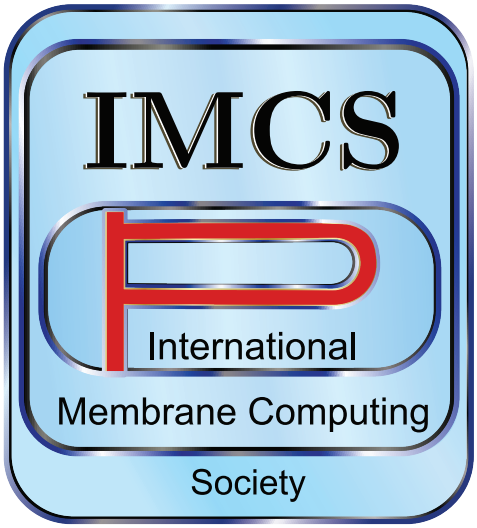
B U L L E T I N

of the International Membrane Computing Society

I M C S



Number 5 June 2018

Bulletin Webpage:
http://membranecomputing.net/IMCSBulletin
Webmaster: Andrei Florea, andrei91ro@gmail.com

Foreword

We have here the fifth issue of our *Bulletin*!

In December 2017, I have announced that Marian Gheorghe (Bradford, UK, m.gheorghe@bradford.ac.uk) will take the task of editing the *IMCS Bulletin*, but, because of some health problems, the change of editors was delayed at least with this volume.

The *Bulletin* follows the "standard" contents of the previous volumes, but there are two points to be emphasized.

First, the fact that a book on membrane computing was published in Japan, by Taishin Y. Nishida, *Cell Membrane Computing*, presented in the *New Books* column below. It is important to mention that the book contains both theoretical chapters, as well as two chapters dealing with applications, one discussing a model of photosynthesis and one dealing with membrane algorithms, two research directions where the author of the book, Taishin Y. Nishida, has pioneering contributions.

The second pleasant observation concerns the abundance of materials provided by the *Open Problems, Research Topics* column, which indicates that still there is a lot to do in membrane computing.

Like in the previous volumes, I would like to stress the fact that the *IMCS Bulletin* is conceived as a working material for the MC community, as a medium for communicating in a fast and efficient way any idea, news, problem, result. As it is already known, each issue gradually grows and remains available at http://membranecomputing. net/IMCSBulletin), also being printed. (If somebody wants to have a printed copy, s/he has to contact the IMCS secretary – see information about the structure of IMCS, including email addresses, in the subsequent pages.)

The "instructions to contributors" are minimal. Any material which any MC researcher considers of interest for the community, helping in achieving the goals of IMCS, is very much welcome and can be submitted at any time to me or to any member of the Bulletin Committee. In what concerns the style and format, the

previous issues of the *Bulletin* are available as a model. (LNCS Latex style is the best to be used.)

The copyright of all materials remains with their authors.

I am hereby inviting all people interested in membrane computing to consult the *Bulletin* and, also, to contribute – "views from outside" are always of interest and useful.

*

The realization of the *Bulletin of IMCS* owns very much (i) to all contributors, (ii) to the webmaster, Andrei Florea, andrei91ro@gmail.com, and (iii) to prof. Gexiang Zhang, the President of IMCS, to his group, and to the Xihua University in Chengdu, China, where the *Bulletin* is printed.

Gheorghe Păun
June 2018

Contents

Letter from The President

Dear IMCS members,

I am pleased to tell you a piece of good news, that is, the web site (`imcs.org.cn`) of IMCS is open to everyone after a long period of effort. The "chapters" of the page are under construction and any suggestion/comment for improving the web site is welcome. I would like to take this opportunity to express my sincere gratitude to the Website Committee Chair, Dr. Xiangxiang Zeng, Xiamen University, China.

In April, the 3rd Chinese Workshop on Membrane Computing (CWMC 2018) was successfully held in Xiamen, a very beautiful city in the east-south of China. During the workshop, two main ideas have been delivered to the participants. On the one hand, the collaborations are emphasized across research groups and with the researchers outside of the membrane computing community, especially the cooperative preparation of the submissions for the coming Asian Branch of International Conference on Membrane Computing (ACMC2018), which will be held at The University of Auckland, 10-16 December, New Zealand. A special case appears this year that three festschrift special issues/volumes have been organized to celebrate three big persons' birthday: Academician Kamala Krithivasan, Academician Mario J. Pérez-Jiménez and professor Kumbakonam Govindarajan Subramanian. On the other hand, the real life applications of membrane computing are highlighted. The current situation is that it is very difficult to obtain the funds for theoretical computer science, due to the economy resulting from Euro crisis. In response to the recommendation from Professor Kumbakonam Govindarajan Subramanian, I am invited by Professor Ibrahim Venkat to deliver a short term courses/workshop to the participants from the whole Malaysia in August at Universiti Sains Malaysia. The focus of the workshop is on the real life applications with membrane computing.

The new journal and also the first membrane computing journal, ***Journal of Membrane Computing (JMC)***, has started to call for high-quality papers. Both the journal web site, `www.springer.com/41965`, and the submission web

site, www.editorialmanager.com/JMEC, are open now. There are 4 issues each year. The first issues will appear online in the fall of this year and will be officially published in 2019 by Springer. All the papers published in JMC have free access in 2019 and 2020. Our goal is to create and run a high-profile journal. Our ambitious short-term goal is to let it be indexed by Institute of Information Sciences (ISI) in the first round of evaluation after three years' publication.

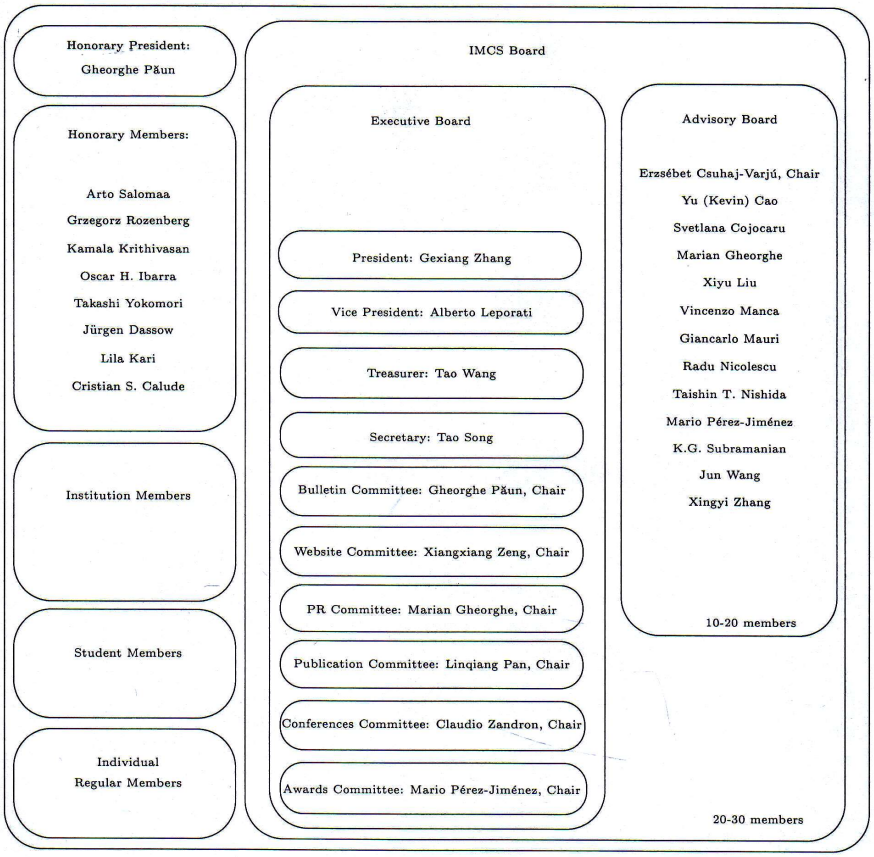
The JMC provides a forum for developing and nurturing an international community of scholars and practitioners who are interested in all aspects of membrane computing: theories, interdisciplinary areas, applications, and implementations. JMC welcomes high-quality submissions that contribute to the full range of membrane computing research, from cell-like P systems, tissue P systems, spiking neural P systems, and other types of P systems, to membrane algorithms, computational complexity, interdisciplinary research combining membrane computing and evolutionary computing and neural networks, to applications like optimization and biosystem modeling, and membrane computing implementations with nanotechnology. This uniquely broad range facilitates the cross-fertilization of ideas between biological and technological studies, and helps to spur on the advancement of an interdisciplinary community that is interested in biologically inspired computational intelligence. Accordingly, the JMC editorial board, consisting of Honorary Editor-in-Chief, Editor-in-Chief, Managing Editor and members, represents experts in a range of fields, including theoretical computer science, engineering, mathematics, and nanotechnology. The journal publishes original, high-quality and previously unpublished research papers, survey and review articles, short communications, and tutorial papers. (See https://www.springer.com/computer/theoretical+computer+science/journal/41965.)

Finally, I would like to thank our Bulletin Chair, Academician Gheorghe Păun, and his committee members, for their efficient work in the past two years. Special thanks are given to all the contributors. Academician Păun has set a very good example to start and run the Bulletin. I also wish a happy birthday to Academician Kamala Krithivasan, Academician Mario J. Pérez-Jiménez, and professor Kumbakonam Govindarajan Subramanian.

Gexiang Zhang
Chengu, China
June 19, 2018

IMCS Matters

Structure of IMCS



The Board of IMCS

The Executive Board:

President: Gexiang Zhang, China, gexiangzhang@gmail.com
Vice President: Alberto Leporati, Italy, alberto.leporati@unimib.it
Treasurer: Tao Wang, China, wangtao2005@163.com
Secretary: Tao Song, China, taosong@hust.edu.cn, songtao0608@hotmail.com
Bulletin Committee Chair: Gheorghe Păun, Romania, gpaun@us.es
Website Committee Chair: Xiangxiang Zeng, China, xzeng@xmu.edu.cn
PR Committee Chair: Marian Gheorghe, U.K. m.gheorghe@bradford.ac.uk
Publication Committee Chair: Linqiang Pan, China, lqpanhust@gmail.com
Conferences Committee Chair: Claudio Zandron, Italy, zandron@disco.unimib.it
Awards Committee Chair: Mario Pérez-Jiménez, Spain, marper@us.es

The Advisory Board:

Erszébet Csuhaj-Varjú, Hungary – Chair, csuhaj@inf.elte.hu
Yu (Kevin) Cao, U.S.A.
Svetlana Cojocaru, Rep. Moldova
Marian Gheorghe, U.K.
Xiyu Liu, China
Vincenzo Manca, Italy
Giancarlo Mauri, Italy
Radu Nicolescu, New Zealand
Taishin T. Nishida, Japan
Mario Pérez-Jiménez, Spain
K.G. Subramanian, India
Jun Wang, China
Xingyi Zhang, China

Honorary President:

Gheorghe Păun, Romania

Honorary Members:

Arto Salomaa, Finland
Grzegorz Rozenberg, The Netherlands
Kamala Krithivasan, India
Oscar H. Ibarra, U.S.A.
Takashi Yokomori, Japan
Jürgen Dassow, Germany
Lila Kari, Canada
Cristian S. Calude, New Zealand

Bulletin Committee

Website Committee

PR Committee

Marian Gheorghe, Bradford, U.K. – Chair, M.Gheorghe@bradford.ac.uk
Petros Kefalas, Sheffield, U.K. (International Faculty, Greece), kefalas@city.academic.gr
Savas Konur, Bradford, U.K., S.Konur@bradford.ac.uk
Maciej Koutny, Newcastle, U.K., maciej.koutny@newcastle.ac.uk
Jianhua Xiao, Nankai, China, jhxiao@nankai.edu.cn

Publication Committee

Linqiang Pan, Wuhan, China – Chair, lqpanhust@gmail.com
Marian Gheorghe, Bradford, U.K., m.gheorghe@bradford.ac.uk
Alberto Leporati, Milan, Italy, alberto.leporati@unimib.it
Gheorghe Păun, Bucharest, Romania, gpaun@us.es
Mario Pérez-Jiménez, Seville, Spain, marper@us.es
Gexiang Zhang, Chengdu, China, zhgxdylan@126.com

The main tasks of the Publication Committee are (1) to explore the possibility to initiate a series of MC monographs/collective volumes, (2) to establish a MC international journal, (3) to advise the organizers of CMC, ACMC, BWMC, MC workshops in what concerns the special issues of journals, (4) to help translating MC books in Chinese.

The Publication Committee can become the Editorial Board of the MC series of books, but, of course, the journal should have a much larger Editorial Board.

Conferences Committee

Claudio Zandron, Milan, Italy – Chair, zandron@disco.unimib.it
Henry Adorna, Quezon City, Philippines
Artiom Alhazov, Chişinău, Rep. of Moldova
Bogdan Aman, Iaşi, Romania
Matteo Cavaliere, Edinburgh, Scotland
Erzsébet Csuhaj-Varjú, Budapest, Hungary
Rudolf Freund, Wien, Austria
Marian Gheorghe, Bradford, U.K. – Honorary Member
Thomas Hinze, Cottbus, Germany
Florentin Ipate, Bucharest, Romania
Shankara N. Krishna, Bombay, India
Alberto Leporati, Milan, Italy
Taishin Y. Nishida, Toyama, Japan

Linqiang Pan, Wuhan, China – responsible of ACMC
Gheorghe Păun, Bucharest, Romania – Honorary Member
Mario J. Pérez-Jiménez, Sevilla, Spain
Agustín Riscos-Núñez, Sevilla, Spain
Petr Sosík, Opava, Czech Republic
K.G. Subramanian, Chennai, India
György Vaszil, Debrecen, Hungary
Sergey Verlan, Paris, France
Gexiang Zhang, Chengdu, China

Awards Committee:

Mario Pérez-Jiménez, Seville, Spain – Chair, marper@us.es
Marian Gheorghe, Bradford, U.K., m.gheorghe@bradford.ac.uk
Giancarlo Mauri, Milan, Italy, mauri@disco.unimib.it
Gheorghe Păun, Bucharest, Romania, gpaun@us.es
Linqiang Pan, Wuhan, China, lqpanhust@gmail.com

Rules of functioning:

1. Prizes to be awarded annually: (1) The PhD Thesis of the Year, (2) The Theoretical Result of the Year, (3) The Application of the Year.
2. Each prize consists of diplomas for each co-author, one copy of the Hamangia thinker[1] and one voucher for a discount in the registration fee for the first of BWMC, CMC or ACMC to take place after the prize was voted; the discount will be fixed by the organizing committee of the meeting; in case of several authors, they will choose the one of them to benefit of the voucher.
3. Any registered member of IMCS can be nominated and can receive any of the three prizes. In cases (2) and (3), the prizes are awarded to the authors of a paper or of an application, with at least one of authors being a member of IMCS. The members of the Awards Committee cannot receive any prize, neither they can be coauthors of papers or applications which receive one of prizes (2) and (3).
4. If the Awards Committee considers necessary, each year at most one of the prizes can be awarded ex aequo, to two winners.
5. Any registered member of IMCS can propose a candidate for any prize, by sending to any member of the Awards Committee the relevant information (and any additional information requested by the Awards Committee). Implicitly, the Awards Committee can itself make nominations.

[1] A Neolithic age clay sculpture, about 4000 years BC, found in Romania – see the image at the next page.

6. The nominations for the year Y should reach the Awards Chair before 20 of January of the year Y+1. The Awards Committee members decide the winners by the middle of February, and the prizes are awarded at the first edition of BWMC, CMC or ACMC where the winners participate in.
7. The members of the Awards Committee and the rules of functioning can be changed every year, after March 1, at the proposal of the Chair person or of any member of the IMCS Board, subject to a vote in the IMCS Board.

The IMCS prizes are mainly meant to reward the excellency in MC research, equally focusing on theory and applications, and to encourage young researchers.

The prizes are not subject to competitions, they do not identify *the best* PhD thesis or paper or application, they just point that a certain work/result is of a high value. This does not imply that other works/results are not so. We cannot rank scientific results like in sport, in a mathematical sense.

We only want to call attention to certain works – thus also calling attention to MC and to IMCS.

The prestige of a prize will be given by the prestige of the winners, also on their evolution in time, during their careers.

To reach these goals, we have to be conservative, exigent, transparent in our nominations and, especially, in selection.

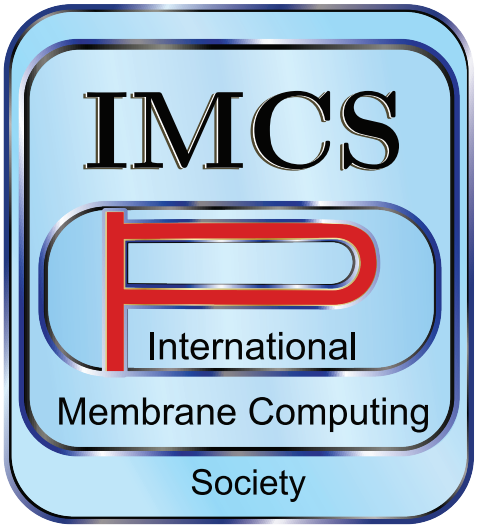
Nominations for the prizes for 2017 are waited for until January 20, 2018, and can be sent at any time, electronically, to any member of the Awards Committee (preferably with a CC to all members).

Constitution of the International Membrane Computing Society – IMCS

Article 1: Name

(1.1.) The name of the Society shall be International Membrane Computing Society, abbreviated IMCS.

(1.2.) The logo of IMCS is the one in the figure below. It should appear in all relevant places, such as IMCS web page, posters, calls, on the cover of the *Bulletin of IMCS*, etc.



Article 2: Objects

(2.1.) The society shall be a nonprofit academic organization, having as its goal to promote the development of membrane computing (MC), internationally, at all levels (theory, applications, software, implementations, connection with related areas, etc.).

(2.2.) A special attention will be paid to the communication/cooperation inside MC community, to connections with other professional scientific organizations with similar aims, and to promoting MC to young researchers.
(2.3.) IMCS will publish proceedings, journals or other materials, printed or electronically, as it sees fit.
(2.4.) IMCS will organize yearly MC meetings, such as the **Conference on Membrane Computing** (CMC), the **Asian Conference on Membrane Computing** (ACMC), the **Brainstorming Week on Membrane Computing** (BWMC), as well as further workshops/meetings, as it sees fit.

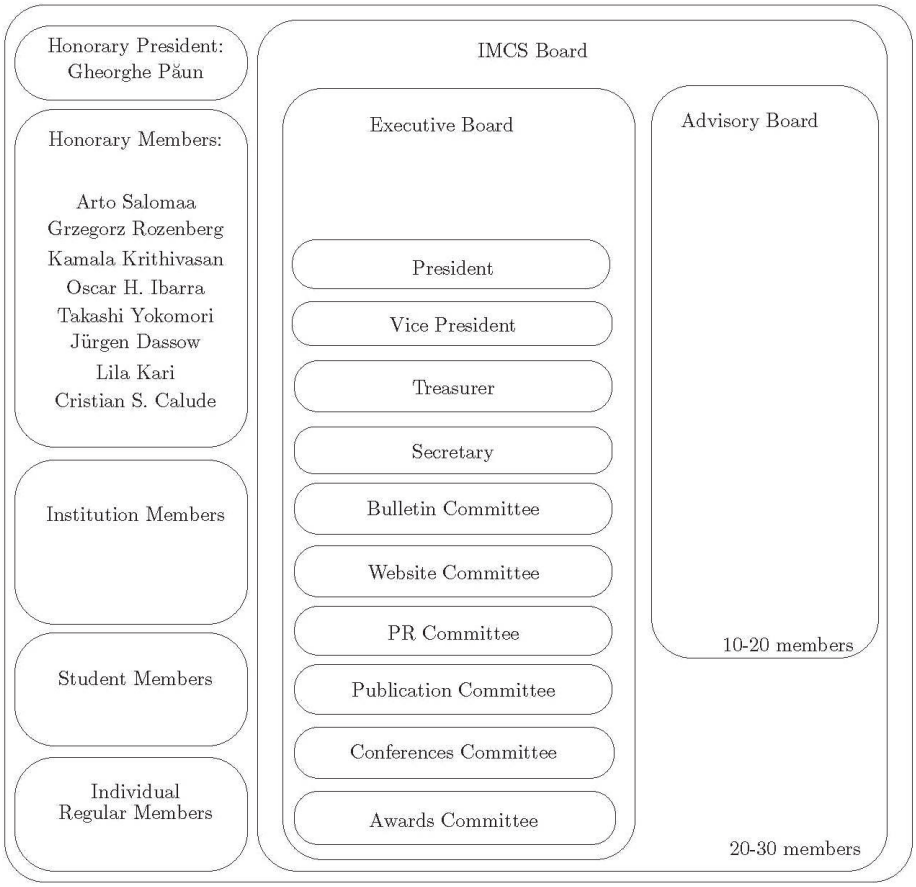
Article 3: Membership

(3.1.) There are four categories of members: **Honorary Members, Regular Members, Student Members**, and **Institution Members**.
(3.2.) The Honorary Members are elected by the IMCS Board (email voting, majority rule). Regular membership is open to all persons interested, on completing a membership form.
(3.3.) Student Members can be undergraduate, master, and PhD students, and they are eligible for various facilities IMCS is planning for students.
(3.4.) Institutions which want to join IMCS and support it can become Institution Members. Any support/sponsorship from an institution will be acknowledged in an appropriate way in IMCS publications.
(3.5.) Any member, of any kind, is supposed to know and accept the Constitution of IMCS.

Article 4: Structure

(4.1.) The structure of IMCS and its governance are as specified in the next figure. The figure also specifies the ten Honorary Members with whom the Society starts (February 2016).
(4.2.) Gheorghe Păun, the founder of MC, is appointed **Honorary President** of IMCS.
(4.3.) The work of IMCS is organized and conducted by the **Board of IMCS**, consisting of the **Executive Board** and the **Advisory Board**. The Advisory Board should have between 10 and 20 members, hence in total the IMCS Board should contain between 20 and 30 members,
(4.4.) The Executive Board consists of four individual positions: **President, Vice President, Treasurer**, and **Secretary**, and six **Committees: Bulletin, Website, PR, Publication, Conferences**, and **Awards Committee**. Each of these six Committees has a chair person. The Advisory Board also has a chair person.
(4.5.) The four individual positions from the Executive Board, the six chair persons of the Committees, the members of the Advisory Board, and the chair of the Advisory Board are elected by the IMCS Board (email voting, majority rule). Each chair of a Committee appoints a number of Committee members as he/she sees fit.

(4.6.) All the elected positions are elected for one year. After one year, a change of an elected person can be proposed by the President or the Vice President of the IMCS Board, by the person itself (resignation), or by two thirds of members of the IMCS Board, and it is voted in the IMCS Board (email voting, majority rule). If there is no change proposal, then the person who occupies any position in the IMCS Board continues in the same position, for one further year.



Article 5: Duties and competencies

(5.1.) The IMCS Board President represents the Society in relation with any external entity, organizes/coordinates the activity of the IMCS Board, initiates voting in the IMCS Board, chairs any panel/meeting of the Society.

(5.2.) The Vice President helps the President in all his/her activity, represents the President when he/she is not available (e.g., in chairing panels/meetings). Every year, the President and the Vice President present a common report

about IMCS activity, first circulating it by email in the IMCS Board and, after possible corrections, posting it in the IMCS web page.

(5.3.) The Treasurer takes care of the income and expenses of IMCS, and each year presents a report in this respect to IMCS Board. This report is analyzed and voted in the IMCS Board (email voting, majority rule).

(5.4.) The Secretary is responsible to keep a track record of the IMCS: memberships, reports, voting results, etc.

(5.5.) The Bulletin Committee takes care of editing the *Bulletin of IMCS*, first accumulating information/materials in an electronic format and then printed, if needed/requested, with a periodicity to be decided by the IMCS Board.

(5.6.) The Website Committee takes care of the IMCS web page, whose structure should be decided by the IMCS Board.

(5.7.) The PR Committee is responsible with developing relationships with other similar organizations and promoting IMCS on various scientific forums, advertising its activity on specialised networks and at international events.

(5.8.) The Publication Committee supervises the publication of proceedings, special issues of journals, collective volumes edited under the auspices of IMCS. Two particular goals of this Committee are to initiate a specialized journal, *International Membrane Computing Journal*, and a specialized series of monographs.

(5.9.) The Conference Committee works as a steering committee for the two MC yearly conferences, CMC and ACMC, looking for venues, suggesting (in cooperation with the organizing committees) program committees and invited speakers, possible sponsors and publications.

(5.10.) The Awards Committee collects nominations and decides the winners of three yearly IMCS Prizes: **(1) The PhD Thesis of the Year, (2) The Theoretical Result of the Year, (3) The Application of the Year**. The Awards Committee has its Rules of functioning, which are voted by the IMCS Board (email voting, majority rule).

Article 6: Voting

(6.1.) Each member of the IMCS Board (between 20 and 30 members) has one vote.

(6.2.) A voting, on any subject, can be initiated by the President, the Vice President, or by two thirds of the IMCS Board members.

(6.3.) The message proposing a vote should specify the issue to be decided in such a way that the alternatives YES and NO are clear. The message should be sent to all members of the IMCS Board, the voting messages of the members should also be sent to all members (full transparency). The voting should last 30 days. If a member is not replying in the first 15 days, the initiator of the vote should contact him/her once again. If a member is not replying even to the second message, then his/her vote is considered *abstaining*.

(6.4.) "Majority rule" means that at least half of the IMCS Board have voted (YES, NO, or abstaining) and the decision is made according to the number of

YES and NO votes which is higher. In case of a draw, the vote of the President is decisive – unless if the President does not decide to repeat the vote, maybe changing the object of the vote.

(6.5.) All ambiguities and uncovered cases should be clarified by discussions in the IMCS Board and, if decided so, proposed as amendments to the Constitution.

Article 7: Panels

(7.1.) On the occasions of IMCS annual meetings, like BWMC, CMC, and ACMC, panels should be organized, chaired by the President, the Vice President or, in their absence, by another member of the IMCS Board designated by the President, to discuss current issues of the Society.

Article 8: Finance

(8.1.) Income: possible membership fees, as decided by IMCS Board, donations, sponsorhips, conference registration fees, participation to research projects.

(8.2.) Expenses: IMCS prizes, students support, *Bulletin of IMCS* hardcopy, maintaining web pages, sponsoring MC conferences – all these and anything else, under the control of the IMCS Board.

Article 9: Amendments

(9.1.) Amendments to IMCS Constitution can be proposed by any member of the IMCS Board, at any time. Any amendment should be discussed and voted in the IMCS Board (email voting, majority rule) and then, if accepted, published in the IMCS web page, thus being available to all members of IMCS.

Article 10: Dissolution

(10.1.) The dissolution of IMCS should be done in two steps: first, a vote in the IMCS Board is organized (email voting, two thirds majority), and, if the dissolution proposal passes, a general vote is organized, where all regular members participate (email voting, two thirds majority; in order the vote to be valid, at least half of the members should vote).

(10.2.) If the Society decides to get dissolved, all remaining assets shall be donated to a similar organization, at the choice of the IMCS Board.

Article 11. Provisory statement

The present Constitution will get provisionally valid by being voted (by email, majority rule), in March 2016, in the IMCS Board, as this Board was constituted by consensus during BWMC 2016 and soon after that. Then, it will be published in the IMCS web page and, as soon as possible, in 2016, it will be voted by all individual members of IMCS (email voting, majority rule). The vote will last one month and to voting will participate all individual members of IMCS, students or regular, registered until the last day of the previous month.

IMCS Journal:
Journal of Membrane Computing
(JMC)

International Membrane Computing Society (IMCS) witnesses the processes of gestation and the birth of the new international journal, *Journal of Membrane Computing (JMC)*. The Editor-in-Chief of JMC, Professor Linqiang Pan, signed the publishing agreement with Springer Nature Singapore Pte Ltd. on March 20, 2017, in Wuhan, China.

JMC is an international journal with four issues per volume (per year). The Journal Homepage *http://www.springer.com/journal/41965* is setting up and will be publicized soon. The first accepted papers are planned to be online in 2018 and thereafter will be published in the Spring of 2019.

JMC aims to foster the dissemination of new discoveries and novel technologies in the area of membrane computing and the related areas like bio-inspired computing and natural computing. The focus of this journal is to provide a publication and communication platform for researchers, professionals and industrial practitioners, covering the theoretical fundamentals and technological advances to various applications. JMC solicits original, high-quality and previously unpublished research papers, survey and review articles, short communications, and tutorial papers.

Springer

springer.com

Journal of Membrane Computing

News from MC Research Groups

A Natural Computing Research Team in Mongolia

Tseren-Onolt Ishdorj and Erdenebaatar Altangerel

Department of Computer Sciennce
School of Information and Communication Technology
Mongolian University of Science and Technology
Ulaanbaatar 13342, Mongolia
tseren-onolt@must.edu.mn, erka@must.edu.mn

A team for natural computing research has been recently established in the Department of Computer Science, School of Information and Communication Technology of the Mongolian Science and Technology University, with generous support provided by Dr Gheorghe Păun, member of the Romanian Academy and Academia Europaea, who has donated books and scientific journals for this research team. The new bachelor and master degree curricula for computer science within the department include courses on natural computing related topics. Three postdoctoral researchers and two PhD students are currently in the research team. They have been conducting research related to membrane computing, gene assembly in ciliates, and biochip design, teaching also nature-inspired computing topics.

The newly established reaserch team has successfully secured research grants funded by the Mongolian Science and Technology Foundation allowing its team members to carry out investigations in natural computing.

A previous research grant, titled "Natural computing: A Study of Membrane Computing Possible Implementation in Parallel Architecture, and Gene-Permutation Application in Cryptanalysis", mainly concentrated on implementations for SNP systems, following the fundamental principles of the model described in [3], [4] , and on gene assembly applications in cryptanalysis, using the models developed in [2], [1].

A more recent grant is the Taiwanese-Mongolian joint research project titled "Surface-enhanced Raman scattering (SERS) detection of bacteria in microfluidic Dielectrophoresis (DEP)" 2017-2020. The teams involved in the project aim to design and develop a biochip for biomedical purposes.

Acknowledgments

The note is supported by Mongolian Science and Technology Foundation (grant MOST-MECSS-2017001, xxx).

References

1. T. Ishdorj, V. Rogojin, I. Petre, Computational Power of Intramolecular Gene Assembly, *International Journal of Foundations of Computer Science*, 18, 5 (2007), 1123 – 1136.
2. A. Karimi, H. S. Shahhoseini, Cryptanalysis of a Substitution-Permutation Network Using Gene Assembly in Ciliates. *Int. J. Communications, Network and Systems Sciences*, 5 (2012), 154–164.
3. Z. Xu, M. Cavaliere, P. An, S. Vrudhula, Y. Cao, The Stochastic Loss of Spikes in Spiking Neural P Systems: Design and Implementation of Reliable Arithmetic Circuits. *Fundamenta Informaticae*, 134, 1-2 (2014), 183–200.
4. T. Ishdorj, A. Leporati, L. Pan, X. Zeng, X. Zhang, Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources. *Theoretical Computer Science*, 411, 25 (2010), 23452358.

A Symbolic Membrane Computing Approach to the Control of Multi-Robot Systems

Andrei George Florea and Cătălin Buiu

Department of Automatic Control and Systems Engineering
Politehnica University of Bucharest
{andrei.florea,catalin.buiu}@acse.pub.ro

Multi-robot systems are groups of robots that share certain resources such as perception, actuation or communication channels. Such a system can be controlled using either a centralized or a distributed model. Centralized models employ a global communication protocol that allows one or more nodes to control a group of robots. The failure of a control node affects the entire list of listening robots.

One of the main advantages of distributed models is that of fault-tolerance because there is no central point of failure. Each robot has an individual control model that uses only local perception and actuation. Defining a distributed control application, at the individual level, can be a difficult task due to the need to model local perceptions, interactions and actions. Symbolic objects can simplify the modelling task by associating a certain entity (sensor measurement, effector command, message, ...) with a symbolic object.

