S_3)}{threemean(H_3)} = 0.81$$
$$t_4 = \frac{threemean(S_4)}{threemean(H_4)} = 0.69$$

And then according to offset degree of each attribute, data are revised.

$$\widehat{S_i} = \frac{1}{t_i} \times S_i \, (i = 1, 2, 3, 4)$$
$$\widehat{S} = [\widehat{S_1}, \widehat{S_2}, \widehat{S_3}, \widehat{S_4}]$$

Step 4. residual vector processing

Take the revised data into linear regression Eq.(16) ten pieces of residual vector can be obtained.The figure of residual vector is shown in Figure 7. And according to [12] the weighted residual 8.3 can be obtained as well.

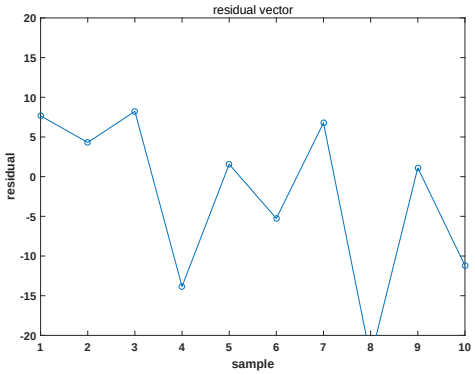


Figure 7: Residual vector of collected data

Because 8.3 is larger than 6.73 and 7, the probability that target requested to identify is *Virginica* is not larger than seventy percent. Therefore based on the collected data, target requested to identify is regarded as a new species.

Above all, this method can identify new target.

In the second condition, choose 10 samples in *Virginica* at random and the sample data becomes 1.25 times.

Step 1. evidence fusion

According to method proposed in [11] and collected data, 40 pieces BPA are reduced shown in Figure 8 and evidence distance is obtained.

And then fuse these BPAs and result converges to *Virginica*.

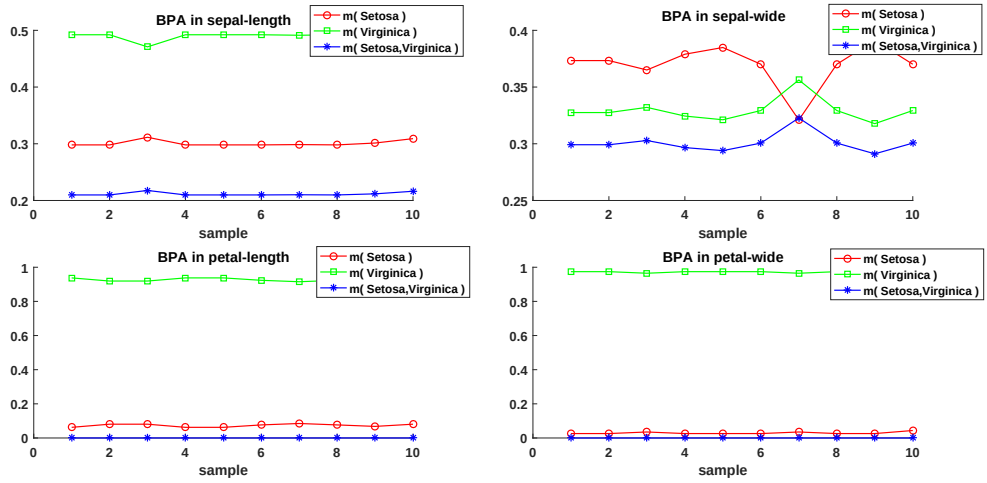


Figure 8: BPAs of collected data

Step 2. linear regression
The following equation is chose to reflect the relation among four attributes of *Virginica*.

$$5.15x_1 + 9.25x_2 + 4.36x_3 + 6.75x_4 = 100 \tag{17}$$

Step 3. data processing
Offset degree is calculated.

$$t_1 = 1.25, t_2 = 1.28, t_3 = 1.19, t_4 = 1.35$$

And then according to offset degree of each attribute, data are revised.

Step 4. residual vector processing
Take the revised data into linear regression Eq.(17) ten pieces of residual vector can be obtained.The figure of residual vector is shown in Figure (9). And according to [12]the weighted residual 4.9 can be obtained as well.