The structure of the control model is also important to consider due to the inherent complexity of the interactions between internal robot components and also with other robots.

Membrane computing offers various modelling solutions that can be successfully applied to distributed robot control. The authors have proposed an initial version of a P colony-based distributed robot control model in [2]. The proposed controller was expanded and described in detail in [3].

The most recent advancement was presented in [1]:

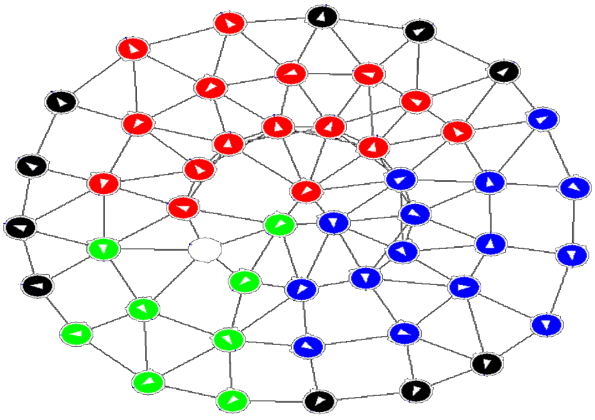
> A. G. Florea and C. Buiu, "A distributed approach to the control of multi-robot systems using XP colonies," *Integrated Computer-Aided Engineering*, vol. 25, no. 1, pp. 15–29, 2018.

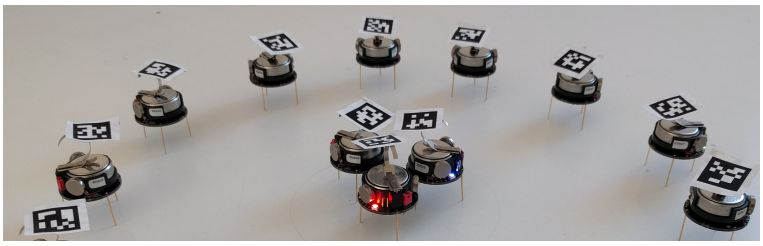
and focused on adding a symbolic robot interaction model that used the recently introduced concept of P swarms. The latter concept increased the transparency of the model because it eliminated the need for tedious background processing in order to ensure that messages from an emitter robot will be composed and sent. The proposed communication protocol does not add additional computational complexity because it only introduces two new variations of the commu-

nication rule: the IN and OUT exteroceptive rules, that use distinct membranes (buffers) to store the newly received or composed messages respectively, instead of the P colony environment. In the end, a local peer-to-peer symbolic communication protocol is now available to the general user.

Another contribution of the paper was a performance analysis of the proposed models. To our knowledge, this was the first time that P colony robotic models were compared with conventional models such as Finite State Machines (FSM). The results showed that although the P colony executor was slightly slower than that of the FSM, this did not proportionally translate into a loss of energy in the robotics context. In fact, noticeable differences were observed only in simulation. In real life, the inherent constructive differences between robots and their physical constraints have compensated for this issue, to the point where the proposed P colony model outperformed the FSM one.

The experiments where performed on 12 real Kilobot robots and up to 50 simulated ones. The main robotic task was that of color segregation, which requires local message exchange, movement and LED color changing.





The entire software stack, including here the P/XP colony executor (`Lulu`) as well as the P colony robot controller (`Lulu_controller`) are free and open-source

software, available on *Github*. Each software component is available in both a *Python* and a *C* version due to the fact that the experiments that were presented in the paper were tested on microcontroller-based robots. This means that the *C* implementation is therefore optimized for tight memory and processor constraints.

- P/XP colony simulator
  - `Lulu_pcol_sim Python` (`https://github.com/andrei91ro/lulu_pcol_sim`)
  - `Lulu_pcol_sim_C` (`https://github.com/andrei91ro/lulu_pcol_sim_c`)
- Distributed robot controller
  - `lulu_kilobot VREP` (`https://github.com/andrei91ro/lulu_kilobot_vrep`)
  - `Lulu_kilobot_C` (`https://github.com/andrei91ro/lulu_kilobot_c`)

References

1. Andrei George Florea and Catalin Buiu. A distributed approach to the control of multi-robot systems using XP colonies. *Integrated Computer-Aided Engineering*, 25(1):15–29, 2018.
2. Andrei George Florea and Catalin Buiu. Development of a software simulator for P colonies. Applications in robotics. *International Journal of Unconventional Computing*, 12(2-3):189–205, 2016.
3. Andrei George Florea and Catalin Buiu. *Membrane Computing for Distributed Control of Robotic Swarms: Emerging Research and Opportunities.* IGI Global, USA, 2017.

PeP, an Open-Source Enzymatic Numerical P Systems Simulator

Andrei George Florea and Cătălin Buiu

Department of Automatic Control and Systems Engineering
Politehnica University of Bucharest
{andrei.florea,catalin.buiu}@acse.pub.ro

PeP is a free and open-source Enzymatic Numerical P System simulator written in Python. The simulator was designed as a command-line tool and thus has no dependencies (other than Python) and can process a P system described using a single text file.

The input file is written in a human-readable format that was inspired from the mathematical notation used to describe a numerical P system. The application can be either executed individually (standalone) or included as a computational module into other applications. The former usage method is adequate for model design and testing while the latter can be used to control other devices using membrane computing models.

The simulator along with source code and documentation are available freely at `https://github.com/andrei91ro/pep`. The documentation contains the following relevant sections:

1. getting started
2. input file syntax reference
3. case studies
4. API reference

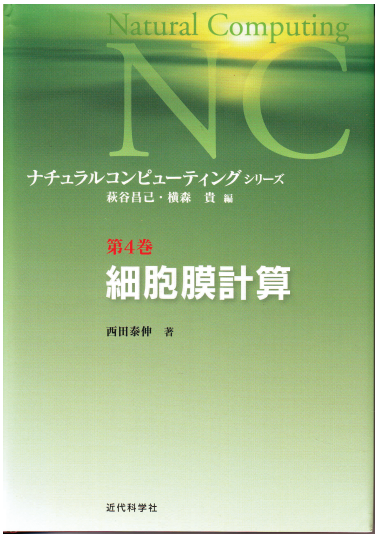
Book Presentation

Cell Membrane Computing

Taishin Y. Nishida

Department of Information Sciences
Toayama Prefectural University, Japan
nishida@pu-toyama.ac.jp and dd50ee70@gmail.com

Let me introduce my book *Cell Membrane Computing* published by Kindai-kagakusya, Tokyo, 2018, ISBN 978-4-7649-0568-9. This book is the first Japanese monograph on membrane computing.



The table of contents is the following:

1. Why membrane computing.
2. Preliminaries for theory of computing.
3. Variant of membrane computing — rewriting of symbol objects.
4. Variant of membrane computing — communication types.
5. Variant of membrane computing — active membrane and massive parallelism.

6. Tissue P systems and spiking neural P systems.
7. Application of P systems 1: A model of photosynthesis.
8. Application of P systems 2: Membrane algorithms.

In Chapter 1, I have shown, using small examples described below, that membranes and communications through membranes may bring new theory of computing. Chapter 2 contains pre-required notions and notations from theory of computing. Chapter 3 to 5 introduce some basic variants of P systems: definitions, classes, hierarchy in generating, accepting or transforming multisets from finite to computational universal, deterministic polynomial time systems solving NP or PSPACE complete problems. In Chapter 6, tissue and spiking neural P systems are described. The last two chapters mainly deal with applications of P systems from my own researches.

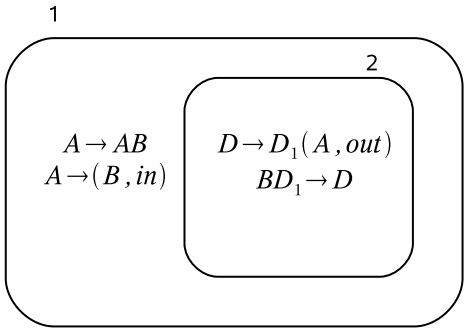


Fig. 1. A simple two membrane model.

Here I will describe the motivating examples in Chapter 1. Let us consider a nested membranes of degree 2 $[[\ ]_2]_1$ and sets of rules $\{A \to AB, A \to (B, in)\}$ in region 1 and $\{D \to D_1(A, out), BD_1 \to D\}$ in region 2 (Figure 1). Configuration changes from the initial configuration $[[D]_2]_1$ (one D in region 2) are illustrated in Figure 2. If there is one region with all rules

$$A \to AB, A \to B, D \to D_1A, BD_1 \to D,$$

then the next multisets are derived from one D:

$$\begin{array}{ccccccc} D \Rightarrow D_1A \Rightarrow D_1AB \Rightarrow & DAB & \Rightarrow D_1A^2B^2 \Rightarrow \cdots \\ \nwarrow \Downarrow & \Uparrow \Downarrow & \Downarrow \\ D_1B & DB & D_1AB^2 \Rightarrow & \cdots \end{array}$$

One can be aware of a remarkable difference between the above expression and Figure 2; the number of A increases boundlessly in the above expression while the number of A is at most 1 in Figure 2. If the systems model some physical or biological phenomena and A corresponds to an "expensive" object, e.g., A needs much resource to produce, the no

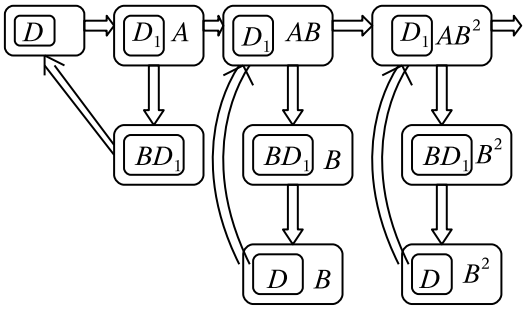


Fig. 2. Changes of configurations.

membrane model would not represent reality. That is, membrane separated regions and communications through membranes will be indispensable components of the system. These examples show that membranes may bring new theory of bio-inspired computing.

The book is written for readers with mathematical background of undergraduate level and interested in computer science especially in natural computing. In Japan, the research community of membrane computing is not large, more precisely, very small. It will be my pleasure if my book makes Japanese researchers to be familiar with membrane computing.

Tutorials, Surveys, Bibliographies

Spiking Neural P Systems: Theoretical Results and Applications

Haina Rong[1], Tingfang Wu[2], Linqiang Pan[2⋆], Gexiang Zhang[1]

[1] School of Electrical Engineering,
Southwest Jiaotong University, Chengdu 610031, China
{ronghaina,zhgxdylan}@126.com
[2] 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
{tfwu,lqpan}@hust.edu.cn

Summary. Spiking neural P systems, namely SN P systems, are a class of distributed and parallel neural-like computation models, inspired from the way neurons communicate by means of spikes. It has been shown that SN P systems have powerful computation capability and significant potential in real-life applications, and they have received more and more attraction from the scientific community. This paper firstly introduces the formal definition of standard SN P systems and some notions which are often used in this field. Then, the theoretical results about the computation power and efficiency of SN P systems are surveyed. The applications of SN P systems are recalled by summarizing the literature about the real-life applications of SN P systems. Finally, some hot topics and further research lines on SN P systems are provided.

Key words: Bio-inspired computing, Membrane computing, P system, Spiking neural P system, Neural-like computing system

1 Introduction

Natural computing is the computational version of the process of extracting ideas from nature to develop computation systems, or using natural materials (e.g.,

⋆ Corresponding author

molecules) to perform computation. Typical examples of natural computing include well-known neural computing inspired by the functioning of the brain, evolution computing inspired by Darwinian evolution of species, swarm intelligence inspired by the collective behavior of groups of organisms, and so on.

Membrane computing is a new branch of natural computing, which was initiated by Gh. Păun in 1998. The area of membrane computing aims to abstract computing ideas from the structure and the functioning of the living cell as well as from the cooperation of cells in tissues, organs, and other populations of cells [34], e.g., constructing computation models and designing algorithms for optimization. The abstracted parallel and distributed computation models in the area of membrane computing are called membrane systems, also called P systems. Since membrane computing have powerful capability in parallel processing and significant potential in various applications, this area has attracted more and more people's interest. The area of membrane computing was listed by Thompson Institute for Scientific Information (ISI) as an emerging research front in computer science in 2003.

According to the membrane structure, membrane systems fall into three main categories: (1) cell-like P systems which have a hierarchical arrangement of membranes as in a cell; (2) tissue-like P systems which have several one-membrane cells as in a tissue; (3) neural-like P systems which have several neurons as in a neural net. This paper will introduce a kind of neural-like P systems, called spiking neural P systems (SN P systems, for short), which emphasize on a specific type of cell, i.e. the neuron.

The human brain is a complex information processing system, where more than a trillion neurons work in a cooperation manner to perform various tasks. Therefore, the human brain structure and functioning, from neurons, astrocytes, and other components to complex networks, is a gold mine for inspiring efficient computing devices. Inspired by the neurophysiological behavior of neurons sending electrical impulses (spikes) along axons from presynaptic neurons to postsynaptic neurons in a distributed and parallel manner, Ionescu et al. proposed SN P systems in 2006 [16]. SN P systems have become an attractive and promising research direction in the community of membrane computing [56]. Since the introduction of SN P systems, much attention has been paid to various topics, such as the establishment of SN P systems variants from different biological inspirations, computation power, computation efficiency, and applications.

In the past twelve years, there are lots of theoretical results and applications published in journals or conference proceedings, however, a few papers ([31] is one of them in Chinese) focused on the survey of the main results of SN P systems in theory and applications. This paper is to briefly summarize the development and main results achieved in computation models, computing power and efficiency of SN P systems, and the applications in fault diagnosis of electric power systems, combinatorial and engineering optimization, signal classification, etc.

The organization of this paper is as follows. Section 2 reviews the standard SN P system and its various variants. Section 3 summarizes the theoretical results

of SN P systems reported in literature. Main applications of SN P systems are presented in Section 4. Future research lines are listed in Section 5.

2 Spiking neural P systems and variants

A standard spiking neural P system (shortly, SN P system), of degree $m \geq 1$, is a construct of the form

$$\Pi = (O, \sigma_1, \ldots, \sigma_m, syn, out),$$

where:

(1) $O = \{a\}$ is the singleton alphabet (a is called spike);
(2) $\sigma_1, \ldots, \sigma_m$ are neurons, of the form

$$\sigma_i = (n_i, R_i), 1 \leq i \leq m,$$

where:
a) $n_i \geq 0$ is the initial number of spikes contained in the neuron;
b) R_i is a finite set of rules of the following two forms:
i) $E/a^c \rightarrow a; d$, where E is a regular expression over $\{a\}$, $c \geq 1$, and $dge0$ is the delay time, that is, the interval between applying the rule and releasing the spike;
ii) $a^s \rightarrow \lambda$, for some $s \geq 1$, with the restriction that $a^s \notin L(E)$ for any rule $E/a^c \rightarrow a; d$ of type i) from R_i;
(3) $syn \subseteq \{1, 2, \ldots, m\} \times \{1, 2, \ldots, m\}$ are synapses between neurons, with $(i, i) \notin syn$ for $1 \leq i \leq m$;
(4) $in, out \in \{1, 2, \ldots, m\}$ indicate the input and output neurons, respectively.

The rules of type i) are called spiking rules (also called firing rules). A spiking rule $E/a^c \rightarrow a; d$ in neuron σ_i is enabled if the following conditions are satisfied: (1) the content of neuron σ_i is described by the regular expression E associated with the rule, e.g., if neuron σ_i contains k spikes, then $a^k \in L(E)$; (2) the number of spikes in neuron σ_i is not less than the number of spikes consumed by the rule, i.e., $k \geq c$. The application of this rule means that neuron σ_i consumes c spikes, and produces one spike after a delay of d steps. If $d = 0$, then the produced one spike emitted by neuron σ_i is replicated and sent immediately to all neurons σ_j such that $(i, j) \in syn$. If $d \geq 1$, assume that the rule is applied at step t, then at steps $t, t+1, \ldots, t+d-1$, the neuron is closed, so that it cannot receive new spikes from neurons which have synapses to the closed neuron (if any of these neurons tries to send a spike to the closed neuron, then the particular spike is lost). At step $t + d$, the neuron fires and becomes open again, so that it can receive spikes (which can be used at step $t + d + 1$, when the neuron can apply rules again)

The rules of type ii) are called forgetting rules. A forgetting rule $a^s \rightarrow \lambda$ in neuron σ_i is enabled only if the neuron contains exactly s spikes. The application of such a rule means that neuron σ_i removes all s spikes, but produces no spike.

If a rule $E/a^c \to a; d$ has $E = a^c$, then it can be written in the simplified form $a^c \to a; d$.

If a rule $E/a^c \to a; d$ has $d = 0$, then it can be written in the simplified form $E/a^c \to a$.

In standard SN P systems, only one spike can be produced by a spiking rule in one time, which is always called a standard spiking rule. With the goal of simplifying the results about universality of SN P systems, Chen et al. proposed extended SN P systems [6], where several spikes can simultaneously be produced by a spiking rule in one time, which is called an extended rule. Specifically, extended rules are of the form $E/a^c \to a^p; d$, where E is a regular expression over $\{a\}$, $c \geq 1$, and $p, d \geq 0$, with the restriction $c \geq p$. The meaning of such a rule is that if the content of the neuron is described by the regular expression E, then c spikes are consumed and p spikes are produced. Because p can be 0 or greater than 0, extended rules are a generalization of both standard spiking and forgetting rules.

It is assumed that a global clock exists in an SN P system, marking the time for the whole system (for all neurons of the system). In each time unit, if a neuron σ_i have enabled rules, then it must apply (at most) one of the enabled rules. In this way, a situation may appear: neuron σ_i contains two enabled rules $E_1/a^{c_1} \to a; d_1$ and $E_2/a^{c_2} \to a; d_2$, and these two rules have $E_1 \cap E_2 \neq \emptyset$. If so, one of them is nondeterministically chosen to be applied. Thus, the rules are used in a sequential manner at the level of each neuron, but neurons function in parallel with each other.

The state of an SN P system at a given time is described by the number of spikes present in each neuron and the number of steps to count down until it becomes open (this number is zero if the neuron is already open). That is, the configuration of system Π is of the form $\langle r_1/t_1, \ldots, r_m/t_m \rangle$ for $r_i \geq 0$ and $t_i \geq 0$, where r_i indicates that neuron σ_i contains r_i spikes, and it will become open after t_i steps, $i = 1, 2, \ldots, m$. With this notation, the initial configuration of system Π is $\langle n_1/0, \ldots, n_m/0 \rangle$. By using the rules as described above, one can get a sequence consecutive configurations. Each passage from a configuration C_1 to a successor configuration C_2 is called a transition and denoted by $C_1 \Longrightarrow C_2$. Any sequence of transitions starting from the initial configuration constitutes a computation. A computation halts if it reaches a configuration where all neurons are open and no rule is enabled. With any computation, halting or not, one associates a spike train, that is, a binary sequence with occurrences of digit 1 (resp., 0) indicating that the output neuron sends one spike (resp., no spike) out of the system.

When an SN P system works as a number generator, the result of a computation can be defined in several ways. With any spike train containing at least two spikes, the time interval between the first two being emitted is considered as the computation result [16]. This way of defining the result of a computation has been extended in [33]: generalizing to the first the time intervals between the first k spikes of a spike train, or the time intervals between all consecutive spikes, or only alternately the time intervals between all consecutive spikes, etc.

The way of defining the result of a computation in membrane computing can also be introduced in SN P systems. That is, one can also consider the result of a computation as the total number of spikes sent into the environment by the output neuron when the computation halts [3].

An SN P system can also work as a number acceptor [16]. In general, a number is introduced in the form of the time interval between two spikes entering the system. If the computation eventually halts, then this number is said to be accepted by the system.

Moreover, the result of a computation can be also defined as the spike train itself [4]. In this way, an SN P system is used as a binary string language generator defined on the binary alphabet $\{0, 1\}$.

3 Theoretical Results of Spiking Neural P Systems

These results mainly concern two aspects: computation power and computation efficiency. We start by briefly presenting some results of the first type.

For standard SN P systems, Ionescu et al. proved that these systems are Turing universal (also say computationally complete), that is, equivalent with Turing machines, when they are used as number generators or number acceptors [16]. Moreover, If a bound is given on the number of spikes present in any neuron along a computation, then a characterization of semilinear sets of numbers is obtained; and a characterization of finite sets can be obtained by standard SN P systems with one neuron.

From both mathematical and computer science points of view, it is always desirable to make the construction of SN P systems as simple as possible without the loss of computation power. Ibarra et al. first investigated that the influence of some ingredients of SN P systems on the computation power of these systems, such as the regular expressions used in spiking rules, the delay in spiking rules, and forgetting rules [15]. It was proved that in the case of removing the delay in spiking rules or forgetting rules, standard SN P systems are Turing universal. Afterwards, Pan et al. improved these results and proved that standard SN P systems are still Turing universal with the restrictions: (1) the delay in spiking rules and forgetting rules are not used, (2) each neuron contains at most two rules, and (3) the rules in the neurons using two rules have the same regular expression which controls their firing [24].

The resource (here, in terms of the number of neurons) needed for constructing universal computing devices of various types has been heavily in computer science, e.g., Songtag et al. constructed a universal recurrent neural network by using 886 neurons [40]. Gh. Păun et al. This issue was also considered in SN P systems. Gh. Păun et al. constructed a universal SN P system having 49 neurons in the case of using extended rules and having 84 neurons in the case of using standard rules for SN P systems used as a device of computing functions; as a number generator, a universal SN P system with standard rules having 76 neurons, and

one with extended rules having 50 neurons were obtained [32]. The comparison result shows that SN P systems have a "desired" computation power while using less resource. Following the research line of small universal computing devices, some bio-inspired features and mathematical strategies have been introduced into universal SN P systems for computing natural numbers and functions, in order to reduce the number of neurons. Some known results are shown in Table 1.

Table 1. The known results of Turing universal SN P systems with small numbers of neurons.

Bio-inspired features	Number of neurons for computing functions	Number of neurons for computing natural numbers	Type of rules
Weighted synapses [30]	38	36	Standard
Rules on synapses [43]	30	-	Extended
	39	-	Standard
Rules on synapses with high capacity neurons [46]	-	6	Extended
Request rules [42]	28	4	Extended
Colored spikes [45]	-	3	Extended

Mathematical strategies	Number of neurons for computing functions	Number of neurons for computing natural numbers	Type of rules
Combined modules [60]	41	41	Extended
	67	63	Standard
Cooperating rules	59 [21]	-	Standard
	-	8 [41]	Extended
High capacity neurons	-	10	Extended
"Super" neurons with an infinite number of rules [22]	-	4	Extended

Besides used as number generators/acceptors and function computing devices, SN P systems can also be used as language generators. In a standard SN P system, the output neuron sends at most one spike into the environment in one time, thus the time instances when one spike exits the output neuron are marked with the digit 1, and the time instances when no spike is emitted by the output neuron are marked with the digit 0. In this way, the system can generate the binary string languages defined on the binary alphabet $\{0, 1\}$. In this definition, it was proved that the generative capacity of standard SN P systems is rather restricted, even some finite languages cannot be generated by these systems. However, by using morphism and projection, standard SN P systems can generate recursively enumerable languages [4].

For extended SN P systems, because several spikes can exit at the same time, the time instances when the output neuron emits i spikes are marked with the symbol b_i. In this way, extended SN P systems can generate any language defined

on an arbitrary alphabet. It was proved that extended SN P systems can directly generate recursively enumerable languages [6].

In standard SN P systems, all neurons function in parallel at the level of the system (i.e., the system works in the synchronous manner), while at most one rule can be applied at the level of each neuron (i.e., the system works sequentially) [16]. Inspired by some biological phenomena and computer science theories, SN P systems working in different modes were investigated, such as non-synchronized (i.e., asynchronous) mode [3], that is, a neuron can apply or not apply its rules in any step; sequential mode: at each step of a computation, one (resp. all) of the neurons with the maximum/minimum number of spikes among the neurons that are active will fire [14, 19]; exhaustive mode: whenever a rule is enabled in a neuron, it is used as many times as possible for the number of spikes from that neuron [17, 29].

With the inspirations of different biological phenomena, various new types of SN P systems were proposed. For example, inspired by the functioning of inhibitory impulses among biological neurons, Pan et al. introduced anti-spikes and inhibitory synapses into SN P systems, and proved a series of normal forms of SN P systems with anti-spikes [23, 44]. Inspired by the fact that astrocytes play an important role in the functioning and interaction of neurons, and astrocytes have excitatory and inhibitory influence on synapses, Hoogeboom et al. proposed SN P systems with astrocytes [27]. It was proved that SN P systems with astrocytes using simple neurons (all neurons have the same rule, one per neuron, of a very simple form) can achieve Turing universality. With the goal of identifying an easy way to determine the applicability of rules, Wang et al. proposed SN P systems with weights, which are inspired by the fact that a biological neuron can fire only when its membrane potential reaches or exceeds its threshold potential [47]. It was proved that SN P systems with weights are universal. Motivated by the excitatory or inhibitory nature of Ranvier nodes in biological neurons, Chen et al. constructed axon P systems [7]. Afterwards, Zhang et al. proved that four nodes (respectively, nine nodes) are enough for axon P systems to achieve Turing universality as number generators (respectively, function computing devices) [59]. Based on such a neurobiological fact that in the chemical synapse transmitting, there are multiple ion channels in a synapse, Peng et al. proposed SN P systems with multiple channels, and proved that such systems are universal [37].

Moreover, with mathematical and computer science motivations, many new types of SN P systems were also proposed. Based on the self-organizing and self-adaptive feature of artificial neural networks, Cabarle et al. introduced structural plasticity into the framework of SN P systems [2]. SN P systems with structural plasticity were proved to be universal. Afterwards, Song et al. proved that SN P systems with structural plasticity without any synapse at the beginning of a computation can also achieve Turing universality [53]. Incorporating ideas from nonstatic (i.e. dynamic) graphs and networks, Cabarle et al. proposed SN P systems with scheduled synapses, where synapses in such systems are available only at a specific schedule or duration [1]. SN P systems with scheduled synapses were

proved to be universal. With mathematical motivation, Wu et al. consider a combination of basic features of cell-like P systems and of SN P systems, that is, consider cell-like P systems with only one kind of objects and spiking rules as those in SN P systems. called cell-like SN P systems [55]. It was proved that cell-like SN P systems with four membranes can achieve Turing universality. In order to simplify the integration-and-fire conditions of SN P systems, Wu et al. exploited polarizations $+, 0, -$ to control the application of rules instead of regular expressions [54]. It was proved that SN P systems with three kinds of polarizations are universal. Inspired by the way that components communicate with each other by a request-response pattern in parallel-cooperating grammar systems, Pan et al. proposed SN P systems with communication on request [26]. It was proved that SN P systems with communication on request are universal when two types of spikes are used.

Another topic on SN P systems is to study their computation efficiency, that is, to study whether SN P systems can solve computationally hard problems in a feasible time. Under the assumption that $P \neq NP$, Leporati ea al. proved that a deterministic SN P system of a polynomial size cannot solve an NP-complete problem in a polynomial time, i.e., Milano theorem for SN P systems [11]. Hence, some features need to be introduced into SN P systems in order to enhance the efficiency of such systems. Generally, there are three kinds of features introduced into SN P systems to solve computationally hard problems as follows.

(1) Pre-computed resources (here, in terms of exponential number of neurons): Chen et al. first exploited SN P systems with pre-computed resources to solve in a constant time the NP-complete problem SAT in a semi-uniform way [5]. Leporati et al. provided semi-uniform and uniform solutions to the numerical NP-complete problem Subset Sum by using SN P systems with exponential size pre-computed resources [20]. Afterwards, Ishdorj et al. shown that the two PSPACE-complete problems QSAT and Q3SAT can be solved in a polynomial time by SN P systems with pre-computed resources in a uniform way. All the systems constructed above work in a deterministic way [18].
(2) Nondeterminism: Leporati et al. provided the solutions to SAT problem and Subset Sum problem in a semi-uniform or uniform way by using nondeterministic SN P systems but without pre-computed resources [11]. In [10], standard SN P systems without the delay feature and having a uniform construction were obtained.
(3) Neuron division and neuron budding: Inspired by neural stem cell division, Pan et al. introduced the features of neuron division and neuron budding into the framework of SN P systems, which can generate exponential workspace in linear time. It was shown that SN P systems with neuron division and neuron budding solve computationally hard problems by means of a space-time tradeoff in a polynomial time, which is illustrated with an efficient solution to SAT problem [25]. Afterwards, Wang et al. exploited SN P systems only with neuron division to provide a uniform solution to SAT problem in a polynomial time [48].

4 Applications of Spiking Neural P Systems

This section presents an overview of SN P systems from the perspective of real-life applications with respect to engineering optimization, fault diagnosis of electric power systems, image processing, and signal identification.

The design of optimization algorithms can build a bridge between SN P systems and real-life applications. In [57], an extended SN P system (ESNPS) was proposed by introducing the probabilistic selection of evolution rules and multi-neurons output and a family of ESNPS, called optimization SN P system (OSNPS), were further designed through using a guider to adaptively adjust rule probabilities to approximately solve combinatorial optimization problems. In [50], OSNPS was used to solve the fault section estimation problem in an electric power system by formulating it into an optimization problem. Thus, OSNPS can search output fault sections in an automatical way when the status information of protective relays and circuit breakers coming from a supervisory control and data acquisition system is considered as the input. Several types of fault cases consisting of a single fault, multiple faults and multiple faults with incomplete and uncertain information in an electric power system can be accurately diagnosed in the simulation experiments.

The application for diagnosing the faults in an electric power system is an attractive research direction and is well investigated in the past years through introducing an algebraic fuzzy reasoning approach into SN P systems called fuzzy reasoning SN P systems (FRSNPS) [35, 51]. The most attractiveness of FRSNPS is that they can offer an intuitive illustration based on a strictly mathematical expression, a good fault-tolerant capacity due to its handling of incomplete and uncertain messages in a parallel manner, a good description for the relationships between protective devices and faults, and an understandable diagnosis model-building process [52]. Until now, FRSNPS have been successfully used to diagnose the faults existing in transformers [35], power transmission networks [52, 36], traction power supply systems of high-speed railways [49], metro traction power systems [13], fault classification of power transmission lines [39] and fault lines detection in a small current grounding system [38].

As many problems in the processing of digital images have features which make it suitable for techniques inspired by nature, in [9], a novel link between SN P systems and Digital Imagery was presented, by providing an implementation of the parallel Guo and Hall algorithm in SN P systems to solve the skeletonization problem.

The learning ability for SN P systems is the fundamentals to be used for classification. In [12], the first attempt was made by using the framework of SN P systems to implement Hebbian learning. In [8], SN P systems were used to identify nuclear export signals and the promising experimental result with accurate rate 74.18 % was obtained.

5 Concluding Remarks and Future Research Lines

By introducing some further biologically-inspired features into SN P systems, it is worth developing new computation models which are "more realistic" to get closer to the brain. Moreover, to bring enough further biologically-inspired features to SN P systems may possibly model and simulate processes taking place in the "real" brain.

Most of the aforementioned SN P systems were proved to be Turing universal. In the area of SN P systems, an challenge and interesting problem is to look for classes of SN P systems which are not equivalent with Turing machines, but also not computing only semilinear sets of numbers. From the point of view of theory, such classes of systems are rather significant, because of the possibility of finding classes of systems with decidable properties. Moreover, from the point of view of applications, these classes of systems are also attractive, because of the possibility of finding properties of the modeled processes by analytical, algorithmic means.

The computation efficiency of SN P systems deserves further efforts. It is interesting to introduce new ingredients in the area of SN P systems to generate an exponential workspace in polynomial time, by trading space for time in solving **NP**-complete problem and **PSPACE**-complete problem.

There are many ingredients of usual P systems which were not considered for SN P systems, e.g., promoters, inhibitors, membrane creation. Thus, with mathematical motivation, considering these ingredients in the framework of SN P systems might also make sense.

Future application work on SNP systems could be focused on the *killer real applications* [58]. There are many possibilities like fault diagnosis with FRSNPS, OSNPS for engineering optimization and classification with the SNP systems with learning ability. Several important topics on FRSNPS were listed in [52]. The extension of OSNPS to numerical optimization problems is an ongoing task. How to develop a learning network with SNP systems is also a challenging topic.

Acknowledgments.

This work was supported by National Natural Science Foundation of China (61320106005, 61672437, 61702428, and 61772214) and by Sichuan Science and Technology Program (2018GZ0085, 2018GZ0185, 2017GZ0159).

References

1. Cabarle, F. G. C., Adorna, H. N., Jiang, M., Zeng, X.: Spiking neural P systems with scheduled synapses. IEEE Transactions on Nanobioscience, doi: 10.1109/TNB.2017.2762580 (2017).
2. Cabarle, F. G. C., Adorna, H. N., Pérez-Jiménez, M. J, Song, T.: Spiking neural P systems with structural plasticity. Neural Computing and Applications, **26**(8), 1905–1917 (2015).

3. Cavaliere, M., Ibarra, O. H., Păun, G., Egecioglu, O., Ionescu, M., Woodworth, S.: Asynchronous spiking neural P systems. Theoretical Computer Science, **410**(24-25), 2352–2364 (2009).
4. Chen, H., Freund, R., Ionescu, M., Păun, G., Pérez-Jiménez, M. J.: On string languages generated by spiking neural P systems. Fundamenta Informaticae, **75**(1-4), 141–162 (2007).
5. Chen, H., Ionescu, M., Ishdorj, T. O. On the efficiency of spiking nNeural P systems. In: Proceedings of the 8th International Conference on Electronics, Information, and Communication. Ulan Bator, Mongolia, 2006, 49–52.
6. Chen, H., Ionescu, M., Ishdorj, T. O., Păun, A., Păun, G., Pérez-Jiménez, M.J.: Spiking neural P systems with extended rules: universality and languages. Natural Computing, **7**(2), 147–166 (2008).
7. Chen, H., Ishdorj, T. O., Păun, G.: Computing along the axon. Progress in Natural Science, **17**(4), 417–423 (2007).
8. Chen, Z., Zhang, P., Wang, X., Shi, X., Wu, T., Zheng, P.: A computational approach for nuclear export signals identification using spiking neural P systems. Neural Computing and Applications, **29**(3), 695–705 (2018).
9. Díaz-Pernil, D., Peña-Cantillana, F., Gutiérrez-Naranjo, M. A.: A parallel algorithm for skeletonizing images by using spiking neural P systems. Neurocomputing, **115**, 81–91 (2013).
10. Leporati, A., Mauri, G., Zandron, C., Păun, G., Pérez-Jiménez, M.J.: Uniform solutions to SAT and Subset Sum by spiking neural P systems. Natural computing, **8**(4), 681–702 (2009).
11. Leporati, A., Zandron, C., Ferretti, C., Mauri, G.: On the computational power of spiking neural P systems. International Journal of Unconventional Computing, **5**, 459–473 (2009).
12. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: A spiking neural P system based model for Hebbian learning. In: Frisco, P., Corne, D.W., Păun, Gh. (eds.) Proc. WMC9, Edinburgh, UK, 2008, 189–207.
13. He, Y., Wang, T., Huang, K., Zhang, G., Pérez-Jiménez, P.J.: Fault diagnosis of metro traction power systems using modified fuzzy reasoning spiking neural P systems. Romanian Journal of Information Science and Technology, **18**(3), 256–272 (2015).
14. Ibarra, O. H., Păun, A., Rodríguez-Patón, A.: Sequential SNP systems based on min/max spike number. Theoretical Computer Science, **410**(30-32), 2982–2991 (2009).
15. Ibarra, O. H., Păun, G., Păun, G., Rodríguez-Patón, A., Sosík, P., Woodworth, S.: Normal forms for spiking neural P systems. Theoretical Computer Science, **372**(2-3), 196–217 (2007).
16. Ionescu, M., Păun, G., Yokomori, T.: Spiking neural P systems. Fundamenta Informaticae, **71**(2), 279–308 (2006).
17. Ionescu, M., Păun, G., Yokomori, T.: Spiking neural P Seystems with an exhaustive use of rules. International Journal of Unconventional Computing, **3**(2), 135–153 (2007).
18. Ishdorj, T. O., Leporati, A., Pan, L., Zeng, X., Zhang, X.: Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources. Theoretical Computer Science, **411**(25), 2345–2358 (2010).
19. Jiang, K., Song, T., Pan, L.: Universality of sequential spiking neural P systems based on minimum spike number. Theoretical Computer Science, **499**, 88–97 (2013).

20. Leporati, A., Gutiérrez-Naranjoo, M. A.: Solving Subset Sum by spiking neural P systems with pre-computed resources. Fundamenta Informaticae, **87**(1), 61–77 (2008).
21. Metta, V. P., Raghuraman, S., Krithivasan, K.: Small universal spiking neural P systems with cooperating rules as function computing devices. In: Gheorghe, M., Rozenberg, G., Salomaa, A., Sosík, P., Zandron C. (eds.) Proc. CMC15, Prague, Czech Republic, 2014, 300–313.
22. Neary, T.: A boundary between universality and non-universality in extended spiking neural P systems. In: Dediu, AH., Fernau, H., Martín-Vide C. (eds.) Proc. LATA 2010, Trier, Germany, 2010, 475–487.
23. Pan, L., Păun, G.: Spiking neural P systems with anti-spikes. International Journal of Computers Communications & Control, **4**(3), 273–282 (2009).
24. Pan, L., Păun, G.: Spiking neural P systems: an improved normal form. Theoretical Computer Science, **411**(6), 906–918 (2010).
25. Pan, L.,Păun, G., Pérez-Jiménez, M.J.: Spiking neural P systems with neuron division and budding. Science China Information Sciences, **54**(8), 1596–1607 (2011).
26. Pan, L., Păun, G., Zhang, G., Neri, F.: Spiking neural P systems with communication on request. International Journal of Neural Systems, **27**(08), 1750042 (2017).
27. Pan, L., Wang, J., Hoogeboom, H. J.: Spiking neural P systems with astrocytes. Neural Computation, **24**(3), 805-825 (2012).
28. Pan, L., Zeng, X.: A note on small universal spiking neural P systems. In: Păun, G., Pérez-Jiménez, P.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A. (eds.), Proc. WMC10, Curtea de Argeş, Romania, 2009, 436–447.
29. Pan, L., Zeng, X.: Small universal spiking neural P systems working in exhaustive mode. IEEE Transactions on Nanobioscience, **10**(2), 99–105 (2011).
30. Pan, L., Zeng, X., Zhang, X., Jiang, Y.: Spiking neural P systems with weighted synapses. Neural Processing Letters, **35**(1), 13–27 (2012).
31. Pan, L., Zhang, X., Zeng, X., Wang, J.: Research advances and prospect of spiking neural P systems. Chinese Journal of Computers, **31**(12), 2090–2096 (2008). (in Chinese)
32. Păun, A., Păun, G.: Small universal spiking neural P systems. BioSystems, **90**(1), 48–60 (2007).
33. Păun, G., Pérez-Jiménez, P.J., Rozenberg, G.: Spike trains in spiking neural P systems. International Journal of Foundations of Computer Science, **17**(04), 975–1002 (2006).
34. Păun, G., Rozenberg, G., Salomaa, A., eds.: The Oxford Handbook of Membrane Computing, Oxford University Press, Inc., New York, NY, USA, 2010.
35. Peng, H., Wang, J., Pérez-Jiménez, M.J., Wang, H., Shao, J., Wang, T.: Fuzzy reasoning spiking neural P system for fault diagnosis. Information Sciences, **235**, 106–116 (2013).
36. Peng, H., Wang, J., Ming, J., Shi, P., Pérez-Jiménez, M.J., Yu, W., Tao, C.: Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems. IEEE Transactions on Smart Grid, doi:10.1109/TSG.2017.2670602 (2017).
37. Peng, H., Yang, J., Wang, J., Wang, T., Sun, Z., Song, X., Luo, X., Huang, X.: Spiking neural P systems with multiple channels. Neural Networks, **95**, 66–71 (2017).
38. Rong, H., Ge, M., Zhang, G., Zhu, M.: A novel approach for detecting fault lines in a small current grounding system using fuzzy reasoning spiking neural P systems. International Journal of Computers, Communications & Control, **13**(2), 458–473 (2018).

39. Rong, H., Zhu, M., Feng, Z., Zhang, G., Huang, K: A novel approach for fault classification of power transmission lines using singular value decomposition and fuzzy reasoning spiking neural P systems. Romanian Journal of Information Science and Technology, **20**(1), 18–31 (2017)
40. Siegelmann, H. T., Sontag, E. D.: On the computational power of neural nets. Journal of computer and system sciences, **50**(1), 132–150 (1995).
41. Song, T., Pan, L.: A small universal spiking neural P system with cooperating rules. Romanian Journal of Information Science and Technology, **17**(2), 177–189 (2014).
42. Song, T., Pan, L.: Spiking neural P systems with request rules. Neurocomputing, **193**, 193-200 (2016).
43. Song, T., Pan, L., Păun, G.: Spiking neural Psystems with rules on synapses. Theoretical Computer Science, **529**, 82–95 (2014).
44. Song, T., Pan, L., Wang, J., Venkat, I., Subramanian, K. G., Abdullah, R.: Normal forms of spiking neural P systems with anti-spikes. IEEE Transactions on Nanobioscience, **11**(4), 352-359 (2012).
45. Song, T., Rodríguez-Patón, A., Zheng, P., Zeng, X.: Spiking neural P systems with colored spikes. IEEE Transactions on Cognitive and Developmental Systems, doi: 10.1109/TCDS.2017.2785332 (2017).
46. Song, T., Xu, J., Pan, L.: On the universality and non-universality of spiking neural P systems with rules on synapses. IEEE Transactions on NanoBioscience, **14**(8), 960–966 (2015).
47. Wang, J., Hoogeboom, H. J., Pan, L.,Păun, G., Pérez-Jiménez, M.J.: Spiking neural P systems with weights. Neural Computation, **22**(10), 2615-2646 (2010).
48. Wang, J., Hoogeboom, H. J., Pan, L.: Spiking neural P systems with neuron division. In: Gheorghe, M., Hinze, T., Păun, G., Rozenberg, G., Salomaa. A. (eds.) Proc. CMC11, Jena, Germany, 2010, 361–376.
49. Wang, T., Zhang, G., Pérez-Jiménez, M.J., Cheng, J.: Weighted fuzzy reasoning spiking neural P systems: application to fault diagnosis in traction power supply systems of high-speed railways. Journal of Computational and Theoretical Nanoscience, **12**(7), 1103–1114 (2015)
50. Wang, T., Zeng, S., Zhang, G., Pérez-Jiménez, M.J., Wang, J.: Fault section estimation of power systems with optimization spiking neural P systems. Romanian Journal of Information Science and Technology, **18**(3), 240–255 (2015).
51. Wang, T., Zhang, G., Pérez-Jiménez, M.J.: Fuzzy membrane computing: theory and applications. International Journal of Computers, Communications & Control, **10**(6), 904-935 (2015).
52. Wang, T., Zhang, G., Zhao, J., He, Z., Wang, J., Pérez-Jiménez, P.J.: Fault diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems. IEEE Transactions on Power Systems, **30**(3), 1182–1194 (2015).
53. Wang, X., Song, T., Gong, F., Zheng, P.: On the computational power of spiking neural P systems with self-organization. Scientific reports, **6**, 27624 (2016).
54. Wu, T., Păun, A., Zhang, Z., Pan, L. Spiking neural P systems with polarizations. IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2017.2726119 (2017).
55. Wu, T., Zhang, Z., Păun, G., Pan, L.: Cell-like spiking neural P systems. Theoretical Computer Science, **623**, 180–189 (2016).
56. Zhang, G., Pérez-Jiménez, M.J., Gheorghe, M.: Real-life applications with membrane computing, Berlin, Springer, 2017.

57. Zhang, G., Rong, H., Neri, F., Pérez-Jiménez, M.J.: An optimization spiking neural P system for approximately solving combinatorial optimization problems. International Journal of Neural Systems, **24**(5) Article No. 1440006 (2014).
58. Zhang, G., Pérez-Jiménez, M.J., Wang, J., Peng, H., Cheng, J., Wang, X., Macías-Ramos, Luis F., Wang, T., Riscos-N'uñez, A., Valencia-Cabrera, L., Shi, P.: Real applications of membrane computing models, In Gheorghe, M., et al. (Eds.), Multidisciplinary Creativity: Homage to Gheorghe Paun on his 65th birthday, 2015, pp. 173–185.
59. Zhang, X., Pan, L., Păun, A.: On the universality of axon P systems. IEEE Transactions on Neural Networks and Learning Systems, **26**(11), 2816–2829 (2015).
60. Zhang, X., Zeng, X., Pan, L.: Smaller universal spiking neural P systems. Fundamenta Informaticae, **87**(1), 117–136 (2008).

A Survey of Tissue-Like P Systems

Bosheng Song[1], Yingxin Hu[1,2], Henry N. Adorna[3], Fei Xu[1⋆]

[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
Email: boshengsong@hust.edu.cn, yingxinhu@hust.edu.cn, fei_xu@hust.edu.cn

[2] College of Information Science and Technology
Shijiazhuang Tiedao University Shijiazhuang
Shijiazhuang 050043, Hebei, China

[3] Algorithms and Complexity Laboratory
Department of Computer Science
University of the Philippines Diliman
Diliman 1101 Quezon City, Philippines
Email: hnadorna@up.edu.ph

Summary. Membrane computing is a branch of natural computing, which abstracts from the architecture and the functioning of living cells. The computing models investigated in membrane computing are distributed and parallel computing devices, which are generically called P systems. Three main families of P systems have been considered until now: cell-like P systems, tissue-like P systems and neural-like P systems. In this work, we first present the definitions of tissue-like P systems and several variants of tissue-like P systems, then some results about Turing universal and computational efficiency are recalled. Finally, a computational complexity theory within the framework of tissue-like P systems is introduced, polynomial complexity classes associated with several variants of tissue-like P systems are defined and some relevant results are presented. Different borderlines between efficiency and non-efficiency on the basis of the length of communication rules are presented.

Keywords: Bio-inspired computing, Membrane computing, Tissue P system, Universality, Computational complexity

1 Introduction

The research area of membrane computing was motivated by the structure and functioning of living cells, more specifically, by the role of membranes in compartmentalization of living cells into "protected reactors". This direction of research

⋆ Corresponding author

was initiated by Gh. Păun [35], and initial models were based on a cell-like arrangement of membranes, delimiting compartments where multisets of objects evolve according to (reaction) rules. The next important step in development of membrane computing was to consider other membrane structure, in [25], tissue-like membrane systems were considered, where membrane structure corresponding to an arbitrary graph. The latest development of this research area is neural-like membrane systems [20], which are motivated by spiking neural networks. For general information about membrane computing, one may consult [37, 39], for up-to-date source of information and several software products for simulating P systems, one can go to the P systems web page `http://ppage.psystems.eu`.

After twenty years of development, a large number of bibliographies has been achieved, including theory, applications and implementation software. From a theoretical point of view, most variants of P systems are Turing complete even when using a small number of membranes, rules of simple forms, ways of controlling the use of rules. Moreover, most of P systems are efficient, that is, with enhanced parallelism, P systems can solve **NP**-complete problems, even **PSPACE**-complete problems in a feasible time by trading space for time, that is, P systems can generate an exponential workspace in a linear time, the way to obtain such an exponential workspace is membrane division [36], string replication [24], membrane separation [27], membrane creation [26] and other operations were used.

As a modeling framework, P systems have several attractive features for handling discrete biological processes: scalability and programmability, ability to handle discrete data, easy understandability, inherent compartmentalization, etc. Usually, applications use in a framework of cell-like P systems and tissue-like P systems, specifically, a P system is constructed according to a given process (for instance, evolution of ecosystems, economic processes, etc), then a program is designed to simulate such P system, computer simulations are performed, and tuning parameters of the P system until we obtain a correct description of the real process. One can refer to [8] for more details.

In this work, we focus on the aspect of theory of tissue P systems. We first present the definitions of tissue P systems and several variants of tissue P systems, then some results about Turing universal and computational efficiency of the variants of tissue P systems are recalled. Finally, a computational complexity theory within the framework of tissue P systems is introduced, polynomial complexity classes associated with different variants of tissue P systems are defined and some relevant results are presented. Several frontiers between efficiency and non-efficiency on the basis of the length of communication rules are presented.

2 Tissue-like P systems

2.1 Basic tissue-like P systems

Tissue P systems are inspired by the structure of tissues and the way of communicating substances between two cells or between a cell and the environment. The

membrane structure of a tissue P system is described by a graph, where cells are placed in nodes of a graph and the environment is viewed as a distinguished node, an arc between two nodes corresponds to a communication channel between two cells or a cell and the environment. If an arc between two nodes exists, then they can communicate by means of communication (symport/antiport) rules [33]. Symport rules move objects from one region to other region together in one direction, while antiport rules move objects between two region in opposite directions. We remark that in [25], rules are rewriting rules, and target indications are associated with each obtained object. The present work focuses on tissue P systems with communication rules evolving through applications of symport/antiport rules.

Formally, we recall the definition of tissue P systems.

Definition 1. *A tissue P system of degree $q \geq 1$ is a construct*

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

where:

(1) Γ is a finite set of objects;
(2) $\mathcal{E} \subseteq \Gamma$ is the set of objects initially placed in the environment of the system, all of them available in an arbitrary number of copies;
(3) $\mathcal{M}_i, 1 \leq i \leq q$, are finite multisets of objects initially placed in the q cells of the system;
(4) R is a finite set of rules of the following forms:
- *Symport rules: $(i, u/\lambda, j)$, where $0 \leq i \neq j \leq q$, $u \in \Gamma^+$;*
- *Antiport rules: $(i, u/v, j)$, where $0 \leq i \neq j \leq q$, $u, v \in \Gamma^+$;*

(5) $i_{out} \in \{0, 1, 2, \ldots, q\}$ is the output region.

When applying a symport rule $(i, u/\lambda, j)$, the multiset u of objects is sent from region i to region j; while when applying an antiport rule $(i, u/v, j)$, the multiset u of objects is sent from region i to region j, and simultaneously the multiset v of objects is sent from region j to region i.

2.2 Some variants of tissue-like P systems

Tissue P systems with symport/antiport rules have been first investigated in [34], and then the model has been extended in [15] by adding a notion of state to communication channels, state on channel can be modified when a communication rule is applied. Instead of using maximal parallelism, in [15], rules are used in a sequential manner for each channel and in a parallel manner for the system, that is, for each channel, at most one rule is used, but all communication channels evolve in parallel at the same time.

Besides maximal parallelism, other strategy of using rules was also considered in membrane computing. In [28], flat maximal parallelism was proposed, where in each membrane, a maximal set of applicable rules is chosen and each rule in the set is applied exactly once in each step. In [48], the flat maximal parallelism

of using rules of a tissue P system with channel states was investigated, where on each channel, a maximal set of applicable rules is chosen and each rule in the set is applied once in each step. Flat maximal parallelism of using rules was also considered in tissue P systems with promoters [32].

Inspired from the living cell, several ways of generating new cells have been considered: membrane division [36], membrane separation [27], membrane creation [24], which make it possible to generate an exponential workspace, and can solve computationally hard problems. In [38], cell division is introduced into tissue P systems, and the `SAT` problem is solved by tissue P systems with cell division (see [11, 12, 14] for more details). Tissue P systems with cell separation were proposed in [29], and a computational complexity theory in the framework of such P systems was investigated.

In biological systems the parallelism occurs at various levels and involves different components and chemical elements. If rules used involve objects from one or two regions, there have been proposed tissue P systems with conditional uniport [50] or evolution-communication [5, 4, 6]. If rules used involve objects from four regions, two acting as inputs and two as outputs, researchers proposed generalized communication P systems [9, 10, 21, 51].

Recently, tissue P systems with evolutional symport/antiport rules were proposed in [49], where objects between cells or between a cell and the environment are moved, and may be evolved during this process. Moreover, cell division is introduced into tissue P systems with evolutional symport/antiport rules, and a variant of P systems, called tissue P systems with evolutional symport/antiport rules and cell division was proposed [49]. However, in [31], cell separation is introduced into tissue P systems with evolutional symport/antiport rules, and a computational complexity theory in the framework of such P systems was investigated. There are several other variants of tissue P systems, for instance, introducing promoters or inhibitors [45], energy [1], protein [7, 47], etc.

2.3 Computation power of tissue-like P systems

Many variants of tissue P systems have been proposed, and most of the models are Turing universal [2, 3, 16, 22, 46]. Here we briefly present the computation power of tissue P systems with channel states and tissue P systems with evolutional symport/antiport rules. For the computation power of many variants of tissue P systems, one can refer to [39] for more details.

Tissue P systems with channel states were proposed in [15], where the systems with one cell, rules of any weight and any number of states can only compute Parikh sets of matrix languages, while the systems with two cells are Turing universal when using minimal antiport rules and arbitrarily many states or antiport rules of weight two and one state; Also the systems with arbitrarily many cells, minimal antiport rules and three states are Turing universal. In [48], the computation power of tissue P systems with channel states working in a flat maximally parallel way were investigated, where the systems with arbitrarily many cells, arbitrarily many

states and antiport rules of length two are able to compute Parikh sets of finite languages; the systems with one cell, arbitrarily many states and noncooperative symport rules can compute at least all Parikh sets of matrix languages; moreover, the systems are Turing universal with one cell, one state and symport rules of length three or with two cells, arbitrarily many states and symport rules of length one or with arbitrarily many cells, four states and symport rules of length one.

In tissue P systems, objects evolve by means of symport/antiport rules, and objects just change their place within the system. In [49], tissue P systems with evolutional symport/antiport rules were considered, objects may be evolved when they are moved between cells or between a cell and the environment. It is proved that tissue P systems with evolutional symport/antiport rules are Turing universal by using one cell and evolutional symport rules of length at most three or evolutional antiport rules of length at most four [49].

In next section, we will recall the computational complexity of tissue P systems with symport/antiport rules (resp., tissue P systems with evolutional symport/antiport rules) and cell division or cell separation.

3 Computational complexity of tissue P systems

3.1 Recognizer tissue P systems with cell division or cell separation

In order to solve decision problems, the notions from classical Computational Complexity Theory are considered in membrane computing. Here we define recognizer tissue P systems with cell division or cell separation [38].

Definition 2. *A recognizer tissue P system with cell division of degree $q \geq 1$ is a tuple*

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

where

- $(\Gamma, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{out})$ *is a tissue P system with cell division of degree* $q \geq 1$*;*
- Γ *contains two distinguished objects* `yes` *and* `no`*;*
- Σ *is an (input) alphabet contained in* Γ*;*
- $\mathcal{M}_1, \ldots, \mathcal{M}_q$*, are finite multisets over* $\Gamma \setminus \Sigma$*;*
- $i_{in} \in \{1, \ldots, q\}$ *is the input cell, and* $i_{out} = 0$*;*
- *all computations halt;*
- *if* $\mathcal{C}$ *is a computation of* Π*, then either object* `yes` *or object* `no` *(but not both) must have been released into the environment at the last step of the computation.*

Definition 3. *A recognizer tissue P system with cell separation of degree $q \geq 1$ is a tuple*

$$\Pi = (\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \Sigma, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{in}, i_{out}),$$

where

- *the tuple* $(\Gamma, \Gamma_0, \Gamma_1, \mathcal{E}, \mathcal{M}_1, \ldots, \mathcal{M}_q, R, i_{out})$ *is a tissue P system with cell separation of degree* $q \geq 1$*, where* Γ *strictly contains an (input) alphabet* Σ *and two distinguished objects* `yes`*,* `no`*, and* $\mathcal{M}_i$ $(1 \leq i \leq q)$ *are multisets over* $\Gamma \setminus \Sigma$*;*
- $i_{in} \in \{1, \ldots, q\}$ *is the input cell and* i_{out} *is the label of the environment;*
- *if* $\mathcal{C}$ *is a computation of* Π*, then either object* `yes` *or object* `no` *(but not both) must have been released into the environment at the last step of the computation.*

A computation $\mathcal{C}$ is said to be an accepting computation (resp., rejecting computation) of a recognizer tissue P system with cell division or with cell separation if object `yes` (resp., object `no`) appears in the environment when the computation reaches halting configuration, while neither object `yes` nor `no` appears in the environment when the computation cannot stop.

We denote by TC (resp., TDC, TSC) the class of recognizer tissue P systems (resp., with cell division or cell separation). For each natural number $k \geq 1$, we denote by $TC(k)$ (resp., $TDC(k)$, $TSC(k)$) the class of recognizer tissue P systems (resp., with cell division or cell separation) and communication rule of length at most k (the length of a symport/antiport rule is the total number of objects involved in the rule). If the alphabet of the environment is empty set, then we denote by $\widehat{TC}$, $\widehat{TDC}$, $\widehat{TSC}$, $\widehat{TC}(k)$, $\widehat{TDC}(k)$, $\widehat{TSC}(k)$, respectively.

We denote by TEC (resp., $TDEC$, $TSEC$) the class of recognizer tissue P systems with evolutional symport/antiport rules (resp., and cell division or cell separation). If the length of an evolutional communication rule is the total number of objects involved in the rule, for each natural number $k \geq 1$, we denote by $TEC(k)$ (resp., $TDEC(k)$, $TSEC(k)$) the class of recognizer tissue P systems (resp., with cell division or cell separation) and evolutional symport/antiport rule of length at most k. If the alphabet of the environment is empty set, then we denote by $\widehat{TEC}$, $\widehat{TDEC}$, $\widehat{TSEC}$, $\widehat{TEC}(k)$, $\widehat{TDEC}(k)$, $\widehat{TSEC}(k)$, respectively. If the length of an evolutional symport/antiport rule is defined as an ordered pair whose first component is the total number of objects involved in the left hand side of the rule and the second component is the total number of objects involved in the right hand side of the rule, then we denote by $TEC(k_1, k_2)$, $TDEC(k_1, k_2)$, $TSEC(k_1, k_2)$, $\widehat{TEC}(k_1, k_2)$, $\widehat{TDEC}(k_1, k_2)$, $\widehat{TSEC}(k_1, k_2)$, respectively.

Next, we present the definition of solving a problem in a uniform way by means of families of recognizer tissue P systems with cell division or cell separation [40].

Definition 4. *Let* $\mathcal{R}$ *be a class of recognizer tissue P systems with input cell. A decision problem* $X = (I_X, \theta_X)$ *is solvable in polynomial time by a family* $\mathbf{\Pi} = (\Pi(n))_{n \in \mathbb{N}}$ *of recognizer tissue P systems from* $\mathcal{R}$ *in a uniform way, and we denote this by* $X \in \mathbf{PMC}_{\mathcal{R}}$ *if the following conditions hold:*

- *The family* $\mathbf{\Pi}$ *is polynomially uniform by Turing machines.*
- *There exists a pair* (cod, s) *of polynomial-time computable functions over* I_X *such that: (a) for each instance* $u \in I_X$*,* $s(u)$ *is a natural number and* $cod(u)$ *is an input multiset of the system* $\Pi(s(u))$*; (b) for each* $n \in \mathbb{N}$*,* $s^{-1}(n)$ *is a*

finite set; and (c) the family $\mathbf{\Pi}$ *is polynomially bounded, sound and complete with regard to* (X, cod, s).

Let $\mathcal{R}$ be a class of recognizer tissue P systems (resp., with cell division or with cell separation). We denote by $\mathbf{PMC}_{\mathcal{R}}$ the set of all decision problems that can be solved in polynomial time by means of families of systems from $\mathcal{R}$. The class $\mathbf{PMC}_{\mathcal{R}}$ is closed under complement and polynomial-time reductions [42].

3.2 Efficiency of tissue P systems with cell division or cell separation

In this subsection, the efficiency of tissue P systems (resp., with cell division or cell separation) is presented.

(1) *Basic tissue P systems*

A basic tissue P system is defined as in Definition 1. In [13], it was shown that families of recognizer tissue P systems which solves a decision problem can be efficiently simulated by a family of recognizer transition P systems solving the same problem. Moreover, it is well known that families of recognizer transition P systems can only solve problems in class **P** in polynomial-time [19]. Hence we have the following result.

Theorem 1. $\mathbf{P} = \mathbf{PMC}_{TC}$.

(2) *Tissue P systems with cell division or cell separation*

By using the technique of dependency graph, it is shown that tissue P systems with cell division and communication rules of length at most 1 can only efficiently solve problems in class **P** [18].

Theorem 2. $\mathbf{P} = \mathbf{PMC}_{TDC(1)}$.

If we consider tissue P systems with cell division and communication rules of length at most 2, it is shown that the `HAM-CYCLE` problem (a well known **NP**-complete problem [17]) can be efficiently solved in polynomial-time by such systems [44]. Hence we have:

Theorem 3. $\texttt{HAM} - \texttt{CYCLE} \in \mathbf{PMC}_{TDC(2)}$.

Theorem 4. $\mathbf{NP} \cup \mathbf{co} - \mathbf{NP} \subseteq \mathbf{PMC}_{TDC(2)}$.

By using the simulation technique, it is proved that only problems in class **P** can be solved in polynomial-time by families of tissue P systems with cell separation when using communication rules of length at most 2 [30].

Theorem 5. $\mathbf{P} = \mathbf{PMC}_{TSC(2)}$.

If we consider tissue P systems with cell separation and communication rules of length at most 3, it is shown that the `SAT` problem can be efficiently solved in polynomial-time by such systems [43]. Hence we have:

Theorem 6. $\texttt{SAT} \in \mathbf{PMC}_{TSC(3)}$.

Theorem 7. $\mathbf{NP} \cup \mathbf{co}-\mathbf{NP} \subseteq \mathbf{PMC}_{TSC(3)}$.

(3) *Tissue P systems with cell division (or cell separation) and without environment*

Now we consider tissue P systems with cell division (or cell separation) and without environment.

By using the algorithmic technique, it is shown that families of tissue P systems with cell separation and without environment can only solve problems in class **P** [23]. Hence we have:

Theorem 8. $\mathbf{P} = \mathbf{PMC}_{\widehat{TSC}}$.

By using the simulation technique, it is shown that each family of recognizer tissue P systems with cell division when using communication rules of length at most $k \geq 1$ and solve a decision problem X in polynomial-time, can be efficiently simulated by a family of recognizer tissue P systems with cell division and without environment when using communication rules of length at most $k \geq 1$, solving X in polynomial-time [41].

Theorem 9. *For each* $k \geq 1$ *we have:* $\mathbf{PMC}_{\widehat{TDC}(k)} = \mathbf{PMC}_{TDC(k)}$.

3.3 Efficiency of tissue P systems with evolutional symport/antiport rules and cell division or cell separation

Evolutional symport/antiport rules were considered in tissue P systems in [49], and also cell division rules were introduced into such P systems. In [31], tissue P systems evolutional symport/antiport rules and cell separation were proposed, and computational complexity of such P systems was investigated.

Bearing in mind that any classical symport/antiport rule of the form $(i, u/v, j)$ can be considered as a particular case of the evolutional symport/antiport rule $[\ u\ ]_i[\ v\ ]_j \rightarrow [\ v\ ]_i[\ u\ ]_j$, hence the following results are obtained.

Theorem 10. *Let* $\mathbf{X} \in \{\mathbf{D}, \mathbf{S}\}$*, we have* $\mathbf{TXC} \subseteq \mathbf{TXEC}$.