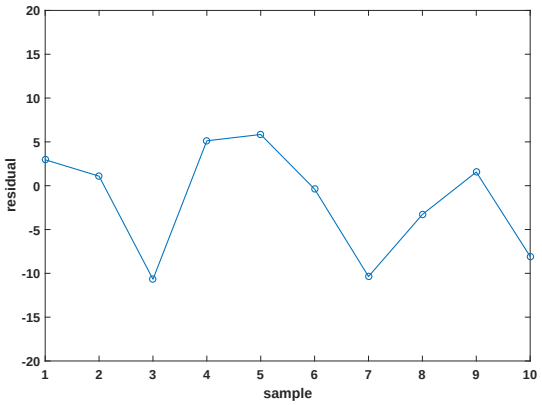


Figure 9: Residual vector of collected data

Because 4.9 is less than 7 and 6.73, the result that target requested to identify is *Virginica* is not rejected.

Above all, when the data are interfered largely, target requested to identify is able to be identified as known species.

5 Conclusion

How to identify new target is a significant problem when it probably exists interference. However all of these works assume the collected data is correct and they are not interfered or just interfered a little. In other words, if the collected data are interfered, the identification of new target probably is wrong. The evidential identification of new target based on residual proposed in this paper considers the probability that data are interfered when identifying whether the target is new. And through data fusion, linear regression, data processing and residual vector processing, weighted residual can be obtained which is able to identify whether it is new target. It considers correlation among different attributes. By numerical example, efficiency and practicability of this method are proved. However in this method the setting of rejection probability which affects the accuracy of result is subjective. Therefore how to find a objective way to identify rejection probability will be significant research indicators of further studies. Also for simplifying, liner regression is elected to represent the correlation among attributes, while other method, such as nonlinear regression, will improve the accuracy.

Acknowledgment

The work is partially supported by National Natural Science Foundation of China (Grant Nos. 61573290, 61503237).

Conflict of interests

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

Bibliography

[1] Abellán, J., Mantas, C. J., Castellano, J. G. (2017): A random forest approach using imprecise probabilities, *Knowledge-Based Systems*, 134, 72–84, 2017.

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

[3] Akyar, H. (2016): Fuzzy risk analysis for a production system based on the nagel point of a triangle. *Mathematical Problems in Engineering*,2016,(2016-3-31) 2016 (4), 1–9.

[4] Bian, T., Zheng, H., Yin, L., Deng, Y. (2018): Failure mode and effects analysis based on D numbers and topsis. *Quality and Reliability Engineering International*, Article ID: QRE2268.

[5] Dempster, A. P. (1967): Upper and lower probabilities induced by a multivalued mapping. *Annals of Mathematics and Statistics*, 38 (2), 325–339, 1967.

[6] Deng, W., Lu, X., Deng, Y. (2018): Evidential Model Validation under Epistemic Uncertainty. *Mathematical Problems in Engineering*, Article ID 6789635, 2018.

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

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

[9] Deng, X., Jiang, W. (2018): Dependence assessment in human reliability analysis using an evidential network approach extended by belief rules and uncertainty measures. *Annals of Nuclear Energy*, 117, 183–193, 2018.

[10] Deng, X., Jiang, W. (2018): An evidential axiomatic design approach for decision making using the evaluation of belief structure satisfaction to uncertain target values. *International Journal of Intelligent Systems*, 33 (1), 15–32, 2018.

[11] Deng, Y. (2015): Generalized evidence theory, *Applied Intelligence*, 43 (3), 530–543, 2015.

[12] Deng, Y., Shi, W. K., Zhu, Z. F., Liu, Q. (2004): Combining belief functions based on distance of evidence, *Decision Support Systems*, 38 (3), 489–493, 2004.

[13] Dong, Y., Wang, J., Chen, F., Hu, Y., Deng, Y. (2017): Location of facility based on simulated annealing and ZKW algorithms, *Mathematical Problems in Engineering*, Article ID 4628501, 2017.

[14] 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.

[15] 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.

[16] Goyal, R. K., Kaushal, S. (2016): A constrained non-linear optimization model for fuzzy pairwise comparison matrices using teaching learning based optimization. *Applied Intelligence*, 1–10, 2016.

[17] Hu, Y., Du, F., Zhang, H. L., (2016): Investigation of unsteady aerodynamics effects in cycloidal rotor using RANS solver. *Aeronauthical Journal* 120 (122), 956–970, 2016.

[18] Inglis, J. (1977): A mathematical theory of evidence, *Technometrics*, 20 (1), 242, 1977.

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

[20] Jiang, W., Wang, S., Liu, X., Zheng, H., Wei, B. (2017): Evidence conflict measure based on OWA operator in open world. *PloS ONE*, 12 (5), e0177828, 2017.