Theorem 11. *Let* $\mathbf{X} \in \{\mathbf{D}, \mathbf{S}\}$ *and for each* $k \geq 1$*, we have*

$$\mathbf{TXC}(k) \subseteq \mathbf{TXEC}(k, k) \subseteq \mathbf{TXEC}(2k).$$

Theorem 12. *Let* $\mathbf{X} \in \{\mathbf{D}, \mathbf{S}\}$ *and for each* $k_1, k_2 \geq 1$*, we have*

$$\mathbf{TXEC}(k_1, k_2) \subseteq \mathbf{TXEC}(k_1 + k_2).$$

In [49], it is shown that only tractable problems can be efficiently solved by families of systems from $TDEC(2)$; and the $\texttt{SAT}$ problem can be solved by a family of systems from $TDEC(4)$.

Theorem 13. $\mathbf{PMC}_{TDEC(1)} = \mathbf{PMC}_{TDEC(2)} = \mathbf{P}$.

Theorem 14. $\mathtt{SAT} \in \mathbf{PMC}_{TDEC(4)}$.

In [31], it is shown that only tractable problems can be efficiently solved by families of systems from $TSEC(n, 1)$ or from $TSEC(1, n)$, for each natural number $n \geq 1$; moreover, the SAT problem can be solved in polynomial time by a family of systems from $TSEC(3, 2)$.

Theorem 15. *For each natural number $n \geq 1$, we have*

$$\mathbf{PMC}_{TSEC(n,1)} = \mathbf{PMC}_{TSEC(1,n)} = \mathbf{P}.$$

Theorem 16. $\mathtt{SAT} \in \mathbf{PMC}_{TSEC(3,2)}$.

By using the simulation technique as similar in [41], we can deduce the following results.

Theorem 17. *For each $k \geq 1$ we have:* $\mathbf{PMC}_{\widehat{TDEC}(k)} = \mathbf{PMC}_{TDEC(k)}$.

Theorem 18. *For each $k_1, k_2 \geq 1$ we have:* $\mathbf{PMC}_{\widehat{TDEC}(k_1,k_2)} = \mathbf{PMC}_{TDEC(k_1,k_2)}$.

4 Conclusions and remarks

In this work, a survey of tissue-like P systems and several variants of such P systems has been presented, some results about Turing universal and computational efficiency have been recalled. Moreover, a computational complexity theory within the framework of tissue P systems with symport/antiport rules (resp., with evolutional symport/antiport rules), with or without environment and using cell division or cell separation has been analyzed, polynomial complexity classes associated with several variants of tissue P systems are defined and some relevant results have been presented. Different borderlines between efficiency and non-efficiency on the basis of the length of communication rules have been presented.

In subsection 2.2, many variants of tissue P systems have been introduced, several of them are inspired by some basic features of biological membranes or other biological processes. An important research line is to consider new variant of tissue P systems and investigate the computational property of such P systems.

From subsection 3.2, we can know that many problems remain open about tissue P systems with evolutional symport/antiport rules and cell division or cell separation. For instance, whether **NP**-complete problems can be solved by a family of systems from $TDEC(3)$? What is the computational efficiency of a family of systems from $TSEC(2, 2)$?

Acknowledgements

The work was supported by National Natural Science Foundation of China (61602192, 61502186), China Postdoctoral Science Foundation (2016M600592, 2017T100554, 2016
M592335).

References

1. Alhazov, A., Freund, R., Leporati, A., Oswald, M., Zandron, C.: (Tissue) P Systems with Unit Rules and Energy Assigned to Membranes. Fund. Informa. 74(4), 391–408, 2006.
2. Alhazov, A., Freund, R., Oswald, M.: Tissue P Systems with Antiport Rules and Small Numbers of Symbols and Cells. Lect. Notes Comput. Sci. 3572, 100–111, 2005.
3. Ardelean, I., Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Peña-Cantillana, F., Reina-Molina, R., Sarchizian, I.: Counting Cells with Tissue-like P Systems. In: Proceedings of the Tenth Brainstorming Week on Membrane Computing, pp. 69–78. Fénix Editora, Sevilla, 2012.
4. Bernardini, F., Freund, R.: Tissue P Systems with Communication Modes. Lect. Notes Comput. Sci. 4361, 170–182, 2006.
5. Bernardini, F., Gheorghe, M.: Cell Communication in Tissue P Systems: Universality Results. Soft Comput. 9, 640–649, 2005.
6. Cavaliere, M.: Evolution-Communication P Systems. Lect. Notes Comput. Sci. 2597, 134–145, 2003.
7. Christina, A.H., Díaz-Pernil, D., Mathu, T.: A Uniform Family of Tissue P Systems with Protein on Cells Solving 3-Coloring in Linear Time. Nat. Comput. (2016). https://doi.org/10.1007/s11047-016-9590-1
8. Ciobanu, G.; Păun, Gh., Pérez-Jiménez, M.J.: Applications of Membrane Computing. Springer-Verlag, Heidelberg, 2006.
9. Csuhaj-Varjú, E., Vaszil, G., Verlan, S.: On Generalized Communicating P Systems with One Symbol. Lect. Notes Comput. Sci. 6501, 160–174, 2010.
10. Csuhaj-Varjú, E., Verlan, S.: On Generalized Communicating P Systems with Minimal Interaction Rules. Theor. Comput. Sci. 412, 124–135, 2011.
11. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Solving Subset Sum in Linear Time by Using Tissue P System with Cell Division. Lect. Notes Comput. Sci. 4527, 170–179, 2007.
12. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: A Uniform Family of Tissue P System with Cell Division Solving 3-Col in a Linear Time. Theor. Comput. Sci. 404(1), 76–87, 2008.
13. Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, Romero-Jiménez, Á.: Efficient simulation of tissue-like P systems by transition cell-like P systems. Nat. Comput. 8(4), 797–806, 2009.
14. Díaz-Pernil, D., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Jiménez, Á.: Computational Efficiency of Cellular Division in Tissue-like Membrane Systems. Rom. J. Inf. Sci. Tech. 11(3), 229–241, 2008.
15. Freund, R., Păun, Gh., Pérez-Jiménez, M.J.: Tissue P Systems with Channel States. Theor. Comput. Sci. 330(1), 101–116, 2005.
16. Freund, R., Verlan, S.: (Tissue) P Systems Working in The k-Restricted Minimally or Maximally Parallel Transition Mode. Nat. Comput. 10, 821–833, 2011.
17. Garey, M.R., Johnson, D.J.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman, San Francisco, 1979.
18. Gutiérrez-Escudero, R., Pérez-Jiménez, M.J., Rius-Font, M.: Characterizing Tractability by Tissue-Like P Systems. Lect. Notes Comput. Sci. 5957, 289–300, 2010.
19. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J., Romero-Jiménez, Á.: Characterizing tractability by cell-like membrane systems.

In K.G. Subramanian, K. Rangarajan, M. Mukund (eds.) Formal models, languages and applications, World Scientific, Singapore, pp. 137–154. 2006.
20. Ionescu, M., Păun, Gh., Yokomori, T.: Spiking Neural P Systems. Fund. Informa. 71(2-3), 279–308, 2006.
21. Krishna, S.N., Gheorghe, M., Ipate, F., Csuhaj-Varjú, E., Ceterchi, R.: Further Results on Generalised Communicating P Systems. Theor. Comput. Sci. 701 146–160, 2017.
22. Krishna, S.N., Lakshmanan, K., Rama, R.: Tissue P Systems with Contextual and Rewriting Rules. Lect. Notes Comput. Sci. 2597, 339–351, 2003.
23. Macías, L.F., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius, M., Valencia-Cabrera, L.: The Efficiency of Tissue P Systems with Cell Separation Relies on the Environment. Lect. Notes Comput. Sci. 7762, 243–256, 2013.
24. Madhu, M., Prakash, V.J., Krithivasan, K.: Rewriting Tissue P Systems. J. Univ. Computer Sci., 10(9), 1250–1271, 2004.
25. Martín-Vide, C., Pazos, J., Păun, Gh., Rodriguez-Paton, A.: Tissue P Systems. Theor. Comput. Sci. 296(2), 295–326, 2003.
26. Mutyam, M., Krithivasan, K.: P Systems with Membrane Creation: Universality and Efficiency. Lect. Notes Comput. Sci. 2055, 276–287, 2001.
27. Pan, L., Ishdorj, T.-O.: P Systems with Active Membranes and Separation Rules. J. Univ. Computer Sci., 10, 630–649, 2004.
28. Pan, L., Păun, Gh., Song, B.: Flat Maximal Parallelism in P Systems with Promoters. Theor. Comput. Sci. 623, 83–91, 2016.
29. Pan, L., Pérez-Jiménez, M.J.: Computational Complexity of Tissue-like P Systems. J. Complexity 26(3), 296–315, 2010.
30. Pan, L., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius, M.: New Frontiers of the Efficiency in Tissue P Systems. In Pan, L., Păun, Gh., Song, T. (eds.) Pre-proceedings of Asian Conference on Membrane Computing (ACMC 2012), Huazhong University of Science and Technology, Wuhan, China, October 15-18, pp. 61-73, 2012.
31. Pan, L., Song, B., Valencia-Cabrera, L., Pérez-Jiménez, M.J.: The Computational Complexity of Tissue P Systems with Evolutional Symport/Antiport Rules, Complexity, Article ID 3745210, 21 pages, 2018.
32. Pan, L., Wang, Y., Jiang, S., Song, B.: Flat maximal parallelism in tissue P systems with promoters. Rom. J. Inf. Sci. Tech. 20(1), 42–56, 2017.
33. Păun, A., Păun, Gh.: The power of Communication: P Systems with Symport/Antiport. New Generat. Comput. 20(3), 295–305, 2002.
34. Păun, A., Păun, Gh., Rozenberg, G.: Computing by Communication in Networks of Membranes. Int. J. Found. Comput. S. 13, 779–798, 2002.
35. Păun, Gh.: Computing with Membranes. J. Comput. Syst. Sci. 61(1), 108–143, 2000.
36. Păun, Gh.: P Systems with Active Membranes: Attacking NP-Complete Problems. J. Auto. Lang. Comb. 6(1), 75–90, 2001.
37. Păun, Gh.: Membrane Computing: An Introduction. Springer-Verlag, Berlin, 2002.
38. Păun, Gh., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Tissue P Systems with Cell Division. Int. J. Comput. Commun. 3(3), 295–303, 2008.
39. Păun, Gh., Rozenberg, G., Salomaa, A.(Eds.): The Oxford Handbook of Membrane Computing. Oxford University Press, New York, 2010.
40. Pérez-Jiménez, M.J., An Approach to Computational Complexity in Membrane Computing. Lect. Notes Comput. Sci. 3365, 85–109, 2005.

41. Pérez-Jiménez, M.J., Riscos-Núñez, A., Rius-Font, M., Romero-Campero, F.J.: The Role of the Environment in Tissue P Systems with Cell Division. In García-Quismondo, M., Macías-Ramos, L.F., Păun, Gh., Valencia-Cabrera, L. (eds.) Proceedings of the Tenth Brainstorming Week on Membrane Computing, Volume II, Seville, Spain, January 30-February 3, 2012, Report RGNC 01/2012, Fénix Editora, pp. 89-104, 2012.
42. Pérez-Jiménez, M.J., Romero-Jiménez, A., Sancho-Caparrini, F.: Complexity Classes in Models of Cellular Computing with Membranes. Nat. Comput. 2(3), 265–285, 2003.
43. Pérez-Jiménez, M.J., Sosík, P.: An Optimal Frontier of the Efficiency of Tissue P Systems with Cell Separation. Fund. Informa. 138(1-2), 45–60, 2015.
44. Porreca, A.E., Murphy, N., Pérez-Jiménez, M.J., An Optimal Frontier of the Efficiency of Tissue P Systems with Cell Division. In Garcí-Quismondo, M., Macías-Ramos, L.F., Păun, Gh., Pérez-Hurtado, I., Valencia-Cabrera, L. (eds.) Proceedings of the Tenth Brainstorming Week on Membrane Computing, Volume II, Seville, Spain, January 30-February 3, 2012, Report RGNC 01/2012, Fénix Editora, pp. 141–166, 2012.
45. Song, B., Pan, L.: Computational Efficiency and Universality of Timed P Systems with Active Membranes. Theor. Comput. Sci. 567, 74–86, 2015.
46. Song, B., Pan, L.: The Computational Power of Tissue-like P Systems with Promoters. Theor. Comput. Sci. 641, 43–52, 2016.
47. Song, B., Pan, L., Pérez-Jiménez, M.J.: Tissue P Systems with Protein on Cells. Fundam. Inform. 144, 77–107, 2016.
48. Song, B., Pérez-Jiménez, M.J., Păun, Gh., Pan, L.: Tissue P systems with channel states working in the flat maximally parallel way. IEEE Trans. NanoBiosci. 15(7), 645–656, 2016.
49. Song, B., Zhang, C., Pan, L.: Tissue-Like P Systems with Evolutional Symport/Antiport Rules. Inf. Sci. 378, 177–193, 2017.
50. Verlan, S., Bernardini, F., Gheorghe, M., Margenstern, M.: Computational Completeness of Tissue P Systems with Conditional Uniport. Lect. Notes Comput. Sci. 4361, 521–535, 2006.
51. Verlan, S., Bernardini, F., Gheorghe, M., Margenstern, M.: Generalized Communicating P Systems. Theor. Comput. Sci. 404, 170–184, 2008.

A Bibliography of Fuzzy Membrane Computing

Hong Peng, Jun Wang and Tao Wang

Research Group on Membrane Computing and Applications
Xihua University
Chengdu, 610039, China
ph.xhu@hotmail.com, wj.xhu@hotmail.com

Below is a bibliography of fuzzy membrane computing (FMC) [27] appearing in the publications, aiming to highlight the combinations of membrane computing and fuzzy logic. FMC is an active and significant research topic for the use of membrane computing to solve real-life applications. The bibliography maybe useful and beneficial to the researchers in the community of membrane computing and related areas, especially to the researchers and students who are working on the applications of membrane computing. The FMC models and their applications are summarized in Table 1.

References

1. J. Casasnovas, J. Miro, M. Moyá, F. Rosselló: A Fuzzy Approach to Membrane Computing with Approximate Copies. *Proc. BWUMC*, 121–127, 2004.
2. J. Casasnovas, F. Rosselló: Fuzzy P Systems and Their Applications in Computational Biology *Proc. EUSFLAT*, 1112-1117, 2005.
3. K. Chen, J. Wang, M. Li, J. Ming, H. Peng: Cell-like fuzzy p system and its application of coordination control in micro-grid. *Bio-Inspired Computing – Theories and Applications*, Volume 562 of the series Communications in Computer and Information Science, vol. 562, 18–32, 2015.
4. K. Huang, T. Wang, Y. He, G. Zhang, M.J. Pérez-Jiménez: Temporal fuzzy reasoning spiking neural P systems with real numbers for power system fault diagnosis. *Journal of Computational and Theoretical Nanoscience*, 13(6), 3804–3814, 2016.
5. K. Huang, G Zhang, X. Wei, H. Rong, Y. He, T. Wang: Fault classification of power transmission lines using fuzzy reasoning spiking neural P systems. *Bio-inspired Computing - Theories and Applications*, 109–117, 2017.
6. R.K. Idowu, R. Chandren, Z.A. Othman: Advocating the use of fuzzy reasoning spiking neural P system in intrusion detection. *2014 Asian Conference on Membrane Computing (ACMC)*, 1-5, 2014.
7. R.K. Idowu, C.M. Ravie, Z.A. Othman: Denial of service attack detection using trapezoidal fuzzy reasoning spiking neural P system. *Journal of Theoretical & Applied Information Technology*, 75(3), 397-404, 2015.

Table 1. The reference list of fuzzy models of membrane computing.

Model	Application	Reference list
Fuzzy reasoning spiking neural P systems	Power systems	[4],[5],[8],[12],[13], [22],[23] [24],[30],[31],[32],[33]
	Power supply systems in rail traffic	[28]
	Intrusion detection	[6]
Weighted fuzzy spiking neural P systems	Power supply systems in rail traffic	[20], [29]
Adaptive fuzzy spiking neural P systems	Power systems	[18], [21], [17]
Trapezoidal fuzzy spiking neural P systems	Power systems	[26],[27]
	Intrusion detection	[7],[14]
Triangular fuzzy spiking neural P systems	Power systems	[16]
Intuitionistic fuzzy spiking neural P systems	Power systems	[10]
Fuzzy tissue-like P syetems	Power systems	[11]
Fuzzy cell-like P syetems	Micro-grid control	[3],[19]

8. L.F. Macías-Ramos, M.A. Martínez-Del-Amor, M.J. Pérez-Jiménez: Simulating FRSN P systems with real numbers in P-Lingua on sequential and CUDA platforms. *Lecture Notes in Computer Science*, vol. 9504, 227–241, 2015.
9. A. Obtulowicz: General Multi-fuzzy Sets and Fuzzy Membrane Systems. *Lect. Notes Comput. Sci.*, 3365, 359-372, 2005.
10. H. Peng, J. Wang, J. Ming, P. Shi, M.J. Pérez-Jiménez, W. Yu, C. Tao: Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems. *IEEE Transaction on Smart Grid*, 2017. Available at http://dx.doi.org/10.1109/TSG.2017.2670602.
11. H. Peng, J. Wang, P. Shi, M.J. Pérez-Jiménez, A. Riscos-Núñez: Fault diagnosis of power systems using fuzzy tissue-like P systems. *Integrated Computer-Aided Engineering*, 24(4), 401–411, 2017.
12. H. Peng, J. Wang, M.J. Pérez-Jiménez, H. Wang, J. Shao, T. Wang: Fuzzy reasoning spiking neural P system for fault diagnosis. *Information Sciences*, 235, 2013: 106–116.
13. M.J. Pérez-Jiménez, C. Graciani, D. Orellana-Martín, A. Riscos-Núñez, et al.: Fuzzy reasoning spiking neural P systems revisited: a formalization. *Theoretical Computer Science*, 2017. Available at http://dx.doi.org/10.1016/j.tcs.2017.04.014.

14. K.I. Rufai: Attack detection in a rule-based system using fuzzy spiking neural P system. *International Journal of Informatics and Communication Technology*, 5(1), 11–20, 2016.
15. A. Syropoulos: Fuzzifying P Systems. *Comput. J.*, 49(5), 619-628, 2006.
16. C. Tao, W. Yu, J. Wang, H. Peng, K. Chen, J. Ming: Fault diagnosis of power systems based on triangular fuzzy spiking neural P systems. *Bio-Inspired Computing - Theories and Applications*, 385-398, 2016.
17. M. Tu, J. Wang, H. Peng, P. Shi: Application of adaptive fuzzy spiking neural P systems in fault diagnosis model of power systems. *Chinese Journal of Electronics*, 23(1), 87–92, 2014.
18. J. Wang, H. Peng, M. Tu, M.J. Pérez-Jiménez, P. Shi: A fault diagnosis method of power systems based on an improved adaptive fuzzy spiking neural P systems and PSO algorithms. *Chinese Journal of Electronics*, 25(CJE-2), 320–327, 2016.
19. J. Wang, K. Chen, M. Li, J. Ming, H. Peng: Cell-like fuzzy P system and its application in energy management of micro-grid. *Journal of Computational and Theoretical Nanoscience*, 13(6), 3643–3651(9), 2016.
20. J. Wang, P. Shi, H. Peng, M.J. Pérez-Jiménez, T. Wang: Weighted fuzzy spiking neural p systems. *IEEE Transactions on Fuzzy Systems*, 21(2), 209–220, 2013.
21. J. Wang, H. Peng: Adaptive fuzzy spiking neural P systems for fuzzy inference and learning. *International Journal of Computer Mathematics*, 90(4), 857–868, 2013.
22. J. Wang, L. Zhou, H. Peng, G. Zhang: An extended spiking neural P system for fuzzy knowledge representation. *International Journal of Innovative Computing, Information and Control*, 7(7A), 3709–3724, 2011.
23. J. Wang, H. Peng: Fuzzy knowledge representation based on an improving spiking neural p system. *2010 Sixth International Conference of Natural Computing*, ICNC2010, 6, 3012–3015, 2010.
24. T. Wang, J. Wang, H. Peng, Y. Deng: Knowledge representation using fuzzy spiking neural P system. *2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications*, Volume-1, 586–590, 2010.
25. T. Wang, J. Wang, J. Peng, H. Wang: Knowledge Representation and Reasoning Based on FRSN P System. *WCICA.*, 849-854, 2011.
26. T. Wang, G. Zhang, J. Zhao, Z. He, J. Wang, M.J. Pérez-Jiménez: Diagnosis of electric power systems based on fuzzy reasoning spiking neural P systems. *IEEE Transactions on Power Systems*, 30(3), 1182–1194, 2015.
27. T. Wang, G. Zhang, M.J. Pérez-Jiménez: Fuzzy membrane computing theory and applications. *International Journal of Computers, Communications & Control*, 10(6), 904–935, 2015.
28. T. Wang, G. Zhang, M.J. Pérez-Jiménez: Fault diagnosis models for electric locomotive systems based on fuzzy reasoning spiking neural P systems. *Lecture Notes in Computer Science*, vol. 8961, 385–395, 2014.
29. T. Wang, G. Zhang, M.J. Pérez-Jiménez, J. Cheng: Weighted fuzzy reasoning spiking neural P systems: application to fault diagnosis in traction power supply systems of high-speed railways. *Journal of Computational and Theoretical Nanoscience*, 12(7), 1103–1114, 2015.
30. T. Wang, G Zhang, H. Rong, M.J. Pérez-Jiménez: Application of fuzzy reasoning spiking neural P systems to fault diagnosis. *International Journal of Computers Communications & Control*, 9(6), 786–799, 2014.
31. G. Xiong, D. Shi, L. Zhu, X. Duan: A new approach to fault diagnosis of power systems using fuzzy reasoning spiking neural P systems. *Mathematical Problems in Engineering*, 2013(1), 211-244, 2013.

32. G. Xiong, D. Shi, J. Chen: Implementing fuzzy reasoning spiking neural P system for fault diagnosis of power systems. *Power & Energy Society General Meeting*, 1-8, 2013.
33. Y. Yahya, A. Qian, A. Yahya: Power transformer fault diagnosis using fuzzy reasoning spiking neural p systems. *Journal of Intelligent Learning Systems & Applications*, 08(4), 77-91, 2016.
34. G. Zhang, J. Wang, J. Cheng, T. Wang (Eds.), Pre-Proceedings of the Second Asian Conference on Membrane Computing (ACMC2013), Chengdu, China, 2013.
35. G. Zhang, J. Cheng, T. Wang, X. Wang, J. Zhu: Member computing: Theory and Application, Beijing: Science Press, 2015.

Open Problems, Research Topics

Open Problems in Membrane Computing and How Not to Solve Them

Alberto Leporati, Luca Manzoni, Giancarlo Mauri, Antonio E. Porreca, and Claudio Zandron

Dipartimento di Informatica, Sistemistica e Comunicazione
UniversitÃă degli Studi di Milano-Bicocca
Viale Sarca 336/14, 20126 Milano, Italy
{leporati,luca.manzoni,mauri,porreca,zandron}@disco.unimib.it

Summary. We present some high-level open problems in the complexity theory of membrane systems, related to the actual computing power of confluence vs determinism, semi-uniformity vs uniformity, deep vs shallow membrane structures, membrane division vs internal evolution of membranes. For each of these problems we present some reasonable approaches that are, however, unable to be employed "as-is" to provide a complete solution. This will hopefully sparkle new ideas that will allow tackling these open problems.

1 Introduction

It is common in most of the literature, not only of P systems but of science in general, to highlight the successful approaches, not only the ones that gave positive results, but also the ones that were successful in proving the negative ones. It is a lot less common to write about the unsuccessful ideas, the roads that, while initially promising, where unable to give a full and satisfactory answer to a research question under examination. Unfortunately, this leads multiple people to follow the same path multiple times without the insight that might have been gained by sharing previous failed attempts. Here we want to take the opportunity to share some problems that we consider of particular interest for the membrane computing community but, since they are still open problem, we are not able to share a solution to them. What we want to share is, instead, a collection of promising attempts that were ultimately unsuccessful in solving the open problems. While unsuccessful, those attempts where far from useless: we were able to gain a deeper understanding of the problems and to solve some special cases. We think that, by

sharing our non-working approaches we might be able to help the community in gaining a better understanding of these open problems and, hopefully, solve them.

2 Confluence vs determinism

P systems solving decision problems (*recogniser P systems* [15]) are usually required to be *confluent* [15] rather than strictly deterministic. That is, they are allowed to have multiple computation, as long as all of them agree on the final result, acceptance or rejection.

This sometimes simplifies the presentation of some algorithms. For instance, a classic membrane computing technique [13, 21] consists in generating all 2^n truth assignments of n variables by using membrane division rules of the form $[x_i]_h \rightarrow [t_i]_h\ [f_i]_h$, with $1 \leq i \leq n$. The membrane division is triggered separately in each membrane with label h by one of the objects x_i, nondeterministically chosen at each computation step. Irrespective of all such nondeterministic choices, the end result is invariably a set of 2^n membranes, each containing a different truth assignment.

Notice, however, that this kind of nondeterminism can be completely avoided by serialising the generation of truth assignments for each variable: first all instances of x_1 trigger the division, then all instances of x_2, and so on. This can be achieved by adding an extra subscript to each object, which counts down to zero and only then starts the division process.

It is often the case that confluent nondeterminism can be avoided in a similar way, although this is usually proved by exhibiting a deterministic algorithm, rather than showing how to remove the nondeterminism from existing algorithms. It is then natural to ask whether this is indeed always the case, or if there exists a variant of P system where confluent nondeterminism is strictly stronger than determinism.

For powerful enough P systems (e.g., able to efficiently simulate deterministic Turing machines, or stronger than that) we feel that the existence of such a variant would be very surprising, although there do exist confluent nondeterministic algorithms with no known deterministic version. For instance, the currently known proof of efficient universality (i.e., the ability to simulate any Turing machine with a polynomial slowdown) of P systems with active membranes using elementary membrane division [1] relies on a massive amount of nondeterministic choices performed at each simulated step; these are due to the fact that send-in communication rules cannot differentiate among membranes having the same label and electrical charges.

2.1 Simulation of priorities

To better examine the problem given by nondeterminism, even in the case of confluence, we will use a simple example. Take the following two rules:

$$[a]_h^+ \rightarrow [\,]_h^0\, b \tag{1}$$

$$[a \rightarrow w']_h^+ \tag{2}$$

When an instance of an object a is present inside a membrane with label h and positive charge, both rule (1) and rule (2) are applicable and, in a nondeterministic system, a nondeterministic choice is performed. If the P system is confluent, then the actual outcome of this choice is immaterial. However, if we know that the P system is a *deterministic* one then we can also infer a stronger condition: there will never be any instance of objects of type a inside any membrane with label h with charge + in the only computation starting from the initial configuration. If not, there will be two different computations generated by the presence of a negating the basic assumption that the P system is deterministic.

If we want to simulate a confluent system by means of a deterministic one in a somewhat direct way, then we must take care of situations like the one above. How can we simulate a situation that a deterministic P system cannot even reach? One such approach is the introduction of rule priorities. As shown in [9], for P systems with active membranes with charges the introduction of rule priorities *does not* change the computational power of confluent systems when the bound on computation time is at least polynomial.

Why it might work

Rule priorities provide a way of addressing the problem of the example above and the more general problem of conflict among rules. Suppose that in the previous example rule (1) had higher priority than rule (2). Then, there is no conflict between the two rules: the first one will always be applied if no other blocking rule with higher priority has already been applied and the second one will be applied to all remaining copies of a. In fact, once a total ordering has been provided among all rules then no further conflict can happen: among two rules one will always have higher priority than the other. Therefore, now in a deterministic system we can have objects of type a inside a membrane with label h and positive charge, since their presence will not generate two distinct computations anymore.

Why it does not work

If with rule priorities it is never possible to obtain two distinct computations due to a conflict between rules, since there are no conflicts anymore, then have we found a way to have one single computation starting from the initial configuration? Have we found a way to obtaining determinism from confluence? The answer is, unfortunately, negative. The main problem is given by *send-in* communication rules. Consider, for example the following rule:

$$a\,[\,]_h^+ \rightarrow [b]_h^- \tag{3}$$

Furthermore, suppose that there are two membranes with label h and positive charge inside the membrane where a is contained. If there is a single instance of a

then rule (3) can send in a in either of the two membranes in a nondeterministic way. If the content of the two membranes differs, then two computations were actually created. In a confluent system this is not a problem, but for a deterministic system the application of a send-in rule is valid in only two situations:

- There were enough instances of object a for it to be sent-in inside all the membranes where rule (3) was applicable;
- The contents of all the membranes where rule (3) was applicable were actually the same.

This problem cannot be solved by rule priorities; in fact, there was no rule conflict in the example that we just presented. This is the main difference between the nondeterminism introduced by send-in rules and the one introduced by rule conflicts. This remains the main obstacle in showing that confluence and determinism give, for powerful enough systems, the same computational power.

3 Semi-uniformity vs uniformity

Recogniser P systems usually appear in families $\boldsymbol{\Pi} = \{\Pi_x : x \in \Sigma^\star\}$, where each member of the family is associated to a string x and accepts if and only if x belongs to a given language.

A family of P systems is usually required to be at least *semi-uniform*, that it, to have an associated Turing machine M with some suitable resource bound (usually, polynomial time) such that M on input x outputs a suitable encoding of Π_x [15, 10].

A more restrictive condition on families of P systems is full-fledged *uniformity* [15, 10]: there exist *two* Turing machines F and E (again, usually with polynomial runtime) such that F on input $n = |x|$ constructs a P system "skeleton" Π_n, valid for all strings of length n, and E on input x produces a multiset w encoding x, which is then placed inside the input region of Π_n, giving the P system Π_x that computes the answer.

It is known [11] that, for restrictive enough resource bounds, uniformity is weaker than semi-uniformity. However, when polynomial-time semi-uniform solutions to problems sometimes appear in the literature first, polynomial-time uniform solutions usually follow.

We conjecture that polynomial-time uniformity and semi-uniformity do indeed coincide for powerful enough P systems, such as standard P systems with active membranes [13]. The idea here is that a semi-uniform family could be made uniform by simulating the "semi-uniform portion" of the construction, depending on the actual symbols of $x \in \Sigma^n$, with the P system constructed for all strings of length n.

3.1 Building and filling the membrane structure at runtime

Given a semi-uniform family of P systems $\boldsymbol{\Pi}$ constructed by a machine M we want to build a two machines E and F that define a uniform family of P systems that solves the same problems that are solved by the systems in $\boldsymbol{\Pi}$.

One of the possible ideas to prove the equivalence of uniformity and semi-uniformity, at least for powerful enough kinds of P systems, is to harvest the power of machine E of the uniformity condition, i.e., the one that has access to the entire input and not only to its length. While that machine does not have the ability construct the membrane structure of the P system or even to put object inside membranes different from the input one, it can perform the same operations of machine M of the semi-uniformity condition, therefore obtaining a "copy" of the initial membrane structure of the P system. Can we make use of such knowledge to overcome the limitation of uniformity and show that we can, in fact, simulate semi-uniformity?

Why it might work

With the knowledge of the initial membrane structure and the content it is quite easy to build, for each object type a new types of objects of the form a_p where p is a path inside the membrane structure. It is usually not hard to write rules "consuming" the path p while moving the objects around following the directions stored in p. Therefore, in polynomial time it is possible to move objects around the membrane structure. Since the objects in the initial system built by M on input x might depend on the input, we can delegate the creation of all the initial objects to machine E. The multiset produces as output of E on input x will contain all the objects present in the initial membrane structure built by M. However, that objects will be subscripted by a path as shown before. In that way they will be able to reach the correct position before starting to act like the object in the system built by M.

Why it does not work

A larger problem that the one tackled before is the fact that there is no assurance that the membrane structure generated by machine M will be the same on two different inputs x and y even when they are of the same length. We might think that machine F of the uniformity condition might be able to build a membrane structure that is the "sum" of all possible membrane structures that machine M can generate for inputs of a certain length. By looking at the literature it is possible to observe that this is an effective method to convert semi-uniform families to uniform ones. This method, however, will not work in general. A combinatorial analysis shows that the membrane structures that can be generated by machine M are too many to obtain a polynomially-sized membrane structure that contains all of them as substructures. The sharp contrast between the efficacy of this method in practice and its viability as a formal tool to prove the equivalence of uniformity and semi-uniformity leaves us with a question: is a super-polynomial number of

different membrane structure for inputs of the same length actually useful? Can we prove that this is never the case?

4 Membrane division vs internal evolution

The computing power of a *single* membrane (for cell-like P systems) or cell (for tissue-like P systems) working in polynomial time usually has a **P** upper bound, as already proved by the "Milano theorem" [21]; the only way to exceed this bound would be to include *really overly powerful* rules (e.g., rules able to perform an **NP**-complete task in a single step). The **P** upper bound can actually be achieved by having cooperative rewriting rules (even minimal cooperation [20, 19] suffices) or rules able to simulate them indirectly (e.g., active membrane rules with membrane charges [9]). Several techniques for simulating polynomial-time Turing machines using a single membrane are known [8].

Any additional power beyond **P** of models presented in the literature is due to membrane division, first exploited in order to solve **NP**-complete problems in polynomial time [13]. Membrane division enables us to create exponentially many processing units working in parallel; by using communication rules, these can synchronise and exchange information (this is the famous space-for-time trade-off in membrane computing).

It is reasonable to expect that P system variants where the power of a single membrane working in polynomial item coincides with **P** can be standardised in a "Turing machine normal form": each membrane performs a Turing machine simulation[1], and the communication and division rules implement a network, whose shape can be exploited to simulate nondeterminism, alternation, or oracle queries [8].

Notice that what previously described does not necessarily carry over to variants of P systems with weaker rules internal to the membranes, such as "P conjecture systems" [14, Problem F] (active membranes without charges), which do not seem able to simulate cooperation [2], or with communication restricted to a single direction, either send-out [6, 7, 18] or send-in only [17].

4.1 Putting a Turing machine inside a membrane

One approach that is at first glance promising to solve the problem of characterising the computational power of a single membrane is to simply substitute the content of a membrane with a Turing machine simulated by that membrane that makes its behaviour indistinguishable from the one of the original membrane. For an example, we will use P system with more than three charges, like the ones in [9].

Let M be a Turing machine with set of states Q, alphabet Σ, transition function δ, and working in space n. Then it can be simulated by a single membrane M_h with the following kinds of rules:

[1] This can be trivially implemented by having each membrane simulate a Turing machine which, in turn, simulates the original membrane via the Milano theorem.

$[a_i \to b_i\ r_d]^{q_i}_{M_h}$ for $q, r \in Q$, $a, b \in \Sigma$, $1 \leq i \leq n$, and $\delta(q, a) = (r, b, d)$

$[r_d]^{q_i}_{M_h} \to [\ ]^{r_{i+d}}_{M_h}\ \#$ for $q, r \in Q$, $r \in \{-1, 1\}$, and $1 \leq i \leq n$

The main idea is that in the objects inside the membrane it is possible to store the tape of the machine and the state and position of the tape head are store in the charge of M_h. By alternating evolution rules (to rewrite the tape) and send-outs (to update the state and position of the tape head) it is possible to simulate an entire Turing machine inside a single membrane. How can we use this to replace the entire inner working of a membrane?

Why it might work

If we consider an isolated membrane and we investigate its behaviour from the outside we are only interested in what is sent out and what is sent in. That is, if we are unable to distinguish a membrane h and a membrane simulating machine M that simulates h then the computational power of a single membrane cannot be greater than the one of machine M and, if M is a deterministic machine working in polynomial time then all the power of a P system comprised of such membranes must reside in the ability of the membranes to divide. Since most variant of P systems when limited to a single membrane are subjected to the Milano theorem, we can think that, with a careful enough "interfacing" with the objects sent in from the environment (which, when they enter, are not part of the machine tape and must be "incorporated" into it by other rules) we might be able to replicate (with a polynomial slowdown) the entire behaviour of a membrane h with the machine M simulated inside another membrane. And, in fact we can. So, why this does not prove that we can replace all the "inner working" of a membrane with a Turing machine?

Why it does not work

When considering a membrane in isolation we can easily replace it with a Turing machine - and we can replace the Turing machine with a membrane simulating it. There is, however, one major problem. Consider the following membrane structure:

$$[\underbrace{[]_k \cdots []_k}_{2^n}]_h$$

If we replace the "inner working" of membrane h with the simulation performed by a Turing machine, then at the moment when the membranes with label k send out - all at the same time - 2^k instances of the same object a then the simulation inside h is not sufficient anymore. It is not possible to write all the objects (either one at time of all together) on the Turing machine tape for reasons either of space (the tape is not long enough) or time (writing them one at a time requires exponentially many time steps). We would like to write, instead the number of objects of type a that has entered membrane h by send-out. This, however, requires the power to count or, more precisely, the power to convert from unary to binary the number of objects. While this is possible with cooperative rewriting rules, it is unclear if for P systems with charges this is a task feasible for a single membrane.

5 Deep vs shallow membrane structures

Let us now consider cell-like P systems with membrane division, for instance P systems with active membranes [13]. It has already been shown that the nesting depth of membranes (more specifically, the nesting depth of membranes with associated division rules, which we might call *division depth*) is one of the most influential variables when establishing the efficiency of these P systems.

Indeed, P systems without membrane division (i.e., with division depth 0) are known to characterise the complexity class **P** in polynomial time [21]. At the other end of the spectrum, we have P systems with active membranes with elementary and non-elementary division rules (i.e., with polynomial division depth), which characterise **PSPACE** in polynomial time.

When only elementary membrane division is allowed (i.e., division depth 1), then the intermediate complexity class $\mathbf{P}^{\#\mathbf{P}}$ is characterised in polynomial time [3, 7]. This class contains all decision problems solved by deterministic polynomial-time Turing machines with oracles for counting problems in the class $\#\mathbf{P}$ [12].

It has been proved that moving from any constant division depth d to division depth $d+1$ allows the P systems to simulate Turing machines with more powerful oracles [4]. We conjecture that this is in fact a proper hierarchy. This result would require proving the upper bounds corresponding to the known lower bounds.

It also remains open to characterise the computing power of polynomial-time P systems with other division depths, such as $O(\log n)$.

More profound reasons for the split

It seems that the problem of the computational power that depth gives to P systems is a more profound question related to the computational power that automata (or limited-resources Turing machines) can gain with different communication topologies. For example, with a polynomial time limit, automata on a grid are usually limited to power of a deterministic Turing machine working in polynomial time. Similar results also holds for tissue P systems embedded in the Euclidean space [5], showing another point of contact with the more general problem. This approach linking complexity and communication topologies is quite new [16] and we think that it might be able to give us a better understanding of the link between membrane depth and computational power in P system.

References

1. Alhazov, A., Leporati, A., Mauri, G., Porreca, A.E., Zandron, C.: Space complexity equivalence of P systems with active membranes and Turing machines. Theoretical Computer Science 529, 69–81 (2014)
2. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J.: On the power of dissolution in P systems with active membranes. In: Freund, R., Păun, G., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing, 6th International Workshop, WMC 2005. Lecture Notes in Computer Science, vol. 3850, pp. 224–240. Springer (2006)

3. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: Simulating elementary active membranes, with an application to the P conjecture. In: Gheorghe, M., Rozenberg, G., Sosík, P., Zandron, C. (eds.) Membrane Computing, 15th International Conference, CMC 2014. Lecture Notes in Computer Science, vol. 8961, pp. 284–299. Springer (2014)
4. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: Membrane division, oracles, and the counting hierarchy. Fundamenta Informaticae 138(1–2), 97–111 (2015)
5. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: Tissue P systems in the Euclidean space. In: Gheorghe, M., Petre, I., Pérez-Jiménez, M.J., Rozenberg, G., Salomaa, A. (eds.) Multidisciplinary Creativity: Homage to Gheorghe Păun on His 65th Birthday, pp. 118–128. Editura Spandugino (2015)
6. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: Monodirectional P systems. Natural Computing 15(4), 551–564 (2016)
7. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: The counting power of P systems with antimatter. Theoretical Computer Science 701, 161–173 (2017)
8. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: Subroutines in P systems and closure properties of their complexity classes. In: Graciani, C., Păun, G., Riscos-Núñez, A., Valencia-Cabrera, L. (eds.) 15th Brainstorming Week on Membrane Computing, pp. 115–128. No. 1/2017 in RGNC Reports, Fénix Editora (2017)
9. Leporati, A., Manzoni, L., Mauri, G., Porreca, A.E., Zandron, C.: A toolbox for simpler active membrane algorithms. Theoretical Computer Science 673, 42–57 (2017)
10. Murphy, N., Woods, D.: The computational power of membrane systems under tight uniformity conditions. Natural Computing 10(1), 613–632 (2011)
11. Murphy, N., Woods, D.: Uniformity is weaker than semi-uniformity for some membrane systems. Foundamenta Informaticae 134(1–2), 129–152 (2014)
12. Papadimitriou, C.H.: Computational Complexity. Addison-Wesley (1993)
13. Păun, G.: P systems with active membranes: Attacking NP-complete problems. Journal of Automata, Languages and Combinatorics 6(1), 75–90 (2001)
14. Păun, G.: Further twenty six open problems in membrane computing. In: Gutíerrez-Naranjo, M.A., Riscos-Nuñez, A., Romero-Campero, F.J., Sburlan, D. (eds.) Proceedings of the Third Brainstorming Week on Membrane Computing. pp. 249–262. Fénix Editora (2005)
15. Pérez-Jiménez, M.J., Romero-Jiménez, A., Sancho-Caparrini, F.: Complexity classes in models of cellular computing with membranes. Natural Computing 2(3), 265–284 (2003)
16. Porreca, A.E.: Communication topologies in natural computing (2018), seminar of the CANA research group, Laboratoire d'Informatique Fondamentale de Marseille, France, `https://aeporreca.org/talks/`
17. Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del Amor, M.A., Riscos-Núñez, A., Pérez-Jiménez, M.J.: Restricted polarizationless P systems with active membranes: Minimal cooperation only inwards. In: Graciani, C., Păun, G., Riscos-Núñez, A., Valencia-Cabrera, L. (eds.) 15th Brainstorming Week on Membrane Computing, pp. 215–252. No. 1/2017 in RGNC Reports, Fénix Editora (2017)
18. Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del Amor, M.A., Riscos-Núñez, A., Pérez-Jiménez, M.J.: Restricted polarizationless P systems with active membranes: Minimal cooperation only outwards. In: Graciani, C., Păun, G., Riscos-Núñez, A., Valencia-Cabrera, L. (eds.) 15th Brainstorming Week on Membrane Computing, pp. 253–290. No. 1/2017 in RGNC Reports, Fénix Editora (2017)

19. Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del-Amor, M.A., Riscos-Núñez, A., Pérez-Jiménez, M.J.: Computational efficiency of minimal cooperation and distribution in polarizationless P systems with active membranes. Fundamenta Informaticae 153(1–2), 147–172 (2017)
20. Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del-Amor, M.A., Riscos-Núñez, A., Pérez-Jiménez, M.J.: Reaching efficiency through collaboration in membrane systems: Dissolution, polarization and cooperation. Theoretical Computer Science 701, 226–234 (2017)
21. Zandron, C., Ferretti, C., Mauri, G.: Solving NP-complete problems using P systems with active membranes. In: Antoniou, I., Calude, C.S., Dinneen, M.J. (eds.) Unconventional Models of Computation, UMC'2K, Proceedings of the Second International Conference, pp. 289–301. Springer (2001)

Other Buds in Membrane Computing

Miguel A. Gutiérrez-Naranjo

Dept. of Computer Science and Artificial Intelligence, University of Seville, Spain
magutier@us.es

Paper written in honor of Mario Pérez-Jiménez,
on the occasion of his 70th birthday.

Summary. It is well-known the huge Mario's contribution to the development of Membrane Computing. Many young researchers may relate his name to the theory of complexity classes in P systems, the research of frontiers of the tractability or the application of Membrane Computing to model real-life situations as the Quorum Sensing System in *Vibrio fischeri* or the Bearded Vulture ecosystem. Beyond these research areas, in the last years Mario has presented many new research lines which can be considered as *buds* in the robust Membrane Computing tree. Many of them were the origin of new research branches, but some others are still waiting to be developed. This paper revisits some of these *buds*.

1 Introduction

Mario has contributed to the development of Membrane Computing in many research lines (see, e.g., [16, 24, 57]). From his early works on the formalization of transition P systems [53] or the links between P systems and diophantine sets [61], Mario has published dozens of papers on Membrane Computing. His contributions cover many different areas of the P systems research, from theoretical ones to real-life case studies applications, together with the development of different simulators or the proposal of many different P systems models. Many young researchers may relate his name to the theory of complexity classes in P systems [51], the research of frontiers of the tractability [46] or to the application of Membrane Computing to model real-life situations as the quorum sensing system in vibrio fischeri [59] or the bearded vulture ecosystem [9], but Mario's contributions go deeper in many other areas.

Beyond these strong branches in the robust Membrane Computing tree, there are many other research lines in Membrane Computing where Mario also has made a pioneer contribution. Some of these ideas were shortly developed in a few papers

and others only appear in the paper where they were presented. I call *buds* to these ideas. In this paper, I revisit some of these *buds* which are waiting for young researchers to be studied in deep.

The paper is organized as follows: In Section 2, some papers where Mario explored the links between Membrane Computing and Artificial Intelligence are revisited. They cover aspects related to sorting and searching algorithms and machine learning. Section 3 recalls other contributions bridging P systems with other aspects of Computer Science, as metrics defined on configurations of P systems, properties or Markov chains or the possibility of computing backwards. Later, Section 4 revisits some papers related to the graphical representation of P systems and Section 5 some other papers not included in the previous sections. Finally, some conclusions are added.

2 Artificial Intelligence

Mario is a full professor at the Department of Computer Science and Artificial Intelligence in the University of Seville, and both disciplines, Computer Science and Artificial Intelligence, have been present in Mario's research. We revisit some of these contributions.

2.1 Sorting

Sorting sequences of items according prefixed criteria is on the basis on many computational processes and this is a recurrent problem in Membrane Computing (see, e.g. [1, 2]). In the case of parallel algorithms, the problem is specially interesting, since it requires an appropriate combination of computation and communication.

In [14], two models for sorting sequences of numbers were presented. They were based on bitonic sorting networks. The key idea is the use of *bitonic mergesort* which is a parallel algorithm for sorting introduced by Batcher [3]. The first of the models consists on N membranes, each of them storing two numbers; one number is an element of the sequence, and the other one is an auxiliary register used to route values. A number is encoded as the multiplicity of a symbol a in each membrane. Moreover, membranes are disposed on a $2D$-mesh, where only communication between neighbor membranes on the mesh is permitted. This model uses a variant of P systems called *P systems with dynamic communication graphs* [11], which is closely related to the implementation of the bitonic sort on the $2D$-mesh[1]. The second model presented in [14] consisted on only one membrane, where all the N numbers were encoded as occurrences of N different symbols. Restrictions on communication were no longer imposed, as if the underlying communication graph were the complete graph. Later, a new model which takes ideas from both models were presented in [15].

[1] Such model of *P systems with dynamic communication graphs* is other of the many *buds* which is waiting to be explored. It follows the same lines as the presented one in [12, 13].

2.2 Searching

The design of solutions for NP problems in Membrane Computing usually trades time against space in order to solve these problems in polynomial (even lineal) time with respect to the size of the input data [51]. The cost is the number of resources, mainly the number of membranes, which grows exponentially. The usual idea of such brute force algorithms is to encode each feasible solution in one membrane. The number of candidates to solution is exponential in the input size, but the coding and checking process can be done in polynomial time. In spite of the theoretical success, such approaches are far to be applicable to real-world problems and other research lines must be explored. In this context, Mario has also made contributions by applying into the Membrane Computing framework some ideas from searching methods studied in Artificial Intelligence.

In [34], the problem of solving the N-queens puzzle with P systems was studied by considering a depth-first search strategy. Depth-first search is a very well-known algorithm for exploring tree or graph data structures. One starts at the root (selecting some arbitrary node as the root in the case of a graph) and explores as far as possible along each branch before backtracking. In the paper, the authors explore the possibilities of introducing such search strategy in the framework of Membrane Computing. The case study was the N-queens puzzle. Such problem is an old well-known problem. It is a generalization of the 8-queens problem which consists on placing N chess queens on a $N \times N$ board. In [30], a first solution to the N-queens problem in Membrane Computing had been shown, but the solution was based on an appropriate encoding of the problem in a SAT formula and the use of a modified solution for the SAT problem with P systems. The same problem of N-queens also was considered in [35] were some ideas from local search were studied in the framework of Membrane Computing. In such paper, from an initial $N \times N$ chessboard with N queens, different movements of the queens are performed in order to *improve* the position. Such improvements are measured by using the notion of *collision* [63]: *The number of collisions on a diagonal line is one less than the number of queens on the line, if the line is not empty, and zero if the line is empty. The sum of collisions on all diagonal lines is the total number of collisions between queens.* The target of the computation was to move from a board to another, with the corresponding encoding, by decreasing the number of *collisions*. Figure 1, borrowed from [35], illustrates the process.

2.3 Machine Learning

Mario has also made contributions by linking Membrane Computing to Machine Learning. In this section, two of them are revisited. The first one by taking ideas form *supervised learning* and the second one from *unsupervised learning*.

Spiking neural P systems (SN P systems) were introduced in 2006 (see [42]) with the aim of incorporating in Membrane Computing ideas specific to spike-based neural networks. Only two years later, a first model for Hebbian learning with spiking neural P systems was presented [32]. The target of this paper was

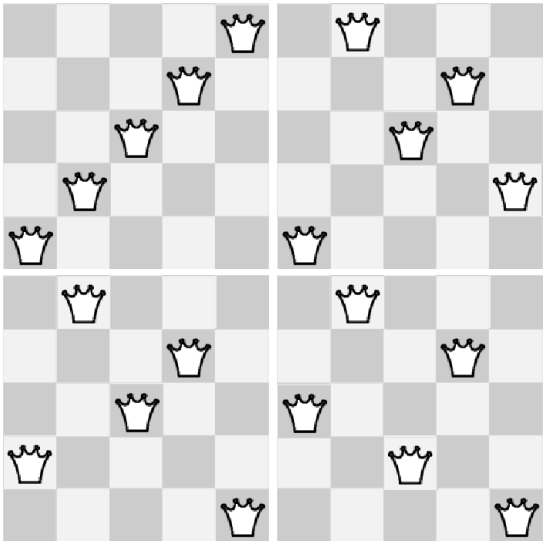


Fig. 1. Starting from a configuration C_0 with 4 collisions (up-left) we can reach C_1 with 3 collisions (up-right) and then C_2 with 2 collisions (bottom-left) and finally C_3 with 0 collisions (bottom-right), which is a solution to the 5-queens problem. Figure borrowed from [35].

to explore the applicability of ideas from the artificial neural networks into the SN P systems. Artificial neural networks [40] is one of the more powerful tools in Machine Learning and the most extended learning algorithm for such networks, backpropagation [62], can be roughly described as the iterative refinement of the weights associated to the synapses among neurons in order to minimize a loss function. Such change in the weights is inspired in the works made by Donald Hebb [41] and all the learning processes based in these principles are called *Hebbian Learning*. In [32], a first approach to Hebbian learning in the framework of Membrane Computing was presented. In such paper, a new feature coming from alive neurons was added to the SN P systems: the *decay*. Such decay of the electric potential inside an alive neuron along time was codified by endowing the rules with a finite non-increasing sequence of natural numbers called the decaying sequence. Beyond these sequences, the learning model is structured in *Hebbian SN system units*, which consider weights associated to the synapses between neurons. According to the learning process inspired by Hebb's work, the weights change along time according to the concept of *efficacy* introduced in the paper.

The second *bud* revisited in this section is related to the data clustering problem. In the paper [32] discussed above, Mario made an exploration by bridging Membrane Computing with supervised learning. The proposal in [50] links P systems with unsupervised learning.

The clustering algorithm presented in this paper is based on a tissue-like P system with a loop structure of cells. The objects of the cells express the candidate cluster centers and are evolved by the evolution rules. Based on the loop membrane structure, the communication rules realize a local neighborhood topology, which helps the co-evolution of the objects and improves the diversity of objects in the system. The tissue-like P system can effectively search for the optimal clustering partition with the help of its parallel computing advantage. The proposed clustering algorithm is evaluated on four artificial data sets and six real-life data sets. Experimental results show that the proposed clustering algorithm is superior or competitive to classical k-means algorithm and other evolutionary clustering algorithms.

3 Computer Science

Besides Artificial Intelligence, Mario has also contributed to bridge Membrane Computing with many other areas in Computer Science.

As a first example, we can consider the study of metrics on configurations presented in [19]. In such paper, two different (weak) metrics were presented. The first one was based on the distance between regions. The distance between two regions was defined as the cardinality of the symmetrical difference of their associated multisets. This definition was used to measure the distance between two occurrences of the same membrane in two different configurations and the difference between configurations is the sum of the differences between their regions. For the definition of the second weak metric, a new auxiliary concept called *dependency graph* was defined. Such concept has been widely used for studying frontiers on complexity classes (see, e.g., [38]), but it was firstly defined in this paper for studying the proximity between configurations. The length between two nodes of the dependency graph was on the basis of the study of proximity between configurations.

Other remarkable Mario's contribution which is still a *bud* in the Membrane Computing tree is related to the link between P systems and Markov chains. In [10], the authors propose a first approach to the problem of computing the natural powers of the transition matrix of a finite and homogeneous Markov chain. Such computation allows to estimate its limit in the case that it is convergent, and therefore, to know the stationary distribution of the process. This subject had been treated with other bio-inspired techniques in [7] where two algorithms based on DNA were described. The proposed cellular computing solution provides an exact solution in a time which is linear in the order of the power and it is independent of the number of states of the Markov chain. Such Markov chains were also considered later in [8]. In such paper, the aperiodicity of irreducible Markov chains was characterized by using P systems.

A different problem was studied in [33], where the problem of computing backwards with P systems was considered. The starting point for this study was to wonder about the previous state of a given one in a computational model where

the time is considered in a discrete way. In such study, the authors consider a variant of the dependency graphs [18] for introducing a representation of the computation of a P system based on matrices. Such matrix representation opens a door for the study of algebraic properties of cell-like and tissue-like P systems and represents one of the most promising *buds* waiting to be explored. Mario has also contributed to the study of the matrix representation of spiking neural P systems [66] which is on the basis of the efficient simulation of such devices [5].

4 Graphics

Many P system models allow to change its membrane structure by adding new membranes (e.g., by the application of *creation* or *division* rules [52]) or removing membranes (by the application of *dissolution* rules [39]). Such evolution in time of the membrane structure of a P system is the starting point for studying the evolution of graphical structures.

The first approach for linking the computation of P systems with the evolution of graphical structures was presented in [25, 26]. In these papers, P systems were used to simulate the growth and development of living plants. This approach mixes L systems [23, 45] and P systems. A different approach was presented in [60]. In this paper, the growth of branching structures was studied by using exclusively P systems. The key idea is the use of a cell-like P system model with evolution and creation rules. The membrane structure of a cell-like model is a tree-like graph which is easily visualized as a branching structure. The geometrical properties of the associated picture can be obtained by the association of *meaning* to the objects in the multisets in the different membranes. For example, each membrane represents a segment in the corresponding picture and the length of the drawn segment depends on the multiplicity of a *length-unit* object. These ideas were also considered in [58], where a specific software for this graphical representation was developed. Some examples of polygons, spirals, friezes and plants can be found in this paper. Figure 2 shows some of them.

A different approach was presented in [31]. It was also related with the graphical representation of P systems, but it explores the possibility of Membrane Computing devices for representing fractals. The starting point here is that a fractal can be considered, roughly speaking, as a self-similar geometrical structure which can be generated by the application of (infinitely repeated) fixed rules. In this way, the generation of fractals can be associated, on the one hand, to the evolution of P systems by using creation rules for obtaining a new (and more precise) stage in the generation of the fractal and, on the other hand, to the *interpretation* of the symbols inside the membranes for representing the *geometrical information* of the fractal. The paper presents a pair of classic fractals, the Koch curve [43, 44] and the middle third Cantor set [6], in the framework of P systems. Besides this mathematical objects, the paper also points out the possibility of using the non-

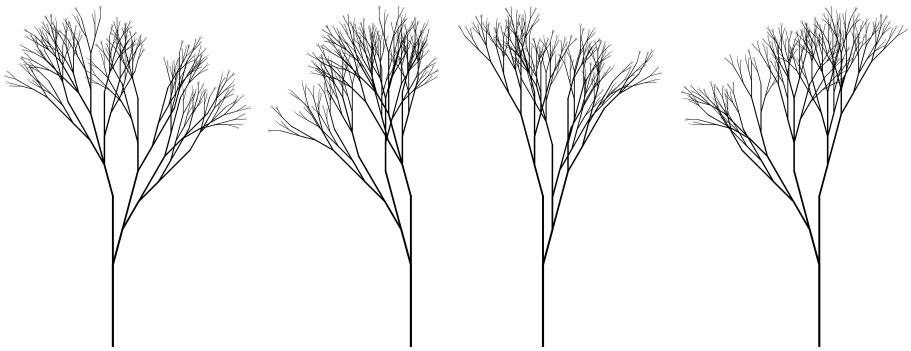


Fig. 2. Graphical representation of four configurations of a P system. Images borrowed from [58].

determinism of P systems for studying random fractals, which can be considered as the formal representation of many real-life objects with fractal dimension[2].

5 Other Buds

The contributions made by Mario cover many different areas, some of them by bridging Membrane Computing with other unexpected research fields. One of these *rara avis* can be found in [28] where a Membrane Computing model for ballistic depositions was presented. The starting point in this study is the evolution of *rough interfaces* between different media. The propagation of forest fires [17] or the growth of a colony of bacteria [4] can be examples showing such interfaces, although all surfaces in Nature can be seen as rough surfaces, since the concept of roughness is associated to the scale and all the natural surfaces are rough at appropriate scale. The evolution of a surface can be modelled by the concepts of *erosion* where some elements are removed or *deposition* where new elements are placed. Ballistic Deposition was proposed by Vold [65] and Sutherland [64] as a model where as the particle follows a straight vertical trajectory until it reaches the surface, whereupon it sticks (see Figure 3). In [28] the problem was modelled by a tissue-like P system model with a linear membrane structure where each cell represents a column of the aggregate and the pieces of information needed for encoding the growth process are encoded on the multisets of objects in the cells.

Other of the explorations looking for links between Membrane Computing and other research areas was presented in [54]. In this case, the target was to bridge P systems and *reaction systems*. Reaction systems, *R systems*, is a bio-inspired computation model [21, 22] which shares with P systems some features as the use

[2] These ideas were also considered in [37], but it is still one of the most promising *buds* waiting to be developed.

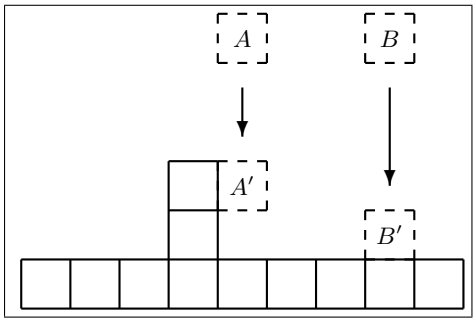


Fig. 3. Ballistic Deposition. Figure borrowed from [28].

of populations of reactants (molecules) which evolve by means of reactions. This paper compares both computation models and further results can be found in [56].

Another theoretical *bud* can be found in [36]. In this paper, the degree of parallelism in P systems is studied. The starting point is to study different tools for comparing the design of P systems able to perform the same task. Two design can be compared according to many different criteria. In this paper, the authors focus the attention on the parallelism. In this way, a *bad* design of a P system consists on a P system which does not exploit its parallelism, that is, it works as a sequential machine: in each step only one object evolve in one membrane whereas the remaining objects do not evolve. On the other hand, a *good* design consists on a P system in which a huge amount of objects are evolving simultaneously in all membranes. If both P systems perform the same task, it is obvious that the second one is a better design that the first one.

6 Conclusions

Mario's contribution to the development of Membrane Computing has been enormous and in the last years he has been one of the pillars of the Membrane Computing community[3]. The research lines opened by him cover all the research fields in Membrane Computing. Many of his proposals have been studied in deep, but some of them are still waiting. In this paper, we have revisited some of them but many others have not been cited in the previous sections. I encourage to young researchers to read Mario's contributions cited in this paper, but not only they. Many other papers not cited here have seminal ideas which are waiting for young and enthusiastic researchers which help these *buds* to develop and became new strong branches in the Membrane Computing tree.

[3] At the moment of writing this paper (April 2018), according to Scopus, Mario has published with 150 co-authors. This fact can give an idea of Mario's role as *engine* of the P systems community.