[21] Jiang, W., Wei, B. (2018): Intuitionistic fuzzy evidential power aggregation operator and its application in multiple criteria decision-making. *International Journal of Systems Science*, 49 (3), 582–594, 2018.

[22] 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.

[23] Jiang, W., Zhan, J. (2017): A modified combination rule in generalized evidence theory. *Applied Intelligence*, 46 (3), 630–640, 2017.

[24] Jousselme, A. L., Grenier, D., loi Boss (2001): A new distance between two bodies of evidence. *Information Fusion*, 2 (2), 91–101, 2001.

[25] Kahraman, C., Onar, S. C., Oztaysi, B. (2015): Fuzzy multicriteria decision-making: A literature review. *International Journal of Computational Intelligence Systems*, 8 (4), 637–666, 2015.

[26] 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.

[27] Kang, B., Chhipi-Shrestha, G., Deng, Y., Mori, J., Hewage, K., Sadiq, R. (2018): Development of a predictive model for Clostridium difficile infection incidence in hospitals using Gaussian mixture model and Dempster-Shafer theory, *Stochastic Environmental Research and Risk Assessment*, 32(6), 1743-1758, 2018.

[28] Kang, B., Deng, Y. (2018): Generating Z-number based on OWA weights usingmaximum entropy, *International Journal of Intelligent Systems*, https://doi.org/10.1002/int.21995, 2018.

[29] 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.

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

[31] Li, F., Zhang, X., Chen, X., Tian, Y. C. (2017): Adaptive and robust evidence theory with applications in prediction of floor water inrush in coal mine, *Transactions of the Institute of Measurement & Control*, 39 (4), 2017.

[32] Li, Y., Chen, J., Ye, F., Liu, D., (2016). The Improvement of DS Evidence Theory and Its Application in IR/MMW Target Recognition, *Journal of Sensors*, (1903792), 2016.

[33] Liu, B., Hu, Y., Deng, Y. (2018): New failure mode and effects analysis based on d numbers downscaling method. *International Journal of Computers Communications & Control*, 13 (2), 205-220, 2018.

[34] Liu, J., Lian, F., Mallick, M., (2016): Distributed compressed sensing based joint detection and tracking for multistatic radar system, *Information Sciences*, 369, 100–118, 2016.

[35] Liu, T., Deng, Y., Chan, F. (2017): Evidential supplier selection based on DEMATEL and game theory, *International Journal of Fuzzy Systems*, DOI: 10.1007/s40815–017–0400–4, 2017.

[36] Liu, W., Liu, H. B., Li, L. L. (2017): A multiple attribute group decision making method based on 2-d uncertain linguistic weighted heronian mean aggregation operator, *International Journal of Computers Communications & Control*, 12(2):254-264, 2017.

[37] Mardani, A., Jusoh, A., Zavadskas, E. K. (2015): Fuzzy multiple criteria decision-making techniques and applications - two decades review from 1994 to 2014. *Expert Systems with Applications*, 42 (8), 4126–4148, 2015.

[38] 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), 2016.

[39] Mo, H., Deng, Y. (2016): A new aggregating operator in linguistic decision making based on D numbers. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 24 (6), 831–846, 2016.

[40] Pawlak, Z. (1982): Rough sets. *International Journal of Computer & Information Sciences*, 11 (5), 341–356, 1982.

[41] Pedrycz, W., Al-Hmouz, R., Morfeq, A., Balamash, A. S. (2014): Building granular fuzzy decision support systems. *Knowledge-Based Systems*, 58, 3–10, 2014.

[42] Rekik, W., Hegarat-Mascle, S. L., Reynaud, R., Kallel, A. (2015): Dynamic estimation of the discernment frame in belief function theory, *International Conference on Information Fusion*, 1135–1142, 2015.

[43] Schubert, J. (2012): *Constructing and evaluating alternative frames of discernment*, Elsevier Science Inc., 2012.

[44] Shafer, G. (1976): *Mathematical Theory of Evidence*, Princeton University Press, Princeton, 1976.

[45] Smets, P. (1990): The combination of evidence in the transferable belief model. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 12 (5), 447–458, 1990.

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

[47] Song, Y., Wang, X., Lei, L., Yue, S., (2016): Uncertainty measure for interval-valued belief structures, *Measurement*, 80, 241–250, 2016.

[48] Wang, J., Qiao, K., Zhang, Z., Xiang, F. (2017): A new conflict management method in dempster¨cshafer theory. *International Journal of Distributed Sensor Networks*, 13, 3(2017-3-01) 13 (3), 155014771769650, 2017.