References

1. Alhazov, A., Sburlan, D.: Static sorting P systems. In: Ciobanu et al. [16], pp. 215–252, https://doi.org/10.1007/3-540-29937-8_8
2. Ardelean, I.I., Ceterchi, R., Tomescu, A.I.: Sorting with P systems: A biological perspective. Romanian Journal of Information Science and Technology 11(3), 243–252 (2008), http://www.imt.ro/romjist/Volum11/Number11_3/pdf/03-Ardelean.pdf
3. Batcher, K.E.: Sorting networks and their applications. In: American Federation of Information Processing Societies: AFIPS Conference Proceedings: 1968 Spring Joint Computer Conference, Atlantic City, NJ, USA, 30 April - 2 May 1968. AFIPS Conference Proceedings, vol. 32, pp. 307–314. Thomson Book Company, Washington D.C. (1968), http://doi.acm.org/10.1145/1468075.1468121
4. Ben-Jacob, E., Schochet, O., Tenenbaum, A., Cohen, I., Czirok, A., Vicsek, T., et al.: Generic modelling of cooperative growth patterns in bacterial colonies. Nature 368(6466), 46–49 (1994), https://www.nature.com/articles/368046a0
5. Cabarle, F.G., Adorna, H., Martínez-del-Amor, M.A.: A spiking neural P system simulator based on CUDA. In: Gheorghe, M., Păun, Gh., Rozenberg, G., Salomaa, A., Verlan, S. (eds.) Membrane Computing - 12th International Conference, CMC 2011, Fontainebleau, France, August 23-26, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7184, pp. 87–103. Springer (2011), https://doi.org/10.1007/978-3-642-28024-5_8
6. Cantor, G.: Über unendliche, lineare punktmannichfaltigkeiten v. Mathematische Annalen 21, 545–591 (1883), http://dx.doi.org/10.1007/BF01446819, 10.1007/BF01446819
7. Cardona, M., Colomer, M., Conde, J., Miret, J., Mir, J., Zaragoza, A.: Markov chains: Computing limit existence and approximations with DNA. Biosystems 81(3), 261–266 (2005), http://www.sciencedirect.com/science/article/pii/S0303264705000663
8. Cardona, M., Colomer, M.A., Pérez-Jiménez, M.J.: Characterizing the aperiodicity of irreducible Markov chains by using P systems. In: Gutiérrez-Escudero et al. [29], pp. 81–95, https://idus.us.es/xmlui/handle/11441/38857
9. Cardona, M., Colomer, M.A., Pérez-Jiménez, M.J., Sanuy, D., Margalida, A.: Modeling ecosystems using P systems: The bearded vulture, a case study. In: Corne, D.W., Frisco, P., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing - 9th International Workshop, WMC 2008, Edinburgh, UK, July 28-31, 2008, Revised Selected and Invited Papers. Lecture Notes in Computer Science, vol. 5391, pp. 137–156. Springer, Berlin Heidelberg (2008), https://doi.org/10.1007/978-3-540-95885-7_11
10. Cardona, M., Colomer, M.Á., Pérez-Jiménez, M.J., Zaragoza, A.: Handling Markov chains with membrane computing. In: Gutiérrez-Naranjo, M.A., Păun, Gh., Riscos-Núñez, A., Romero-Campero, F.J. (eds.) Fourth Brainstorming Week on Membrane Computing. vol. I, pp. 99–111. Fénix Editora, Sevilla, Spain (January 30 - February 3 2006), https://idus.us.es/xmlui/handle/11441/36996
11. Ceterchi, R., Pérez-Jiménez, M.J.: On two-dimensional mesh networks and their simulation with P systems. In: Mauri et al. [49], pp. 259–277, https://doi.org/10.1007/978-3-540-31837-8_15
12. Ceterchi, R., Pérez-Jiménez, M.J.: A perfect shuffle algorithm for reduction processes and its simulation with P systems. International Conference on Computers

and Communications pp. 92–97 (June 1-3 2004), http://www.cs.us.es/~marper/investigacion/oradea.pdf
13. Ceterchi, R., Pérez-Jiménez, M.J.: Simulating shuffle-exchange networks with P systems. In: Gheorghe Păun et al. [55], pp. 117–129, https://idus.us.es/xmlui/handle/11441/34641
14. Ceterchi, R., Pérez-Jiménez, M.J., Tomescu, A.I.: Simulating the bitonic sort using P systems. In: Eleftherakis, G., Kefalas, P., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing, 8th International Workshop, WMC 2007, Thessaloniki, Greece, June 25-28, 2007 Revised Selected and Invited Papers. Lecture Notes in Computer Science, vol. 4860, pp. 172–192. Springer, Berlin Heidelberg (2007), https://doi.org/10.1007/978-3-540-77312-2_11
15. Ceterchi, R., Pérez-Jiménez, M.J., Tomescu, A.I.: Sorting omega networks simulated with P systems: Optimal data layouts. In: Díaz-Pernil et al. [20], pp. 79–92, https://idus.us.es/xmlui/bitstream/handle/11441/68018/dl-ps-4a.pdf
16. Ciobanu, G., Pérez-Jiménez, M.J., Păun, Gh. (eds.): Applications of Membrane Computing. Natural Computing Series, Springer (2006)
17. Clar, S., Drossel, B., Schwabl, F.: Forest fires and other examples of self-organized criticality. Journal of Physics: Condensed Matter 8(37), 6803 (1996), http://stacks.iop.org/0953-8984/8/i=37/a=004
18. Cordón-Franco, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Exploring computation trees associated with P systems. In: Mauri et al. [49], pp. 278–286, https://doi.org/10.1007/978-3-540-31837-8_16
19. Cordón-Franco, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Weak metrics on configurations of a P system. In: Gheorghe Păun et al. [55], pp. 139–151, https://idus.us.es/xmlui/handle/11441/34641
20. Díaz-Pernil, D., Graciani, C., Gutiérrez-Naranjo, M.A., Păun, Gh., Pérez-Hurtado, I., Riscos-Núñez, A. (eds.): Sixth Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla, España (2008)
21. Ehrenfeucht, A., Rozenberg, G.: Basic notions of reaction systems. In: Calude, C., Calude, E., Dinneen, M.J. (eds.) Developments in Language Theory, 8th International Conference, DLT 2004, Auckland, New Zealand, December 13-17, 2004, Proceedings. Lecture Notes in Computer Science, vol. 3340, pp. 27–29. Springer (2004), https://doi.org/10.1007/978-3-540-30550-7_3
22. Ehrenfeucht, A., Rozenberg, G.: Reaction systems. Fundamenta Informaticae 75(1-4), 263–280 (2007), http://content.iospress.com/articles/fundamenta-informaticae/fi75-1-4-15
23. Frijters, D., Lindenmayer, A.: A model for the growth and flowering of aster novae-angliae on the basis of table $< 1, 0 >$ L-systems. In: Rozenberg, G., Salomaa, A. (eds.) L Systems, Most of the papers were presented at a conference in Aarhus, Denmark, January 14-25, 1974. Lecture Notes in Computer Science, vol. 15, pp. 24–52. Springer (1974), http://dx.doi.org/10.1007/3-540-06867-8_2
24. Frisco, P., Gheorghe, M., Pérez-Jiménez, M.J.: Applications of Membrane Computing in Systems and Synthetic Biology. Springer Publishing Company, Incorporated (2014)
25. Georgiou, A., Gheorghe, M.: Generative devices used in graphics. In: Alhazov, A., Martín-Vide, C., Păun, Gh. (eds.) Preproceedings of the Workshop on Membrane Computing. Technical Report, Vol. 28/03. pp. 266–272. Research Group on Mathematical Linguistics, Universitat Rovira i Virgili, Tarragona, Spain (2003)

26. Georgiou, A., Gheorghe, M., Bernardini, F.: Membrane-based devices used in computer graphics. In: Ciobanu, G., Păun, Gh., Pérez-Jiménez, M.J. (eds.) Applications of Membrane Computing, pp. 253–281. Natural Computing Series, Springer Berlin Heidelberg (2006), http://dx.doi.org/10.1007/3-540-29937-8_9
27. Graciani, C., Păun, Gh., Romero-Jiménez, A., Sancho-Caparrini, F. (eds.): Fourth Brainstorming Week on Membrane Computing, vol. II. Fénix Editora, Sevilla, Spain (2006)
28. Graciani-Díaz, C., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: A membrane computing model for ballistic depositions. In: Gutiérrez-Naranjo, M.A., Păun, G., Romero-Jiménez, Á., Riscos-Núñez, A. (eds.) Fifth Brainstorming Week on Membrane Computing. pp. 179–197. Fénix Editora, Sevilla, España (1/29/07-2/2/07 2007), https://idus.us.es/xmlui/handle/11441/38583
29. Gutiérrez-Escudero, R., Gutiérrez-Naranjo, M.A., Păun, Gh., Pérez-Hurtado, I., Riscos-Núñez, A. (eds.): Seventh Brainstorming Week on Membrane Computing, vol. I. Fénix Editora, Sevilla, Spain (2009)
30. Gutiérrez-Naranjo, M.A., Martínez-del-Amor, M.A., Pérez-Hurtado, I., Pérez-Jiménez, M.J.: Solving the N-queens puzzle with P systems. In: Gutiérrez-Escudero et al. [29], pp. 199–210, https://idus.us.es/xmlui/handle/11441/38865
31. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Fractals and P systems. In: Graciani et al. [27], pp. 65–86, https://idus.us.es/xmlui/handle/11441/38328
32. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: A first model for Hebbian learning with spiking neural P systems. In: Díaz-Pernil et al. [20], pp. 211–233, https://idus.us.es/xmlui/handle/11441/38778
33. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Computing backwards with P systems. In: Gutiérrez-Escudero et al. [29], pp. 211–226, https://idus.us.es/xmlui/handle/11441/38866
34. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Membrane computing meets artificial intelligence: A case study. In: Martínez del Amor et al. [47], pp. 133–144, https://idus.us.es/xmlui/handle/11441/39004
35. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Implementing local search with membrane computing. In: Martínez del Amor et al. [48], pp. 159–168, https://idus.us.es/xmlui/handle/11441/39476
36. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A.: An approach to the degree of parallelism in P systems. In: Graciani-Díaz et al. [27], pp. 87–104, https://idus.us.es/xmlui/handle/11441/38335
37. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J.: How to express tumours using membrane systems. Progress in Natural Science 17(4), 449–457 (2007), http://www.tandfonline.com/doi/abs/10.1080/10020070708541022
38. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J.: On the power of dissolution in P systems with active membranes. In: Freund, R., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) Workshop on Membrane Computing. Lecture Notes in Computer Science, vol. 3850, pp. 224–240. Springer, Berlin Heidelberg (2005), https://doi.org/10.1007/11603047_16
39. Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J.: Computational efficiency of dissolution rules in membrane systems. International Journal of Computer Mathematics 83(7), 593–611 (2006), https://doi.org/10.1080/00207160601065413

40. Haykin, S.S.: Neural networks and learning machines. Pearson Education, Upper Saddle River, NJ, third edn. (2009)
41. Hebb, D.O.: The organization of behavior: A neuropsychological theory. Wiley, New York (Jun 1949)
42. Ionescu, M., Păun, Gh., Yokomori, T.: Spiking neural P systems. Fundamenta Informaticae 71(2-3), 279–308 (2006), http://content.iospress.com/articles/fundamenta-informaticae/fi71-2-3-08
43. Koch, H.v.: Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire. Arkiv för Matematik, Astronomi och Fysik 1, 681–704 (1904)
44. von Koch, H.: Une méthode géomtrique élémentaire pour létude de certaines questions de la théorie des courbes planes. Acta Mathematica 30, 145–174 (1906), http://dx.doi.org/10.1007/BF02418570
45. Lindenmayer, A.: Developmental systems without cellular interactions, their languages and grammars. Journal of Theoretical Biology 30(3), 455 – 484 (1971)
46. Macías-Ramos, L.F., Martínez-del-Amor, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Valencia-Cabrera, L.: Unconventional approaches to tackle the P *Versus* NP problem. In: Gheorghe Păun, Rozenberg, G., Salomaa, A. (eds.) Discrete Mathematics and Computer Science. In Memoriam Alexandru Mateescu (1952-2005). pp. 223–237. The Publishing House of the Romanian Academy (2014)
47. Martínez del Amor, M.A., Păun, Gh., Pérez Hurtado, I., Riscos-Núñez, A. (eds.): Eighth Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla, Spain (2010)
48. Martínez del Amor, M.A., Păun, Gh., Pérez Hurtado, I., Romero Campero, F.J., Valencia-Cabrera, L. (eds.): Ninth Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla, Spain (2011)
49. Mauri, G., Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G., Salomaa, A. (eds.): Membrane Computing, 5th International Workshop, WMC 2004, Milan, Italy, June 14-16, 2004, Revised Selected and Invited Papers, Lecture Notes in Computer Science, vol. 3365. Springer (2005)
50. Peng, H., Zhang, J., Wang, J., Wang, T., Pérez-Jiménez, M.J., Riscos-Núñez, A.: Membrane clustering: A novel clustering algorithm under membrane computing. In: Macías-Ramos, L.F., Martínez-del-Amor, M.A., Păun, Gh., Riscos-Núñez, A., Valencia-Cabrera, L. (eds.) Twelfth Brainstorming Week on Membrane Computing. pp. 311–327. Fénix Editora, Sevilla, España (February 2014), https://idus.us.es/xmlui/handle/11441/33577
51. Pérez-Jiménez, M.J.: A computational complexity theory in membrane computing. In: Păun, Gh., Pérez-Jiménez, M.J., Riscos-Núñez, A., Rozenberg, G., Salomaa, A. (eds.) Workshop on Membrane Computing. Lecture Notes in Computer Science, vol. 5957, pp. 125–148. Springer, Berlin Heidelberg (2009), https://doi.org/10.1007/978-3-642-11467-0_10
52. Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Jiménez, A., Woods, D.: Complexity - membrane division, membrane creation. In: Păun et al. [57], pp. 302 – 336
53. Pérez-Jiménez, M.J., Sancho-Caparrini, F.: A formalization of transition P systems. Fundamenta Informaticae 49(1-3), 261–272 (2002), http://content.iospress.com/articles/fundamenta-informaticae/fi49-1-3-20
54. Gheorghe Păun, Pérez-Jiménez, M.J.: Towards bridging two cell-inspired models: P systems and r systems. In: Martínez del Amor et al. [48], pp. 305–316, https://idus.us.es/xmlui/handle/11441/39575

55. Gheorghe Păun, Riscos-Núñez, A., Romero-Jiménez, Á., Sancho-Caparrini, F. (eds.): Second Brainstorming Week on Membrane Computing. Fénix Editora, Sevilla, Spain (2004)
56. Păun, Gh., Pérez-Jiménez, M.J., Rozenberg, G.: Bridging membrane and reaction systems further results and research topics. In: Valencia-Cabrera, L., García-Quismondo, M., Macías-Ramos, L.F., Martínez-del-Amor, M.A., Păun, Gh., Riscos-Núñez, A. (eds.) Eleventh Brainstorming Week on Membrane Computing. pp. 243–256. Fénix Editora, Sevilla, España (February 2013), https://idus.us.es/xmlui/handle/11441/33819
57. Păun, Gh., Rozenberg, G., Salomaa, A. (eds.): The Oxford Handbook of Membrane Computing. Oxford University Press, Oxford, England (2010)
58. Rivero-Gil, E., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: Graphics and P systems: Experiments with JPLANT. In: Díaz-Pernil et al. [20], pp. 241–253, https://idus.us.es/xmlui/handle/11441/38787
59. Romero-Campero, F.J., Pérez-Jiménez, M.J.: A model of the quorum sensing system in *Vibrio fischeri* using P systems. Artificial Life 14(1), 95–109 (2008), https://doi.org/10.1162/artl.2008.14.1.95
60. Romero-Jiménez, A., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J.: The growth of branching structures with P systems. In: Graciani et al. [27], pp. 253–265, http://www.gcn.us.es/4BWMC/vol2/thegrowth.pdf
61. Romero Jiménez, Á., Pérez-Jiménez, M.J.: Generation of diophantine sets by computing P systems with external output. In: Calude, C., Dinneen, M.J., Peper, F. (eds.) Unconventional Models of Computation, Third International Conference, UMC 2002, Kobe, Japan, October 15-19, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2509, pp. 176–190. Springer (2002), https://doi.org/10.1007/3-540-45833-6_15
62. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. Nature 323, 533–536 (1986)
63. Sosic, R., Gu, J.: Efficient local search with conflict minimization: A case study of the N-queens problem. IEEE Transactions on Knowledge and Data Engineering 6(5), 661–668 (1994)
64. Sutherland, D.: Comments on Vold's simulation of floc formation. Journal of Colloid and Interface Science 22(3), 300 – 302 (1966), http://www.sciencedirect.com/science/article/pii/0021979766900373
65. Vold, M.J.: A numerical approach to the problem of sediment volume. Journal of Colloid Science 14(2), 168 – 174 (1959), http://www.sciencedirect.com/science/article/pii/0095852259900418
66. Zeng, X., Adorna, H., Martínez-del-Amor, M.A., Pan, L., Pérez-Jiménez, M.J.: Matrix representation of spiking neural P systems. In: Gheorghe, M., Hinze, T., Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) Membrane Computing - 11th International Conference, CMC 2010, Jena, Germany, August 24-27, 2010. Revised Selected Papers. Lecture Notes in Computer Science, vol. 6501, pp. 377–391. Springer, Berlin Heidelberg (2010), https://doi.org/10.1007/978-3-642-18123-8_29

Towards a General Framework for Membrane Algorithms

José Antonio Andreu-Guzmán[1] and Luis Valencia-Cabrera[1]

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla, Spain
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
josandguz@gmail.com,lvalencia@us.es

Summary. Optimization problems have always been, and nowadays even more than ever, a subject of major concern, given their influence in the solution of very relevant daily matters (*i.e.*, worldwide packaging and transport, involving problems like `KNAPSACK` or `TSP`). Many classical meta-heuristics have been proposed to solve these kinds of problems, from simulated annealing to evolutionary algorithms or collective intelligence approaches like ant or bee colonies. Membrane computing brought an alternative set of hybrid algorithms taking the best of P systems and *classical* meta-heuristics in an attempt to improve the previous approaches. However, the heterogeneity of those approaches also brought some mess to the discipline, and a thesis project was proposed to try to formalise a general conceptual framework for membrane algorithms, and also provide a first set of tools inside P-Lingua for the simulation of those approaches formalised within the framework.

1 Introduction

Humanity has always faced challenges throughout its history. Along with the need of solving whatever problems emerging during the evolution, comes an inherent desire of moving forward and set major achievements as individuals and as a society. In this sense, the development of science and technology over the last centuries has stimulated the improvements in all areas of life. Thus, theoretical findings enable qualitative jumps in the resolution of daily-life problems faced by individuals, companies, institutions and countries. For instance, if we pay attention to the extent the packages transport and delivery is increasing worldwide, we will notice how important optimizing these processes is becoming, from the point of view of space utilization, or time/fuel consumption, among others. As already mentioned, in the essence of these real problems we can find very well known optimization problems as `KNAPSACK` [8] or `TSP` [9].

There exist many possible algorithms dealing with this kind of problems, ranging from brute-force algorithms to artificial intelligence techniques based on classical search variants, meta-heuristics, collective intelligence approaches or deep learning methods. No solution is perfect in any situation, so we must take into account the advantages and drawbacks involved. Thus, while the first approaches based on brute-force algorithms generally offered exact solutions whenever it was possible, there are many scenarios where those solutions would be impractical,non-tractable from a computational point of view, so that big instances relevant for real applications could not be processed in the reasonable time required. On the other hand, more sophisticated techniques can provide solutions being *good enough* depending on the purposes, not guaranteeing finding the perfect solution but getting as close as possible given a set of computational restrictions in terms of space and time.

Within this context, probably the most widely used approaches to address these optimization problems during the last few decades have been based on classical meta-heuristics iteratively enhancing a solution or set of solutions until reaching the halting condition set. A bunch of these meta-heuristics have been proposed along the years, from simulated annealing [7] to many variants of evolutionary algorithms [5]. Additionally, different forms of collective intelligence approaches have also appeared, from particle swarm optimization to ant and bee colonies, generally inspired by nature, and in other cases by physics (see quantum-inspired computing).

Since the proposal by Nishida in [10], the previous efforts were combined with the new ideas brought by Membrane Computing [13], showing relevant contributions in both areas. As we will detail in the next sections, this thesis project aims to progress in this direction, from both a theoretical and practical point of view, from the formalization and organized classification to the development of sotware tools for their practical application.

2 General idea

Taking into account the context outlined in the previous section, we thought it would be interesting to organize the main approaches detected in membrane algorithms and provide a general formalization, a global conceptual framework and a first set of general-purpose software tools to handle these systems through the infrastructure provided by `P-Lingua` [3] and `MeCoSim` [14]. These are the main goals of the thesis project where its author, José Antonio Andreu Guzmán, is working. More specifically, the focus of the thesis is to provide a review of all contributions to membrane algorithms, classify and extract conclusions about the usefulness, the opportunities emerged from the synergy between Membrane Computing [12] and approximate algorithms to solve **NP**-complete problems (these including *classical* meta-heuristics, different branches in Evolutionary Computing[1], forms of collective intelligence and so on).

The aim of this previous review and organized classification is to set a first step towards a conceptual framework unifying the plethora of techniques based on the refinement of a set of tentative approximate solutions to a problem through a collective effort of different *individuals* cooperating. The idea would be producing a global formal definition encompassing the different membrane algorithms (MA)[10, 19]. This could provide something similar to the general formalized model existing for the multienvironment P systems, as shown in 1.

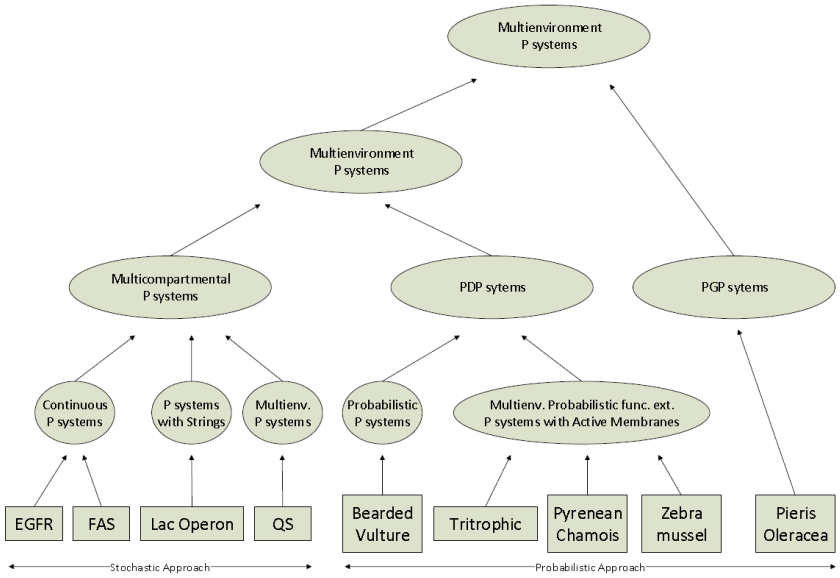


Fig. 1. A general framework for multi-environment P systems

As we can observe and recall, that view collected within the same definition different types of membrane devices characterized by the presence of many P systems distributed among a set of environment or compartments. Initially, each variant (multicompartmental, PDP, PGP systems and so on) had been formalized in a different way, with its own ingredients making them potentially useful for different scenarios; however, a general definition was provided as a unifying framework to comprehend them all and have a clearer overview. Thus, creating this general structure for membrane algorithms would be advisable for the development and visibility of this set of algorithms. This would make it easier for new collaborators or students to understand the main principles, and to decide which MA is better for each situation. Additionally, it could be useful as a helper tool depicting the global scene making easier to decide at first sight which variants or branches might be potentially worth being further explored, as it is another goal of our work: designing alternative types and variants of membrane algorithms, analysing the benefits and drawbacks of considering certain structural and/or functional ingredients.

Besides the inherent advantage of that clarifying view just mentioned, the general formal definition would ease the extension of P-Lingua to incorporate membrane algorithms, initially including some of these hybrid methods with elements from Ant colony Optimization (ACO) [2], Quantum-Inspired Evolutionary Algorithm (QIEA) [16], Genetic Algorithm (GA)[4, 5], Differential Evolution (DE) [15] or Particle Swarm Optimization [6] (PSO), among others. Therefore, another major goal of this thesis project is to provide new extensions in the language, parsers, simulators and accompanying tools within P-Lingua framework for some variants of membrane algorithms, along with possible improvements in MeCoSim to facilitate the work with MA, thus triggering the evolution of this kind of algorithms and the development of the discipline. The provision of visualization tools to follow the simulations of the new variants of MA proposed within the project would aid in the design of solutions and the analysis of the results obtained when testing with certain benchmark problems.

To sum up, amongst the goals of the thesis we intend to offer a perspective of the advantages and limitations of the different types of MA in opposition to classic meta-heuristics, providing a vision of the state-of-the-art of the discipline. Additionally, we will propose variants to improve the performance of certain types of MA, presenting the profit in execute time of the application of MA in parallel computers provided by this improved solutions.

3 Current status

From the beginning of our work we intended not to have a prefixed idea regarding the general formal model to define nor the structure of our classification. Bearing in mind that focus, we started with a first review of the bibliography related with membrane algorithms and their corresponding *classical* predecessors. Thus, we considered the valuable collections provided in [18] and started analysing the different types of algorithms detailed there.

Thus, while deepening into the analysis of the different techniques and progressively building the general formal model, we decided to start paying attention to those types of MA where the density of previous papers was lower, hence offering a bigger space for further studies and potential improvements. Thus, it is possible to divide or classify the contributions according to the type of meta-heuristic used, or depending on the type of membrane structure used. We have started with Ant Colony Optimization P Systems (ACOPS), as explained above, given the fact that, as far as we are concerned, few works addressed these algorithms, and so far only one type of membrane structure (in this case OLMS) was proposed, so we can explore several new variants to overcome some possible limitations and compare their performance. This decision was based on the low amount of papers in [18] studying ACOPS, in comparison with other MAs like GA or QIEA.

According to the promising works we can find in the state of art of ACOPS, previous to our contributions, it was proved it is efficient the use of a membrane

structure, taking advantage of the inherent parallelism and the movements of the different ants crossing the membranes to combine good results along the different iterations. This implied an improvement with respect to the classic ACO algorithm, and we considered it would be worth exploring them more in depth. Consequently, we are studying different variants of ACOPS [17], looking for desirable properties with respect to time or space complexity, along with other features related with the ease of use and design. Thus, on the one hand, we made changes in the form that ants communicate across the membranes, modifying the kind of colonies and kind of ants used. On the other hand, we started exploring changes in the hierarchical membrane structure used. To check the efficiency of this changes our processes generally are NP-complete problems like Travelling Salesman Problem (TSP) or Knapsack.

In summary, we chose a classic meta-heuristic with a clear intention on going from particular to general. We started with ACO, but not forgetting to gather, during this specific study, the common elements we might extract from this and other meta-heuristics and their hybrid variants in membrane computing. With a deeper exploration, we will be able to include new branches of Membrane Algorithms and progress with the formal model generalizing the different types of MA. At the same time, we have included ACOPS in P-Lingua framework, and prepared P-Lingua and MeCoSim to check the new variants developed against a solution of TSP. Once we have recently finished with the first simulator for ACOPS within P-Lingua, we have started conducting new studies about the influence of the different parameters in the results obtained with ACOPS, and then moving some steps forward with new structural and functional ingredients to improve the results currently published.

References

1. Eiben, A.E., Smith, J. *Introduction to Evolutionary Computing*, Springer-Verlag, 2003.
2. Dorigo, M., Stätzle, T. *Ant Colony Optimization*, Bradford Company, Scituate, MA, USA, 2004.
3. García-Quismondo, M., Gutierrez-Escudero, R., Pérez-Hurtado, I., Pérez-Jiménez, M.J., Riscos-Núñez, A. An overview of P-Lingua 2.0. *Lecture Notes in Computer Science*, 5957 (2010), 264–288.
4. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
5. Holland, J.H. Adaptation in Natural and Artificial Systems, *University of Michigan Press*, Ann Arbor, (1975).
6. Kennedy, J., Eberhart, R.C. *Swarm Intelligence*. Morgan Kaufman Publishers Inc., San Francisco, CA, USA, 2001.
7. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. Optimization by simulated annealing.*Science*, **220** (1983), 671–680.

8. Mathews, G.B. On the partition of numbers. *Proceedings of the London Mathematical Society*, **28** (1897), 486–490.
9. Menger, K. Das botenproblem. In Menger, K. (Ed.) *Ergebnisse eines Mathematischen Kolloquiums 2*, Teubner, Leipzig, (1932), 11-12.
Note: Previous works treated this problem since the 1800s (Hamilton, W.R., Kirkman,T.P., among others), previous to its general form by Menger.
10. Nishida, T.Y. Membrane Algorithms: Approximate Algorithms for NP-Complete Optimization Problems. In: Ciobanu G., Păun G., Pérez-Jiménez M.J. (eds) *Applications of Membrane Computing*. Natural Computing Series. Springer, Berlin, Heidelberg, 2006, 303-314.
11. Olsson, A. *Particle Swarm Optimization: Theory, Techniques and Applications, Engineering Tools, Techniques and Tables*. Nova Science Publishers, Incorporated, 2011.
12. Păun, Gh., Rozenberg, G., Salomaa, A. (eds.) *The Oxford Handbook of Membrane Computing*. Oxford University Press, New York, 2010.
13. Păun, Gh. Computing with membranes. *Turku Center for Computer Science*, TUCS Technical report 208 (1998), 1–42.
14. I. Pérez-Hurtado, L. Valencia-Cabrera, M.J. Pérez-Jiménez, M.A. Colomer, A. Riscos-Núñez, Mecosim: A general purpose software tool for simulating biological phenomena by means of P systems, in: Proceedings of International Conference on Bio-Inspired Computing: Theories and Applications, 2010, pp. 637–643.
15. K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach to Global Optimization (Natural Computing Series), Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.
16. G. Zhang, Quantum-inspired evolutionary algorithms: a survey and empirical study, *Journal of Heuristics*, **17**, 3 (2011), 303–351.
17. Zhang, G., Cheng, J., Gheorghe, M. A membrane-inspired approximate algorithm for traveling salesman problems. *Romanian Journal of Information Science and Technology*, **14** (2011), 3–19.
18. Zhang, G., Gheorghe, M., Pan L., Pérez-Jiménez M.J. Evolutionary membrane computing: A comprehensive survey and new results. *Information Sciences*, **279** (2014), 528-551.
19. Zhang, G., Liu, C.X., Rong, H.N. Analyzing radar emitter signals with membrane algorithms. *Mathematical and Computer Modelling*, **52** (2010), 1997–2010.

Laser Dynamics from a Membrane Computing Perspective

David Orellana-Martín[1], Luis Valencia-Cabrera[1], José L. Guisado[2], Francisco Jiménez-Morales[3], Agustín Riscos-Núñez[1], Mario J. Pérez-Jiménez[1]

[1]Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
{dorellana,lvalencia,ariscosn,marper}@us.es
[2]Department of Computer Architecture and Technology
jlguisado@us.es
[3]Department of Condensed Matter Physics
jimenez@us.es
Universidad de Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain

Summary. Modelling real systems and processes is a task classically performed through the approach of differential equation systems, defining the evolution of different variables as the different components of the system. A bad feature that differential equations have is that if some new parameter has to be introduced in the system, then often the previous equations are not useful, and the whole system has to be remodeled again. Bio-inspired computational models are abstractions of reality into a mathematical system that works with specific semantics and can perform some tasks, such as solving problems or demonstrating the universality of themselves or other models. An interesting application of these models is the modelling of real-life processes, where some of them as the so-called *P systems* have demonstrated previously that their performance is remarkable. This is not only for the similarity of the results with the experimental ones, but for its adaptability and modularity of the system, that is, if a new component of the real system is taken into account, not the whole system but a small part of it has to be changed in order to simulate the changed scenario. In this work, a first look at the dynamics of a laser physical system is given, reproducing the behavior of a first model with a PDP system.

Key words: Membrane Computing, Laser Dynamics, PDP system, Modelling.

1 Introduction

Lasers are devices that emit light through a process of optical amplification based on the stimulated emission of electromagnetic radiation. In fact, its name comes from "Light Amplification by Stimulated Emission of Radiation". In [21, 22], a

wide explanation of them can be found. While the principles of such devices were introduced by A. Einstein in [9], they were not applied until the 1950s. Nowadays, there exists a vast amount of applications based on lasers, for example, communications, optical memories and medicine, between other ones. For a good coverage of the history of laser early development, the reader should take a look at [14].

Cellular automata were discovered in the 1940s by Stanislaw Ulam and John von Neumann as a computational paradigm based on a regular grid of cells (that is why it is called *cellular* automata) in a given topology, each of them in one of a finite number of states. Classical models, like *Rule 110* [23] and *Conway's Game of Life* [11] use 1 and 2 dimensions, respectively. However, these devices are not restricted to 2 dimensions, but they can be defined by a grid of $n \in \mathbb{N}$ dimensions. In the beginning, Stephen Wolfram developed a systematic theoretical study of one-dimensional cellular automata, called *elementary* cellular automata. Later on, in [23] he claimed the usefulness of them not only as a theoretical model, but as a framework to simulate natural processes, applicable to cryptography and biology, as some patterns generated by them can be found in the nature [24], and several more applications. In [12, 13], a cellular automata model simulating the behavior of lasers is defined, giving some hints about how laser dynamics can be simulated by a computational simulation paradigm, and how good are they treating the topology of the real process.

Membrane Computing is a bio-inspired paradigm based on the structure and behavior of living cells. It was first introduced in [16], trying to give an alternative perspective to fields such as formal language theory and computability theory. The main devices within this framework are the so-called P systems. Several kinds of these systems have been defined, some of them are explained in [17, 18]. Apart from theoretical results, such as computational completeness and efficiency, a wide range of applications have been found by specific types of P systems. As some of them can be found in [10, 19, 26], we want to stress the impact of probabilistic-like systems, called *PDP systems*, in the field of ecosystems. From the first successful implementation of a model for the endangered species *Gypaetus barbatus*, or bearded vultures, in the Pyrenean and Prepyrenean mountains of Catalonia [6], passing through the Pyrenean chamois [4] and the zebra mussel [7] in the fluvial reservoir of Riba-roja, to the Giant Panda conservation in China [26], it has been proved that this framework is plausible for the simulation of real-life processes. In fact, in [1, 2], two simple models are defined to simulate two classical physics problems such as the Stern-Gerlach experiment and the Uranium 238 decay. Some tools have been developed in order to simulate and validate these models, such as `P-Lingua` [27], `MeCoSim` [28] as well as some GPU based simulators in the `PMCGPU` project [29]. In this work, we want to prove the usefulness of these kinds of systems in the simulation of the dynamics of lasers, as a different approach to the simulation of these devices. The paper is organized as follows: in the next sections, some definitions are given to make the work self-contained. Section 3 will be devoted to a brief explanation of laser dynamics. In Secion 4, cellular automata will be defined and a glimpse of the model of laser dynamics from [12, 13] will be given. Section 5

will introduce the framework of PDP systems and in the next one a first model to model laser dynamics within the framework of Membrane Computing will be explained. The work will be closed with some conclusions and open research lines.