[49] Wen Jiang, Boya Wei, X. L. X. L., Zheng, H. (2017): Intuitionistic fuzzy power aggregation operator based on entropy and its application in decision making, *International Journal of Intelligent Systems*, https://doi.org/10.1002/int.21939, 2017.

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

[51] Xiao, F. (2017): An improved method for combining conflicting evidences based on the similarity measure and belief function entropy. *International Journal of Fuzzy Systems*, DOI: 10.1007/s40815–017–0436–5, 2017.

[52] Xiao, F. (2017): A novel evidence theory and fuzzy preference approach-based multi-sensor data fusion technique for fault diagnosis. *Sensors*, 17 (11), DOI: 10.3390/s17112504, 2017.

[53] Xiao, F., Aritsugi, M., Wang, Q., Zhang, R. (2016): Efficient processing of multiple nested event pattern queries over multi-dimensional event streams based on a triaxial hierarchical model. *Artificial Intelligence in Medicine*, 72 (C), 56–71, 2016.

[54] Xiao, F., Zhan, C., Lai, H., Tao, L., Qu, Z. (2017): New parallel processing strategies in complex event processing systems with data streams. *International Journal of Distributed Sensor Networks*, 13 (8), 1–15, 2017.

[55] Xiao, F., Zhan, C., Lai, H., Tao, L., Qu, Z. (2017: New parallel processing strategies in complex event processing systems with data streams. *International Journal of Distributed Sensor Networks*, 13 (8), 1–15, 2017.

[56] Xu, H., Deng, Y. (2018): Dependent evidence combination based on shearman coefficient and pearson coefficient. *IEEE Access*, 6 (1), 11634–11640, 2018.

[57] Xu, S., Jiang, W., Deng, X., Shou, Y. (2017): A modified physarum-inspired model for the user equilibrium traffic assignment problem. *Applied Mathematical Modelling*, In Press, DOI: 10.1016/j.apm.2017.07.032, 2017.

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

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

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

[61] Ye, F., Chen, J., Li, Y., Kang, J., (2016). Decision-Making Algorithm for Multisensor Fusion Based on Grey Relation and DS Evidence Theory, *Journal of Sensors*, Article ID 3954573, http://dx.doi.org/10.1155/2016/3954573, 2016.

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

[63] 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. *Journal of Risk & Reliability*, 230 (6), 2016.

[64] Zadeh, L. A. (2011): A note on Z-numbers, *Information Sciences*, 181 (14), 2923–2932, 2011.

[65] Zavadskas, E. K., Antuchevicience, J., Hajiagha, S. H. R. (2015): The interval-valued intuitionistic fuzzy multimoora method for group decision making in engineering, *Mathematical Problems in Engineering*, 560690, 2015.

[66] Zavadskas, E. K., Antucheviciene, J., Turskis, Z., Adeli, H., (2016): Hybrid multiple-criteria decision-making methods: A review of applications in engineering. *Scientia Iranica*, 23 (1), 1–20, 2016.

[67] Zhang, D., 2017. High-speed train control system big data analysis based on the fuzzy rdf model and uncertain reasoning. *International Journal of Computers Communications & Control*, 12 (4), 577–591, 2017.

[68] Zhang, L., Wu, X., Qin, Y., Skibniewski, M. J., Liu, W. (2016). Towards a Fuzzy Bayesian Network Based Approach for Safety Risk Analysis of Tunnel-Induced Pipeline Damage. *Risk Analysis*, 36 (2), 278–301, 2016.

[69] Zhang, L., Wu, X., Zhu, H., AbouRizk, S. M. (2017). Perceiving safety risk of buildings adjacent to tunneling excavation: An information fusion approach. *Automation in Construction*, 73, 88–101, 2017.

[70] 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.

[71] 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*, DOI: 10.1007/s11634–017–0300–3, 2017.

[72] Zhao, Y., Jia, R., Shi, P. (2016): A novel combination method for conflicting evidence based on inconsistent measurements. *Information Sciences*, 367-368, 125–142, 2016.

[73] Zheng, H., Deng, Y., (2018): Evaluation method based on fuzzy relations between Dempster-Shafer belief structure, *International Journal of Intelligent Systems*, https://doi.org/10.1002/int.21956, 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] Zhou, X., Deng, X., Deng, Y., Mahadevan, S. (2017): Dependence assessment in human reliability analysis based on D numbers and AHP. *Nuclear Engineering and Design*, 313, 243–252, 2017.

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

Author index