2 Preliminaries

Some basic concepts are going to be introduced in this section. The reader is supposed to have a basic knowledge about alphabets, multisets, graphs and trees. For a broader definition, the reader is invited to take a look at [17, 18].

2.1 Lattices and topology

A *lattice* in $\mathbb{R}^n$ is a subgroup of the additive group $\mathbb{R}^n$ which is isomorphic to the additive group $\mathbb{Z}^n$, and which spans with the real vector space $\mathbb{R}^n$. Given a lattice Λ in an n-dimensional space, each element in Λ can be labelled by a natural number i. If Λ is a finite 2-dimensional square lattice, we can define the label of an element as $i = (x, y)$, being (x, y) the *position* of the element in the lattice, defined in a natural way. Given two elements of a lattice, we define the operator $+ : \mathcal{L}^2 \to \mathcal{L}$ as a sum of elements in the lattice into another one. In the previous example, $i + j = (x_i + x_j, y_i + y_j)$. The size of the lattice Λ is denoted by $|\Lambda|$.

The lattice can be a finite or an infinite set of elements. If the lattice is finite, its bounds can act as real bounds, that is, they can act as actual elements with static states, or they can be joint with other sides of the lattice, giving the lattice a different topology. Usually, 2-dimensional square cellular automata use cylinder or torus topology.

3 Laser dynamics

The basic components of a laser system [21, 20, 25] are a *laser medium*, where particles interact with each other, the *pumping process*, through which electrons get artificially excited into higher quantum-mechanical energy levels and *optical feedback elements*, that reflect repeatedly the radiation beam into the laser medium, building a resonant cavity in laser oscillators, which we will refer to generically as "laser devices" in this work, as in [12, 13].

In Figure 1, a description of these components can be visualized in a graphical way.

Lasers are based on the principle of *stimulated emission*, that is, an electron can decay to a lower energy state while it is stimulated by the presence of a photon, with the restriction that the latter has to have energy equal to the difference between the two energy levels, and as a result of this process, a new photon with the same wavelength, phase and polarization as the first one is emitted. In order to have a simplified model, only two energy levels for electrons, E_2 (upper level) and

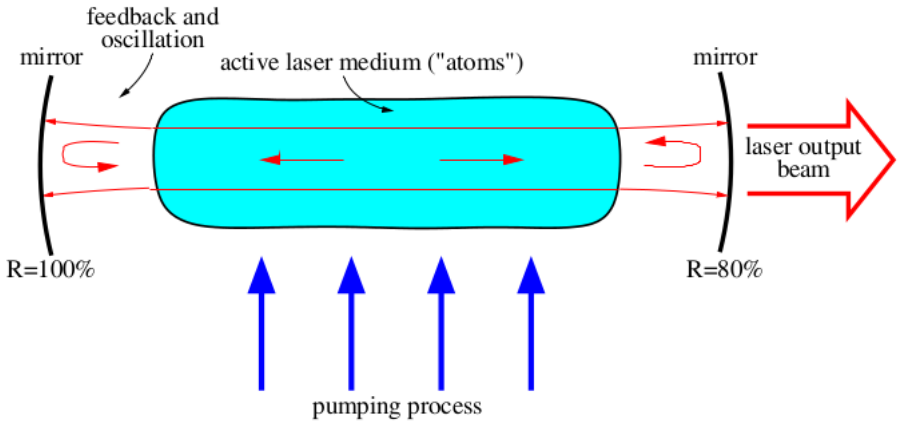


Fig. 1. Components and operation of a typical laser. R stands for reflectivity.

E_0 (lower level) are considered. The pass of an electron from the upper to the lower level is called *population inversion*. Under the assumption of this simplification, the dynamics of the laser can be modeled by the following coupled differential equations [21, 22]:

$$\frac{dn(t)}{dt} = KN(t)n(t) - \frac{n(t)}{\tau_c} \tag{1}$$

$$\frac{dN(t)}{dt} = R - \frac{N(t)}{\tau_a} - KN(t)n(t) \tag{2}$$

being:

- $n(t)$ the number of laser photons;
- $N(t)$ the population inversion (the number of electrons in the upper level);
- τ_c the decay time of photons in the cavity;
- τ_a the decay time of the upper laser level (E_2);
- R the pumping rate; and
- K the coupling constant.

First equation denotes the variation of laser photons through time, while Equation 2 give the temporal variation of the population inversion. The term $KN(t)n(t)$ accounts for the stimulated emission, $\frac{n(t)}{\tau_c}$ and $\frac{N(t)}{\tau_a}$ for the decay of photons and electrons in the upper level, respectively, and R introduces the pumping of electrons with a pumping rate R to the upper laser level E_2.

4 Cellular automata

Cellular automata are computational models defined informally as a grid of cells with states like *on* and *off* that will evolve through a set of rules.

4.1 Formal definition

Definition 1. *A cellular automaton can be defined as a tuple* $C = (\mathcal{L}, \mathcal{S}, S_0, \mathcal{N}, g)$ *where:*

- $\mathcal{L} \in \mathbb{R}^n$ *is a lattice;*
- $\mathcal{S}$ *is a finite set of states;*
- $S_0 \in \mathcal{S}^m$*, being* m *the number of cells of the automaton;*
- $\mathcal{N} = \{r_1, r_2, \ldots, r_\sigma\}$*,* $r_i \in \mathcal{L}$ *(*$1 \leq i \leq \sigma$*) and* σ *is the neighborhood size; and*
- $g : S^\sigma \to S$ *is a map from a set of states to a single one.*

A cellular automaton can be seen as a grid of m cells arranged with the cellular space given by $\mathcal{L}$ such that: (a) S_0 represents the initial state of the grid; (b) the neighborhood of the cell $i \in \mathcal{L}$ is given by the set of cells $\{r_1, r_2, \ldots, r_\sigma\}$, where r_i can depend on the cell i, even being itself.

A *configuration* at any instant of such kind of cellular automaton is described by the state of each cell at the moment. The initial configuration of $C = (\mathcal{L}, \mathcal{S}, S_0, \mathcal{N}, g)$ is $(s_1, s_2, \ldots, s_m)$, being $s_i \in \mathcal{S}$ $(1 \leq i \leq m)$.

A transition from a configuration $\mathcal{C}_t$ to another configuration $\mathcal{C}_{t+1}$ is obtained by applying g to all the cells in the cellular space $\mathcal{L}$. Thus, the state of each cell at any time step is determined by the state of the neighbouring cells at the previous time step. A *computation* of the system is a sequence of transitions starting from the initial configuration, where any term of the sequence other than the first, is obtained from the previous configuration in one transition step, and it is denoted by $\mathcal{C}_t \Rightarrow_C \mathcal{C}_{t+1}$.

4.2 Cellular automata model of laser dynamics

In [12, 13], a cellular automaton modelling the behavior of a laser is defined as a tuple $C = (\mathcal{L}, \mathcal{S}, S_0, \mathcal{N}, g)$, where:

- $\mathcal{L}$ is a 2-dimensional square lattice (a subset $\mathcal{L} \in \mathbb{Z}^2$) containing $N_c = L \times L$ sites, with periodic boundary conditions, that is, it follows the topology of a torus. Each site or element of the lattice is a cell, which is labelled by its position $\mathbf{r} = (i, j) \in \mathcal{L}$, where i and j are the row and the column indices, respectively;
- Four variables are associated at each cell, being $a_{\mathbf{r}}(t) \in \{0, 1\}$ and $c_{\mathbf{r}}(t) \in \{0, 1, \ldots, M\}$ being the state of the electron (being 1 when the electron is in the upper state) and the number of photons, respectively; and $\widetilde{a}_{\mathbf{r}}(t) \in \{0, 1, \ldots, \tau_a\}$ and $\widetilde{c}^k_{\mathbf{r}}(t) \in \{0, 1, \ldots, \tau_c\}$ the number of time steps that an electron has been in its upper state and that a photon $k \in \{1, 2, \ldots, M\}$ has been in a cell, respectively;
- In the initial configuration, all electrons are in their lower state and there are no photons in the system;

- $\mathcal{N}$ is the *Moore neighbourhood*, that is, given a cell $i = (x, y)$, $\mathcal{N} = \{(x-1, y-1), (x-1, y), (x-1, y+1), (x, y-1), (x, y), (x, y+1), (x+1, y-1), (x+1, y), (x+1, y+1)\}$
- g is a set of rules based on the value of the variables of each cell, which are not going to be reproduced here. For the definition of these rules, we invite the reader to view paper [13] or the thesis [12] for a wider explanation and meaning.

To get an idea, rules from g reproduce the different processes of the system, that is, *pumping*, *stimulated emission*, photon and electron decay and the introduction of noise photons. Three variables are taken into account when trying to reproduce the behavior of laser systems: the pumping rate, that makes an electron to promote to an upper level with probability λ, the electron decay time τ_a and the photon decay time τ_c.

First successful model works with $\lambda = 0.192$, $\tau_a = 30$ and $\tau_c = 10$, where λ is adimensional and τ_a and τ_c are measured in time steps. The evolution of the system simulates the behavior of a laser with such characteristics, as graphics of the Figure 3 demonstrate.

Some other results can be found in the reference, but only this example is going to be used for our purpose.

5 PDP systems

PDP systems are a variant of P systems inspired by the functioning of cells. Cells are able to run multiple processes in parallel in a perfectly synchronized manner, making them good candidates to be imitated for modeling complex problems. A PDP system can be viewed as a cellular tissue in which each cell is within a special compartment called environment. The cells have a particular structure hierarchy in which there is a skin membrane that defines and distinguishes the inside from the outside. In turn, inside a cell there are a number of hierarchically arranged membranes, where organelles or chemical substances capable of evolving according to specific reactions of the membrane may appear. PDP systems are probabilistic P systems, that is, the applications of their rules is commanded by a predefined probability on them. For a more exhaustive explanation of this model, see [5].

The key of software implementations of these systems are conflicts. If two or more rules compete for a resource, the algorithm has to take a strategy. The resolution of conflicts depends on the algorithm used to simulate the system. Some algorithms as the *Binomial Block Based simulation algorithm* (`BBB`) [3], the *Direct Non Deterministic distribution with Probabilities algorithm* (`DNDP`) [4] and the *Direct distribution based on Consistent Blocks Algorithm* (`DCBA`) [15] have been developed, each of them treating these conflicts in a different way. Thus, the state of the system at any time step is determined by the state of the system at the previous time step.

6 Laser dynamics modelling through a PDP system

In this first approach, to model the system, we use a PDP system of degree $(1,1)$ $\Pi = (\Sigma, G, R_E, \Gamma, R, T, \{f_r : r \in R_\Pi\}, M_{11})$, where:

- $\Gamma = \{a, z\} \cup \{p_i | 1 \leq i \leq \tau_c\} \cup \{e_i | 1 \leq i \leq \tau_a\}$, and $\Sigma = \emptyset$;
- $G = (V, E)$, where $V = \{e_1\}$ and $E = \emptyset$;
- $R_E = \emptyset$;
- R is the set of the following rules:

  6.1 Rules to simulate the *pumping process*:

  $[a]_1 \xrightarrow{\lambda} [e_1]_1$

  $[a]_1 \xrightarrow{1-\lambda} [a]_1$

  6.2 Rules to simulate the laser action:

  $$\left.\begin{array}{l} [e_i\, p_j\, p_k]_1 \xrightarrow{0.5} [a\, p_1\, p_{j+1}\, p_{k+1}]_1 \\ [e_i\, p_j\, p_k]_1 \xrightarrow{0.5} [e_{i+1}\, p_{j+1}\, p_{k+1}]_1 \end{array}\right\} \text{for } 1 \leq i \leq \tau_a, 1 \leq j \leq k \leq \tau_c$$

  6.3 Rules to reproduce the photon decay:

  $[p_i]_1 \longrightarrow [p_{i+1}]_1$ for $1 \leq i \leq \tau_c - 1$

  $[p_{\tau_c}]_1 \longrightarrow [z]_1$

  6.4 Rules to reproduce the electron decay:

  $[e_i]_1 \longrightarrow [e_{i+1}]_1$ for $1 \leq i \leq \tau_a - 1$

  $[e_{\tau_a}]_1 \longrightarrow [a]_1$

  6.5 Rules to introduce noise photons:

  $[z]_1 \xrightarrow{0.0003} [z\, p_1]_1$

  $[z]_1 \xrightarrow{0.9997} [z]_1$
- $T = 1000$; and
- $M_{11} = \{a^{160000}, z^{160000}\}$.

The inner probabilistic P system has a single membrane labelled by 1, so the initial configuration can be represented as in Figure 2.

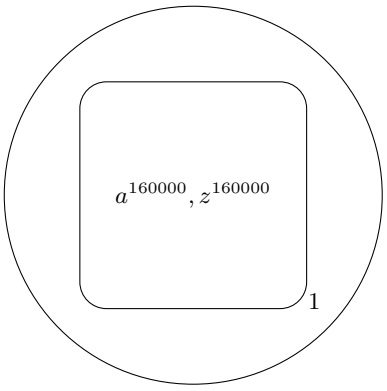


Fig. 2. Initial configuration of the PDP system Π.

Objects a represent electrons in the lower energy state E_0, while e_i represent electrons that have been i time steps in population inversion. Analogously, objects z represent photons that are not in the system and p_i represent photons that have been i time steps in the system.

Basically, the system works as said: electrons (respectively, photons) are excited (resp., introduced in the system) by rules from **6.1** (resp., **6.5**), as well as rules from **6.4** (resp., **6.3**) represent their evolution through time until reaching τ_a (resp., τ_c), while they become an object a (resp,. z) again. Rules from **6.2** simulate the interaction between electrons in population inversion and photons.

In the first case, we use the following constants: $\lambda = 0.192$, $\tau_c = 10$ and $\tau_a = 30$.

For the simulations, we have used the `DCBA algorithm` implemented in CUDA [29], using the cluster of the RGNC group [30] to increase their speed. The results seem pretty interesting since they reproduce almost exactly the obtained in the cellular automata approach and the experimental results. In Figure 4 it can be seen that the graphics obtained with this model are very similar with the ones of Figure 3.

7 Conclusions and future work

In this work, a new approach to the modelling of laser dynamics has been developed. In [12, 13], a cellular automaton modelling this process is defined, taking advantage their inherent parallelism to implement the algorithm in distributed systems. In this case, we use the parallelism of P systems and its implementation in CUDA to obtain similar results understanding the principles of the phenomenon. The use of probabilities seems reasonable since physical processes are usually probabilistic. In this first version, we only use a single membrane and a single environment to simulate the whole system, but we could try to use more membranes to simulate different parts of the laser, as well as to simulate the interaction of particles from the outside of the laser and so on. It can be used in a second version since the results with parameters $\lambda = 0.0125$, $\tau_c = 10$ and $\tau_a = 180$, the results are not similar to the experimental ones. Another interesting research line could be the use of *stochastic* P systems instead of *probabilistic* ones, since the equations defining the dynamics come with two constants K and R that could be introduced directly in the system.

Acknowledgements

This work was supported by project TIN2017-89842-P, cofinanced by Ministerio de Economa, Industria y Competitividad (MINECO) of Spain, through the Agencia Estatal de Investigacin (AEI), and by Fondo Europeo de Desarrollo Regional (FEDER) of the European Union.

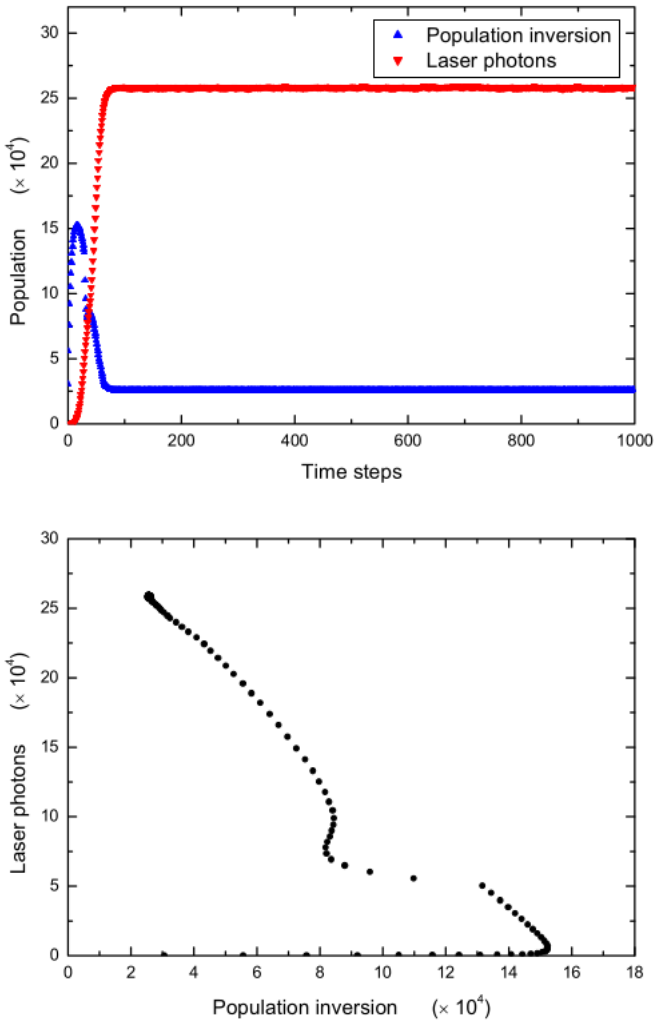


Fig. 3. Evolution of the cellular automaton for $\lambda = 0.192$, $\tau_c = 10$ and $\tau_a = 30$. Up: number of laser photons and population inversion versus time. Down: evolution in a phase space with the number of laser photons versus the population inversion.

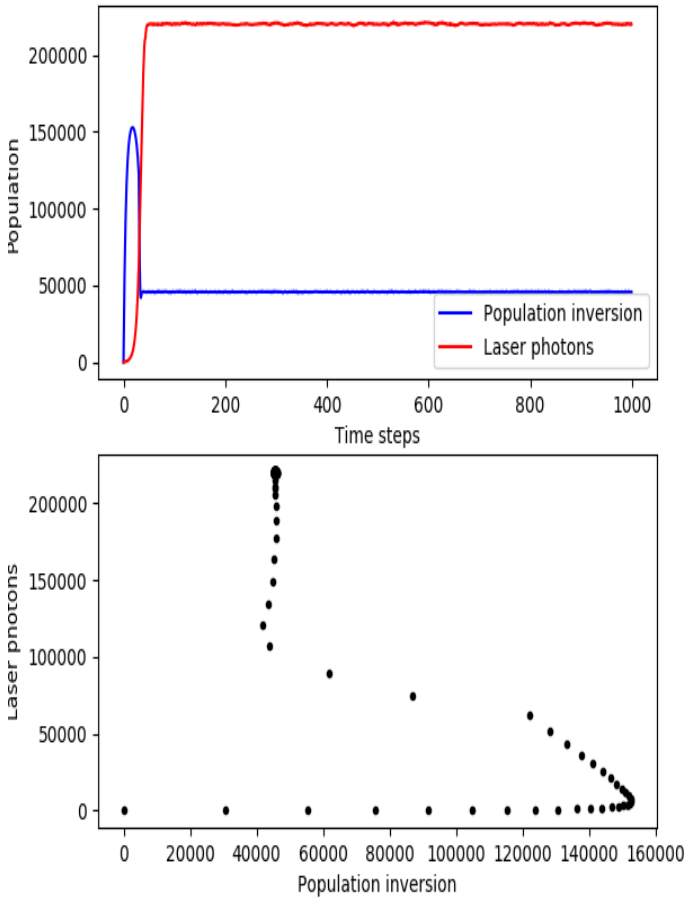


Fig. 4. Evolution of the PDP system for $\lambda = 0.192$, $\tau_c = 10$ and $\tau_a = 30$. Up: number of laser photons and population inversion versus time. Down: evolution in a phase space with the number of laser photons versus the population inversion.

References

1. M. Arazo, M. Barroso, O. De la Torre, L. Moreno, A. Ribes, P. Ribes, A. Ventura, D. Orellana-Martn. Stern-Gerlach Experiment. *Proceedings of the Fourteenth Brainstorming Week on Membrane Computing*, February 1 - 5, 2016, Sevilla, Spain, 101-112.
2. M. Arazo, M. Barroso, O. De la Torre, L. Moreno, A. Ribes, P. Ribes, A. Ventura, D. Orellana-Martn. Uranium-238 decay chain. *Proceedings of the Fourteenth Brainstorming Week on Membrane Computing*, February 1 - 5, 2016, Sevilla, Spain, 113-130.
3. M. Cardona, M.À. Colomer, A. Margalida, A. Palau, I. Pérez-Hurtado, M.J. Pérez-Jiménez, D. Sanuy. A computational modeling for real ecosystems based on P systems. *Natural Computing*, **10**, 1 (2011), 39-53.
4. M.À. Colomer, S. Lavn, I. Marco, A. Margalida, I. Pérez-Hurtado, M.J. Pérez-Jiménez, D. Sanuy, E. Serrano, L. Valencia-Cabrera. Modeling population growth of Pyrenean Chamois (Rupicapra p. pyrenaica) by using P systems. *Membrane Computing, 11th International Conference, CMC 2010, Jena, Germany, August 24-27, 2010, Revised Selected Papers*. Lecture Notes in Computer Science, 6501 (2011), 144-159. A preliminary version in *Proceedings of the Eleventh International Conference on Membrane Computing*, Jena, Germany, 24-27 August 2010, Verlag ProBusiness Berlin, 2010, ISBN 978-3-86805-721-8, pp. 121-135.
5. M.À. Colomer, A. Margalida, M.J. Pérez-Jiménez. Population Dynamics P System (PDP) Models: A Standardized Protocol for Describing and Applying Novel Bio-Inspired Computing Tools. *PLOS ONE*, **8** (4): e60698 (2013).
6. M.À. Colomer, A. Margalida, D. Sanuy, M.J. Pérez-Jiménez. A bio-inspired computing model as a new tool for modeling ecosystems: The avian scavengers as a case study. *Ecological modelling*, **222**, 1 (2011), 33-47.
7. M.À. Colomer, A. Margalida, L. Valencia, A. Palau. Application of a computational model for complex fluvial ecosystems: The population dynamics of zebra mussel Dreissena polymorpha as a case study. *Ecological Complexity*, **20** (2014), 116-126.
8. T.H. Cormen, C.E. Leiserson, R.L. Rivest. An Introduction to Algorithms. *The MIT Press*, Cambridge, Massachusetts, 1994.
9. A. Einstein. Strahlungs-Emission und -Absorption nach der quantentheorie. *Deutsche Physikalische Gesellschaft*, **18** (1916), 318323.
10. P. Frisco, M. Gheorghe, M. J. Pérez-Jiménez. Applications of Membrane Computing in Systems and Synthetic Biology. *Emergence, Complexity and Computation Series*, 2014.
11. M. Gardner. Mathematical Games The fantastic combinations of John Conway's new solitaire game "life". *Scientific American*, **223** (1970), 120123.
12. J.L. Guisado. Modelling and Simulation of Laser Dynamics with Cellular Automata on Parallel and Distributed Computers. PhD Thesis, Universidad de Sevilla, Sevilla, Spain, 2009.
13. J.L. Guisado, F. Jiménez-Morales, J.M. Guerra. Cellular automaton model for the simulation of laser dynamics. *Physical Review* **67** (2003), 066708.
14. J. Hecht. Beam, The Race to Make the Laser. *Oxford University Press*, Oxford, 2005.
15. M.Á. Martínez, I. Pérez-Hurtado, M. García-Quismondo, L.F. Macías-Ramos, L. Valencia-Cabrera, Á. Romero-Jiménez, C. Graciani, A. Riscos-Núñez, M.À. Colomer, M.J. Pérez-Jiménez. DCBA: Simulating population dynamics P systems with proportional objects distribution. *Membrane Computing, 13th International Conference,*

CMC 2012 Budapest, Hungary, August 28-31, 2012, Revised Selected Papers. Lecture Notes in Computer Science, 7762 (2013), 257-276. A preliminary version in *Proceedings of the 13th International Conference on Membrane Computing*, Budapest, Hungary, August 28-31, 2012, pp. 291-310.
16. Gh. Păun. Computing with membranes. *Journal of Computer and System Sciences*, **61**, 1 (2000), 108-143, and *Turku Center for CS-TUCS Report* No. 208, 1998.
17. Gh. Păun. Membrane Computing: An introduction. *Springer Natural Computing Series*, 2002.
18. Gh. Paun, G. Rozenberg, A. Salomaa, eds. The Oxford Handbook of Membrane Computing. *Oxford University Press*, Oxford, U.K., 2009.
19. L. Pan, Gh. Paun, M. J. Pérez-Jiménez, T. Song. Bio-inspired Computing: Theories and Applications. *Communications in Computer and Information Science Series*, 2014.
20. M. Sargent, M.O. Scully, W.E. Lamb. Laser Physics. *Addison-Wesley*, Reading, MA, 1974.
21. A. Siegman. Lasers. *Unversity Science Books*, Mill Valley, CA, 1986.
22. O. Svelto. Principles of lasers. *Plenum Press*, New York, 4th edition, 1998.
23. S. Wolfram. A new kind of science. *Wolfram Media*, Champaign, IL, 2002.
24. S. Wolfram. Cellular automata as models of complexity. *Nature*, 311 (1984), 419424.
25. A. Yariv. Quantum Electronics. *Wiley*, New York, USA, 3rd edition, 1989.
26. G. Zhang, M. J. Pérez-Jiménez, M. Gheorghe. Real-life applications with Membrane Computing. *Emergence, Complexity and Computation Series*, 2017.
27. `http://www.p-lingua.org/wiki/index.php/Main_Page`
28. `http://www.p-lingua.org/mecosim/`
29. `https://sourceforge.net/projects/pmcgpu/`
30. `http://www.gcn.us.es/gpucomputing`

MACPS (Modelling and Analysis of Cyber-Physical Systems)

Marian Gheorghe[1] and Florentin Ipate[2]

[1] School of Electrical Engineering and Computer Science
University of Bradford
Bradford BD7 1DP, UK
m.gheorghe@bradford.ac.uk
[2] Department of Computer Science
Faculty of Mathematics and Computer Science and ICUB
University of Bucharest
Str. Academiei 14, Bucharest 010014, Romania
florentin.ipate@ifsoft.ro

Summary. This introduces the MACPS (Modelling and Analysis of Cyber-Physical Systems) project, supported by the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-III-P4-ID-PCE-2016-0210. The site of the project is[3].

1 Introduction

"Cyber-Physical Systems (CPSs) are complex, multi-disciplinary, physically-aware next generation engineered systems that integrate embedded computing technology (cyber part) into the physical phenomena by using transformative research approaches" [9]. Although CPS concept is relatively new, coined initially around 2006, the system components are well-known and they are divided into the physical world, interfaces and the cyber system. The controller of the physical part and the theory behind control systems are the basis for all state-of-the-art continuous time dynamical systems and are largely used in designing CPS systems, thus have a crucial role in CPS design. The sensor network research and technologies related to sensor nodes and sensor networks - where a sensor node integrates sensors, actuators, computing elements (e.g. processor, memory, etc.) - are the most utilised networking design approaches for such systems.

The novelty comes from the high level of complexity and heterogeneity of such systems, the multitude of the physical parts with their sensors and actuators, the

[3] http://fmi.unibuc.ro/macps/

changing environment these components are interacting with, the ubiquitous network and communication aspects. Like many complex systems, CPS exhibit emergent behaviour, which result from (local) interactions among system components, with limited or constrained behaviour, and which might have faulty behaviour or cease to exist when removed from the system. Due to their ubiquity and impact on many aspects of our lives, one of the greatest challenges of future research is to efficiently establish the emergent behaviour of these systems [5]. CPS are safety- and time-critical systems facing various, sometimes unknown, situations, whereby correct and prompt answers to environment inputs and stimuli are expected.

The CPS analysis and design, although share many common features, there are significant differences with respect to methods, models and techniques utilised for specific systems. Techniques for modelling, specifying and analysing such systems exist; however, each such technique highlights certain features and disregards others to make the approach tractable. A new approach is necessary by "considering the cutting-edge technologies, necessary system-level requirements, and overall impact on the real world" [9]. In order to achieve this ambitious goal the approach should provide a sound theoretical basis, new specification methods, efficient and scalable analysis models and a combination of state-of-the-art tools. The benefits of the approach should be measurable and these will be proved by carefully selecting illustrative case studies from the areas mentioned above.

In MACPS [2] we look at two modelling paradigms that we use in the context of the CPS mentioned above: extended finite state machines and membrane (P) systems. The notion of state is at the heart of formal analysis, verification and model based testing. Most of the models built in this respect are based on finite state machines. However, in practice a formal (state-based) model of a system may not exist and, even when such a model has been built, it may have become obsolete due to the frequent changes in the specification or the unknown/uncertain behaviour that might appear or might need to incorporate uncertainties from its inception. One way to reverse engineer a finite state model of a system is through learning from queries. Learning finite state machines from queries was introduced by Angluin in [3]; the paper also provides a learning algorithm, called L*. In Angluin's setting, a learner asks queries and a teacher provides the answers. The L* algorithm infers a regular language, in the form of a deterministic finite automaton from the answers to a finite set of membership queries and equivalence queries. A membership query asks whether a certain input sequence is accepted by the system under test or not. In addition to membership queries, L* uses equivalence queries to check whether the learning algorithm is completed. The equivalence oracle provides counterexamples if the automaton constructed from the information available so far does not match the given language. Using equivalence queries in addition to membership queries, the L* algorithm can learn finite automata in polynomial time in its number of states.

However, an important limitation of the existing state-based learning techniques is that they consider only finite state machines (or finite automata). While finite state machines can successfully model the control aspects of the system, their

capability of modelling the system data is quite limited. On the other hand, extended finite state machines, such as X-machines, that combine the control aspects of the system with system data, exist and can be used to remove this limitation. An example of such extended finite state machine is the (communicating) X-machine [4], which provides a precise definition of both the individual and collective behaviour of agents by specifying their states, internal memory, functions (which represent the transitions between states) and communication protocols. The communicating X-machine is the basis of a very successful approach to agent-based modelling and of an agent-based simulation framework called FLAME (Flexible Large-scale Agent Modelling Environment) [1].

Another important advantage of X-machines is the existence of the associated test generation methods: these guarantee correctness properties of the implementation under test and clearly define constructive requirements, called "design for test" conditions, that a system must satisfy in order to be testable. By applying these methods, a complex software system is decomposed into a hierarchy of X-machines, designed in a top-down way and tested in a bottom-up way [10, 6].

One of the most natural ways of describing component-to-component interaction (inspired from biology but with wide applications) is provided by membrane computing [24,26]. Inspired by biochemical reactions across membranes in cells, membrane systems (also called P systems) are among the few existing approaches to combine quantitative, qualitative and topological system aspects with a flexible mechanism to capture the inherent parallelism of biochemical reactions. Membrane systems have been extended, e.g., to tissue systems or population systems, to capture features of multicellular organisms and cell populations [13].

In recent years, significant progress has been made in using P systems to model and simulate systems and problems from various areas. Although this model is primarily inspired from cell biology, the set of applications of this computational paradigm goes far beyond this class of systems, showing its adequacy for specifying highly complex systems [13]. In order to facilitate the modelling, in many cases various features have been added in an ad-hoc manner to these classes of P systems. This has led to a multitude of P systems variants, without a coherent integrating view. The newly introduced concept of kernel P systems (kP systems) [7] provides a response to this problem. A kP system integrates in a coherent and elegant manner many of the P system features most successfully used for modelling various applications and, consequently, it provides a framework for analysing these models. Furthermore, the expressive power of these systems has been illustrated by a number of representative case studies [7]. The kP system model is supported by a modelling language, called kP-Lingua, capable of mapping a kP system specification into a machine readable representation. Furthermore, kP systems are supported by a software framework, kPWorkbench [8], which integrates a set of related simulation and verification tools and techniques.

A different membrane system, called numerical P system [14], has been developed with the aim of abstracting from a distributed way of calculating polynomials values. It turned out that the model, slightly modified and called enzymatic P sys-

tems, and a software platform developed for them have been very effective in specifying and implementing controllers for mobile robots [12]. Recently the approach has been utilised for developing and implementing membrane controllers for trajectory tracking of some mobile robots [15]. However, these two approaches considering the formal verification of P systems and the use of this modelling paradigm for specifying mobile robot controllers have not been considered together. Another important aspect not considered in this context is that of synthesising such systems when uncertainties or even unknown elements might appear.

In conclusion, the two modelling approaches (state based formalisms, in particular X-machines, and P systems, in particular kP systems and enzymatic numerical P systems) have complementary capabilities and have proven their adequacy for specifying and analysing certain classes of relatively complex applications. X-machines have been successfully utilised for system specification, simulation and testing, relying on of very powerful methods and techniques, but these require fundamental developments in new learning approaches for state machines, with respect to unknown/uncertain scenarios, enhancing testing techniques that check that the system under test approximately conforms to the model and the verification scales up properly with the size of the CPS scenarios. While kP systems are supported by formal verification techniques and tools, these suffer from the state explosion problem, which can be alleviated by transforming these models into approximate state based models, using appropriate learning techniques and the specification capabilities of the enzymatic P systems have to be complemented by the verification approaches specific to kP systems.

2 Objectives

MACPS aims to develop a novel integrative approach to CPS specification, analysis and simulation. This consists of a three layer method: (a) a new theoretical basis for modelling and verification CPS applications that act and interact with uncertain and even unknown environment scenarios; (b) an integrated formal verification and testing approach for these systems; (c) a toolkit allowing the specification, analysis and simulation.

The theoretical basis of the research proposal envisages the development of new learning methods for state based systems and P systems; development of appropriate semantics (with spatial capabilities) for X-machines and P systems models; a unifying approach on kP systems and enzymatic P systems. The development of adequate formal verification and testing methods for CPS applications includes the integration of suitable verification methods and testing techniques for cyber-physical systems mentioned above, based on state based formalisms (in particular X-machines) and P systems (in particular kP systems and enzymatic numerical P systems). Learning techniques for X-machines and P systems are being developed that will remove or at least alleviate the state explosion problem associated with verifying complex systems, such as CPS. The X-machine and P systems models

produced will be then used in formal verification and testing. Formal verification capabilities will be provided through the development of a formal semantics. Existing testing methods for X-machines are adapted to the case when the implementation approximately conforms to the model under test. Furthermore, all modelling, verification and testing methods emerging from the project will be accompanied by powerful simulators and a toolkit that will integrate all these techniques.

In order to achieve these aims, a number of research objectives are followed:

- Development of approximate learning techniques for the X-machine formalism and synthesis algorithms for P systems. The X-machine approach follows the blueprint set out by Angluin [3] and is the subject of a paper to be submitted shortly to a highly regarded journal. The synthesis of P systems will expand the approached initiated in [16] by considering similar synthesis algorithms for kP systems and enzymatic P systems. In these cases the structure of the model is more complex than that considered in [16] and the approach will include uncertainties as well. These are challenging new features that the synthesis approach should tackle in an efficient and effective way.
- Definition of a formal semantics for X-machines and P systems. This will capture the spatio-temporal and dynamic aspects of these models. Using this logic, spatial and temporal properties can be expressed as queries for the model, e.g. that the system will eventually (temporal modality) reach a stable state in which all components of a certain kind are within a certain behaviour (spatial modality). Both temporal and spatial modalities have been investigated separately, but in this approach we will investigate the combination of these two and will study complexity aspects and efficient algorithms for parsing formal queries.
- Relationships between kP systems and enzymatic P systems. This research will reveal ways to efficiently convert enzymatic P systems into kP systems as well as new ways of defining P systems models including features of both these classes. Computational power and complexity aspects will be investigated and efficient algorithms to map them into kP systems.
- Development of test generation techniques for X-machines that check that the system approximately conforms to the X-machine model. These are extensions of existing techniques to the case when the conformance considered is not equivalence, but a (well-defined) approximation. As in the case of the learning algorithm, the test suite checks that the system under test behaves like the X-machine model only for the scenarios for which the approximation has been built and the result produced for sequences outside these scenarios is irrelevant.
- Integrative model checking approaches. We seek to extend the approach in [11], where various model checkers, both non-deterministic and probabilistic have been used to verify synthetic biology specification.
- Development of an integrated modelling, simulation, learning, verification and testing toolkit for CPS applications. This tool will integrate the above-mentioned components as well as related model checkers and testing tools. The tool will provide seamless communication amongst various components

and will handle the results in an intuitive manner. The tool will be used in several complex case studies of cyber-physical systems, from the automotive industry and robotics.

In order to achieve the ambitious aim of the project and its objectives, we will develop a completely new set of learning and synthesis methods for state machine and complex classes of P systems that include uncertain and unknown behaviour; investigate efficient translation algorithms of classes of P systems into kP systems and X-machines; develop a new formal semantics for the formalisms described; provide a multi-dimensional verification approach to such systems; provide a push-button set of tools and verify the approach for two significant CPS classes.

3 Impact

Our approach is new and with long term impact on CPS modelling, but also on various fields of computer science, such as formal verification, testing, formal specification, model learning and system synthesis, language engineering. On one hand, our approach employs a model learning approach, which is used to produce manageable models on which powerful verification algorithms can be applied. The idea has, so far, not been used for membrane systems or X-machines in particular, and CPS in general. On the other hand, formal verification through model checking techniques is complemented by testing techniques; these yield powerful test data, which can be regarded as benchmarks on which the system is validated. Besides the theoretical results produced, the project also has a strong engineering content, as it aims to produce an integrated toolkit for the modelling, verification and testing of CPS. Furthermore, the case studies used to guide our approach are not toy examples, but complex systems taken from already successful projects or newly defined systems, but of a similar size.

4 Team

The team working on this project consists of Prof. Florentin Ipate (University of Bucharest, principal investigator), Prof. Marian Gheorghe (University of Bradford), Prof. Felician Campean (University of Bradford), Prof. Alin Stefanescu (University of Bucharest), Dr. Sorina Predut (University of Bucharest), PhD student Ana Turlea (University of Bucharest, supervisor Prof. Florentin Ipate) and collaborators Prof. Gexiang Zhang (Southwest Jiaotong University, China) and Dr. Raluca Lefticaru (University of Bradford).

References

1. Flame web site. `http://flame.ac.uk/`. Accessed: 2018-03-05.

2. Macps web site. `http://fmi.unibuc.ro/macps/`. Accessed: 2018-05-14.
3. Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.
4. Tudor Balanescu, Anthony J. Cowling, Horia Georgescu, Marian Gheorghe, Mike Holcombe, and Cristina Vertan. Communicating stream X-machines systems are no more than X-machines. *J. UCS*, 5(9):494–507, 1999.
5. Ezio Bartocci, Oliver Höftberger, and Radu Grosu. Cyber-physical systems: Theoretical and practical challenges. *ERCIM News*, 2014(97), 2014.
6. K. Bogdanov, M. Holcombe, F. Ipate, L. Seed, and S. Vanak. Testing methods for X-machines: a review. *Form. Asp. Comput.*, 18(1):3–30, 2006.
7. Marian Gheorghe and Florentin Ipate. A kernel P systems survey. In *Membrane Computing - 14th International Conference, CMC 2013, Chişinău, Republic of Moldova, August 20-23, 2013, Revised Selected Papers*, pages 1–9, 2013.
8. Marian Gheorghe, Savas Konur, Florentin Ipate, Laurentiu Mierla, Mehmet E. Bakir, and Mike Stannett. An integrated model checking toolset for kernel P systems. In *Membrane Computing - 16th International Conference, CMC 2015, Valencia, Spain, August 17-21, 2015, Revised Selected Papers*, pages 153–170, 2015.
9. Volkan Gunes, Steffen Peter, Tony Givargis, and Frank Vahid. A survey on concepts, applications, and challenges in cyber-physical systems. *TIIS*, 8(12):4242–4268, 2014.
10. Mike Holcombe and Florentin Ipate. *Correct systems - building a business process solution*. Applied computing. Springer, 1998.
11. Savas Konur and Marian Gheorghe. A property-driven methodology for formal analysis of synthetic biology systems. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 12(2):360–371, 2015.
12. Ana Brândusa Pavel and Catalin Buiu. Using enzymatic numerical P systems for modeling mobile robot controllers. *Natural Computing*, 11(3):387–393, 2012.
13. Gheorghe Păun. Membrane computing. In *Handbook of Natural Computing*, pages 1355–1377. 2012.
14. Gheorghe Păun and Radu A. Păun. Membrane computing and economics: Numerical P systems. *Fundam. Inform.*, 73(1-2):213–227, 2006.
15. Xueyuan Wang, Gexiang Zhang, Ferrante Neri, Tao Jiang, Junbo Zhao, Marian Gheorghe, Florentin Ipate, and Raluca Lefticaru. Design and implementation of membrane controllers for trajectory tracking of nonholonomic wheeled mobile robots. *Integrated Computer-Aided Engineering*, 23(1):15–30, 2016.
16. Gexiang Zhang, Marian Gheorghe, Linqiang Pan, and Mario J. Pérez-Jiménez. Evolutionary membrane computing: A comprehensive survey and new results. *Inf. Sci.*, 279:528–551, 2014.

Calls for Participation to MC and Related Conferences/Meetings

19th International Conference on Membrane Computing (CMC19)

4 - 7 September, 2018, Dresden, Germany
`http://cmc19.uni-jena.de`

organised by the Friedrich-Schiller University Jena (Germany)
under the auspices of the
International Membrane Computing Society (IMCS)

AIM AND SCOPE

Membrane Computing is an area of computer science aiming to abstract computing ideas and models from the structure and the functioning of living cells, as well as from the way the cells are organized in tissues or higher order structures. It deals with membrane systems, also called P systems, which are distributed and parallel algebraic models processing multisets of objects in a localised manner (evolution rules and evolving objects are encapsulated into compartments delimited by membranes), with an essential role played by the communication among compartments and with the environment.

CMC19 aims at continuing the fruitful tradition of 18 previous editions enriched by some new ideas and inspirations emphasising multi disciplinarity and innovative capacity. The conference is intended to bring together researchers working in Membrane Computing and related areas in a friendly atmosphere enhancing communication and cooperation. We are pleased to hold CMC19 at the hotel venue NH Dresden-Neustadt, Hansastr. 43, D-01097 Dresden, Germany.

CMC19 receives financial support by Deutsche Forschungsgemeinschaft (German Research Foundation, grant HI 801/4-1).

Dresden is said to be one of the most beautiful and picturesque cities in Germany. The former Saxonian Residence and now the capital of the German free state Saxony with approximately 560,000 inhabitants belongs to the most visited in Germany with 4.3 million overnight stays per year. Along the Elbe river, the royal buildings are among the highly impressive buildings in Europe. Further high-

lights like the Frauenkirche (Church of our Lady), famous museums, fascinating exhibitions of art, exquisite vineyards, and many further cultural attractions complete the long list of outstanding sights worth to be visited in Dresden and its environment.

You are cordially welcome to CMC19 in Dresden, Germany.

INVITED AND KEYNOTE SPEAKERS

- Gheorghe Paun (Romanian Academy, Bucharest, Romania): Membrane Computing after 20 Years
- Erzsébet Csuhaj-Varjú (Eötv"os Loránd University, Budapest, Hungary): Keynote: How Membrane Computing Influences Theoretical Computer Science
- Alberto Leporati (University Milan-Bicocca, Italy): Time and Space Complexity of P Systems – And Why They Matter
- Jörg Teichmann (Bergmannstrost Hospital Associated with Martin-Luther University, Halle, Germany): Keynote: Data Mining using Membrane Computing: Examining Pseudonymous
  Healthy State and Medical Data from More Than 600,000 Patients Covering a German County
- Thomas Preusser (Accemic Technologies GmbH Dresden, Germany): A Brute-Force Solution to the 27-Queens Puzzle Using Distributed Computation
- Stefan Schuster (Friedrich Schiller University Jena, Germany): Algebra Meets Biology

SESSIONS

We plan to have six dedicated conference sessions in which new and original results from accepted regular and late papers will be presented:

– Membrane-based Algorithms
– New Variants of P Systems
– Theoretical Insights
– Spiking Neural P Systems
– Applications of Membrane Computing
– Late-breaking Results

PROCEEDINGS

The proceedings volume containing 21 contributions will be available during the conference. The post-conference proceedings, a volume in Springer Lecture Notes in Computer Science series (LNCS), for selected and additionally refereed papers,

will be published after CMC19. In addition, a selection of best papers is planned to be published within the new Springer Journal of Membrane Computing.

Springer will sponsor the Best Student Paper Award (EUR 500.00).

DATES AND DEADLINES

Submission deadline: ..... May 07, 2018
Notification of acceptance: ..... June 07, 2018
Early bird registration: ..... June 11, 2018
Camera-ready version: ..... July 02, 2018
Registration deadline: ..... July 02, 2018
Conference: ..... September 04-07, 2018

REGISTRATION

Conform the tradition, CMC19 is planned as a friendly working meeting with a good balance of science, tourism, socializing, and scientific collaboration. The basic CMC19 registration fee of EUR 330 (for early bird registration until 11th June 2018, and EUR 370 afterwards) covers conference participation, printed proceedings volume, welcome bag, conference material, lunches during the conference, and coffee break refreshments. Additional supplementary fees will be made for participation in the half-day tourist program (EUR 60) and for the CMC19 dinner (EUR 50). Registration can be done online via the conference webpage.

ACCOMMODATION AT THE CONFERENCE HOTEL

The NH Hotel Dresden-Neustadt offers a special rate to CMC19 participants for accommodation including breakfast during the conference and for the weekend before and after the conference (EUR 74.10 per person and night plus EUR 5.00 city tax per person and night, special rate available for bookings before July 21, 2018). Please note that car parking at the hotel ground might cause an additional fee. The conference webpage provides a direct link for online booking.

TRAVELLING INFORMATION

We recommend to travel to Dresden by airplane. The Dresden International Airport is connected via Frankfurt, Munich, and Dusseldorf with most of the European airports and with many airports worldwide. For participants from Czech Republic, it is more convenient to reach Dresden by an international train (Eurocity) via Prague.

Directly from the Dresden airport terminal (underground level), the city train (S-Bahn) departs every 30 minutes from the early morning to midnight. A single ticket available at the station and valid for a city trip costs around EUR 2.40. To reach the conference hotel, we recommend to leave the city train at the station Dresden-Neustadt. The trip takes approximately 15 minutes. From there, it is a 15 minutes walk to the NH Hotel Dresden-Neustadt. To do so, please use the northern exit to Hansastrasse, a broad avenue with a median green strip and pedestrian paths at the left and right sides. Follow the street northwards for around 950 m up to the crossroad with Fritz-Reuter-Strasse where the NH Hotel Dresden-Neustadt is situated. Alternatively, you can hire a taxi either from the airport (approximately EUR 35) or from the station Dresden-Neustadt (approximately EUR 10).

PROGRAM COMMITTEE

Sergey Verlan (Paris, France)
Claudio Zandron (Milan, Italy) – co-chair
Gexiang Zhang (Chengdu, China)

STEERING COMMITTEE

Henry Adorna (Quezon City, Philippines)
Artiom Alhazov (Chisinau, Moldova)
Bogdan Aman (Iasi, Romania)
Matteo Cavaliere (Manchester, UK)
Erzsebet Csuhaj-Varju (Budapest, Hungary)
Giuditta Franco (Verona, Italy)
Rudolf Freund (Vienna, Austria)
Marian Gheorghe (Bradford, UK) - Honorary member
Thomas Hinze (Jena, Germany)
Florentin Ipate (Bucharest, Romania)
Shankara N. Krishna (Bombay, India)
Alberto Leporati (Milan, Italy)
Taishin Y. Nishida (Toyama, Japan)
Linqiang Pan (Wuhan, China)
Gheorghe Paun (Bucharest, Romania) - Honorary member
Mario J. Perez-Jimenez (Sevilla, Spain)
Agustin Riscos-Nunez (Sevilla, Spain)
Jose M. Sempere (Valencia, Spain)
Petr Sosik (Opava, Czech Republic)
Kumbakonam G. Subramanian (Chennai, India)
Gyorgy Vaszil (Debrecen, Hungary)
Sergey Verlan (Paris, France)
Claudio Zandron (Milan, Italy)
Gexiang Zhang (Chengdu, China)

ORGANIZING COMMITTEE

Joern Behre (Jena, Germany)
Thomas Hinze (Jena and Dresden, Germany)

CONTACT INFORMATION

Thomas Hinze
Friedrich-Schiller University Jena
Department of Bioinformatics
Ernst-Abbe-Platz 2
D-07743 Jena, Germany

Asian Branch of International Conference on Membrane Computing (ACMC2018) 10-16 December, Auckland–New Zealand

Website: 2018.asiancmc.org

Call for papers

ACMC 2018, the Asian Branch of International Conference on Membrane Computing, is one of the flagship conferences on Membrane Computing, aiming to provide a high-level international forum for researchers working in membrane computing and related areas, especially for the ones from the Asian region. This conference is the seventh edition of ACMC with the six editions having successfully taken place in Wuhan (China, 2012), Chengdu (China, 2013), Coimbatore (South India, 2014), Anhui (China, 2015), Bangi (Malaysia, 2016), Chengdu (China, 2017) and also a geographical expansion of ICMC (International Conference on Membrane Computing). The other branch, European Branch of ICMC (ECMC), which is held every year from the year of 2000 in different European Countries.

Like artificial neural networks, evolutionary algorithms, swarm intelligence, cellular automata and DNA computing, membrane computing is also a branch of natural computing or nature-inspired computing and was initiated by Gheorghe Păun in 1998. It aims to abstract computing models, called membrane systems or P systems, from the structure and the functioning of the living cell as well as from the cooperation of cells in tissues, organs, and populations of cells. This research area has grown into a vigorous scientific discipline and has attracted a large number of researchers all over the world.

ACMC 2018 is planned as a friendly interactive conference with several introductory tutorials, several keynote lectures and some specialized sessions, which will cover a wide range of topics on membrane computing, including theory, applications, implementation and various aspects related to membrane computing. The registration fee will be of about xxxx NZD per person. This fee covers the participation, the special dinner, the tourist programme, as well as the lunches. The accommodation, to be paid by each participant, will be available at various prices. Various accommodation possibilities will be posted soon on the conference page.

Topics

The Seventh Asian Branch of International Conference on Membrane Computing (ACMC2018) provides an open platform to bring together scholars worldwide to present their recent work on membrane computing. The topics of this conference are as follows (but not limited to):

(1) Theoretical aspects of membrane computing

- Various variants of computing models: cell-, tissue- and neural-like P systems.
- Computing power of membrane computing models.
- Computing efficiency of membrane computing models.

(2) Applications of membrane computing

- Robots controller design.
- Modeling using P systems for biological systems, ecological systems, etc.
- Membrane-inspired optimization algorithms for various problems.
- Fault diagnosis of various systems, such as robots, power systems, etc.
- Other applications.

(3) Implementation of membrane computing models

- Software implementation.
- Hardware implementation.
- Biological implementation.
- Other implementation.

(4) Related topics

- Bio-Inspired Optimization Techniques, such as optimization algorithms inspired by cells or DNA computing
- Other topics related to membrane computing or dealing with cellular computing, DNA computing, etc.

Awards

ACMC 2018 sets up two best papers awards, BEST PAPER AWARD and BEST STUDENT PAPER AWARD, which aims to promote the academics, encourage young scientists to participate in academic activities, further to improve the paper quality and expand conference influence.

1. Best Paper Award Regulations:

(1) Eligibility

The considered papers must satisfy:

(i) The paper is accepted by the ACMC 2018;

(ii) One of the authors must register;

(iii) One of the authors must present the paper at the conference.

(2) Requirement

The paper should present a significant contribution with regard to theoretical results, applications or implementation of P systems, or comprehensive and high-level surveys on a specific topic, and should be written in a professional way.

2. Best Student Paper Award Regulations

(1) Eligibility

The considered papers must satisfy:

(i) The paper is accepted by the ACMC 2018;

(ii) The first author must be an undergraduate, master or PhD student.

(iii) The student must finish the registration process and present the paper at the conference.

(2) Requirement

The paper should present a significant contribution with regard to theoretical results, applications or implementation of P systems, or comprehensive and high-level surveys on a specific topic, and should be written in a professional way.

3. Procedure

The best paper and best student paper are selected by a Technical Committee through evaluating the reviewing reports, quality and presentation.

4. Award

The best papers elected by these regulations are awarded:

(1) A certificate of "ACMC 2018 Best Paper Award" or "ACMC 2018 Best Student Paper Award". The certificate is signed by the conference chair;

(2) Prize or gifts.

Invited Speakers: to be announced

Publications

Papers accepted for presentation will appear in conference pre-proceedings of ACMC 2018. After the conference, all the accepted papers will be selected to be considered for publication in the following reputed publications:

(1) A selection of the ACMC2018 accepted papers will be re-reviewed and published in the Springer journal "Journal of Membrane Computing (JMC)";

(2) A selection of the ACMC2018 accepted papers will be re-reviewed and published in the SCI-indexed journal "International Journal of Unconventional Computing (IJUC)";

(3) A selection of the ACMC2018 accepted papers will be combined with the BIC-TA2018 papers to be re-reviewed and published in Communications in Computer and Information Science (CCIS) (EI-indexed volume), Springer;

(4) A selection of the ACMC2018 accepted papers will be combined with the ECMC2018 papers to be re-reviewed and published in Lecture Notes in Computer Science (LNCS) (EI-indexed volume), Springer;

(5) A selection of the ACMC2018 accepted papers will be re-reviewed and published in the EI-indexed international journals.

Important Dates

Submission deadline: **Sunday September 16, 2018**
Acceptance notification: Sunday October 21, 2018
Camera-ready version and early registration: Sunday November 18, 2018

Submission

Authors are invited to submit their original research contributions (including significant work in progress) on membrane computing, its applications and related subjects. Papers (of reasonable length) should be formatted according to Lecture Notes in Computer Science (LNCS) format (please refer to **ftp://ftp.springer.de/pub/tex/latex/llncs/latex2e/llncs2e.zip**). All papers should be submitted as PDF files through EasyChair conference system website. The submission Web page for ACMC 2018 is **https://easychair.org/conferences/?conf=acmc2018**.

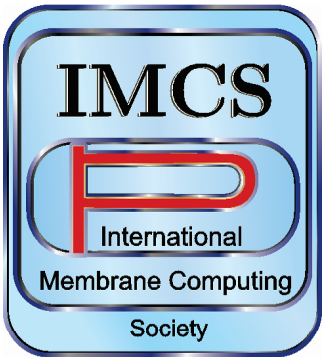
If there is any difficulty or problem, please do not hesitate to contact the organizer by email: zhgxdylan@126.com.

Host institutions

The University of Auckland, Auckland, New Zealand

Sponsors

International Membrane Computing Society (IMCS)



The University of Auckland

Committees

Steering Committee:

Henry Adorna (Quezon City, Philippines)
Artiom Alhazov (Chisinau, Moldova)
Bogdan Aman (Iasi, Romania)
Matteo Cavaliere (Edinburgh, Scotland)
Erzsébet Csuhaj-Varjú (Budapest, Hungary)
Giuditta Franco (Verona, Italy)
Rudolf Freund (Wien, Austria)
Marian Gheorghe (Bradford, UK) - Honorary member
Thomas Hinze (Cottbus, Germany)
Florentin Ipate (Bucharest, Romania)
Shankara N. Krishna (Bombay, India)
Alberto Leporati (Milan, Italy)
Taishin Y. Nishida (Toyama, Japan)
Linqiang Pan (Wuhan, China) – Co-Chair
Gheorghe Păun (Bucharest, Romania) - Honorary member
Mario J. Pérez-Jiménez (Sevilla, Spain)
Agustín Riscos-Núñez (Sevilla, Spain)
José M. Sempere (Valencia, Spain)
Petr Sosík (Opava, Czech Republic)
Kumbakonam Govindarajan Subramanian (Chennan, India)
György Vaszil (Debrecen, Hungary)
Sergey Verlan (Paris, France)
Claudio Zandron (Milan, Italy) – Co-Chair
Gexiang Zhang (Chengdu, China)

Program Committee:

Henry Adorna (Quezon City, Philippines)
Cătălin Buiu (Bucharest, Romania)
Matteo Cavaliere (Edinburgh, Scotland)
Erzsébet Csuhaj-Varjú (Budapest, Hungary)
Xiaoju Dong (Shanghai, China)
Marian Gheorghe (Bradford, UK)
Ping Guo (Chongqi, China)
Thomas Hinze (Cottbus, Germany)
Florentin Ipate (Bucharest, Romania)
Alberto Leporati (Milan, Italy)
Xiyu Liu (Jinan, China)
Taishin Nishida (Toyama, Japan)
Tseren-Onolt Ishdorj (Mongolia)
Jose M. Sempere (Valencia, Spain)
Ravie Chandren Muniyandi (Bangi, Malaysia)
Radu Nicolescu (Auckland, Australian)

Linqiang Pan (Wuhan, China) – Chair
Gheorghe Păun (Bucharest, Romania)
Hong Peng (Chengdu, China)
Mario J. Pérez-Jiménez (Sevilla, Spain)
Agustín Riscos-Núñez (Sevilla, Spain)
Tao Song (Qingdao, China)
Kumbakonam Govindarajan Subramanian (Chennan, India)
D.G. Thomas (Chennai, India)
Sergey Verlan (Paris, France)
Jun Wang (Chengdu, China)
Jianhua Xiao (Tianjing, China)
Hsu-Chun Yen (Taiwai, China)
Claudio Zandron (Milan, Italy)
Xiangxiang Zeng (Xiamen University)
Gexiang Zhang (Chengdu, China) – Co-Chair
Xingyi Zhang (Anhui, China)
Xue Zhang (Boston, USA)
Xuncai Zhang (Zhengzhou, China)

Organizing Committee:

Michael J. Dinneen (Auckland, New Zealand)
Radu Nicolescu (Auckland, New Zealand)

Contact

Dr. Radu Nicolescu
Department of Computer Science
University of Auckland
Private Bag 92019
Auckland 1142
New Zealand
E-mail: `r.nicolescu@auckland.ac.nz`

Reports on MC Conferences/Meetings

Report about the 16th Brainstorming Week on Membrane Computing

The Sixteenth Brainstorming Week on Membrane Computing (BWMC) was held in Sevilla, Spain, from January 30 to February 2, 2018, hosted by the Research Group on Natural Computing (RGNC) from the Department of Computer Science and Artificial Intelligence of Universidad de Sevilla.

As the previous editions, the BWMC was a cooperation-based conference, focused on the interaction among the participants. Mornings were devoted to "provocative" sessions presenting ideas, open problems and new research lines. After the lunch break, joint work sessions were scheduled to let the attendants meet each other interests and try to get new results based both on previous open problems or problems proposed in the conference itself. The program is available at `www.gcn.us.es/16bwmc_program`.

Rudolf Freund announced the two upcoming conferences UCNC 2018 and MCU 2018, both to be held in Fontainebleau, in June 2018. Also Agustín Riscos-Núñez presented the BICAS session at HPCS 2018 to be held in Orléans in July 2018.

After each BWMC, a proceedings volume and one or two special issues of various international journals were published. Here is their list:

- BWMC 2003: *Natural Computing* – volume 2, number 3, 2003, and *New Generation Computing* – volume 22, number 4, 2004;
- BWMC 2004: *Journal of Universal Computer Science* – volume 10, number 5, 2004, and *Soft Computing* – volume 9, number 5, 2005;
- BWMC 2005: *International Journal of Foundations of Computer Science* – volume 17, number 1, 2006;
- BWMC 2006: *Theoretical Computer Science* – volume 372, numbers 2-3, 2007;
- BWMC 2007: *International Journal of Unconventional Computing* – volume 5, number 5, 2009;
- BWMC 2008: *Fundamenta Informaticae* – volume 87, number 1, 2008;
- BWMC 2009: *International Journal of Computers, Control and Communication* – volume 4, number 3, 2009;

- BWMC 2010: *Romanian Journal of Information Science and Technology* – volume 13, number 2, 2010;
- BWMC 2011: *International Journal of Natural Computing Research* – volume 2, numbers 2-3, 2011;
- BWMC 2012: *International Journal of Computer Mathematics* – volume 99, number 4, 2013;
- BWMC 2013: *Romanian Journal of Information Science and Technology* – volume 17, number 1, 2014;
- BWMC 2014: *Fundamenta Informaticae* – volume 134, numbers 1-2, 2014;
- BWMC 2015: *Natural Computing* – volume 15, issue 4, 2016 (some papers from ACMC 2015 were also selected for this special issue);
- BWMC 2016: *Theoretical Computer Science* – in press, corrected proof (some papers from ACMC 2016 were also selected for this special issue);
- BWMC 2017: *Theoretical Computer Science* – to appear.

***

The titles and abstracts of the papers that will appear in the proceedings of this BWMC are listed below.

Introducing the Concept of Activation and Blocking of Rules in the General Framework for Regulated Rewriting in Sequential Grammars

Artiom Alhazov, Rudolf Freund, Sergiu Ivanov

Abstract. We introduce new possibilities to control the application of rules based on the preceding application of rules which can be defined for a general model of sequential grammars and we show some similarities to other control mechanisms as graph-controlled grammars and matrix grammars with and without applicability checking as well as grammars with random context conditions and ordered grammars. Using both activation and blocking of rules, in the string and in the multiset case we can show computational completeness of context-free grammars equipped with the control mechanism of activation and blocking of rules even when using only two nonterminal symbols.

One-Membrane P Systems with Activation and Blocking of Rules

Artiom Alhazov, Rudolf Freund, Sergiu Ivanov

Abstract. We introduce new possibilities to control the application of rules based on the preceding applications, which can be defined in a general way for (hierarchical) P systems and the main known derivation modes. Computational completeness can be obtained even for one-membrane P systems with non-cooperative rules and using both activation and blocking of rules, especially for the set modes of derivation.

When we allow the application of rules to influence the application of rules in previous derivation steps, applying a non-conservative semantics for what we consider to be a derivation step, we can even "go beyond Turing".

Input-Driven Tissue P Automata

Artiom Alhazov, Rudolf Freund, Sergiu Ivanov, Marion Oswald, Sergey Verlan

Abstract. We introduce several variants of input-driven tissue P automata where the rules to be applied only depend on the input symbol. Both strings and multisets are considered as input objects; the strings are either read from an input tape or defined by the sequence of symbols taken in, and the multisets are given in an input cell at the beginning of a computation, enclosed in a vesicle. Additional symbols generated during a computation are stored in this vesicle, too. An input is accepted when the vesicle reaches a final cell and it is empty. The computational power of some variants of input-driven tissue P automata is illustrated by examples and compared with the power of the input-driven variants of other automata as register machines and counter automata.

A Note on a New Class of APCol Systems

Lucie Ciencialová, Erzsébet Csuhaj-Varjú, György Vaszil

Abstract. We introduce a new acceptance mode for APCol systems (Automaton-like P colonies), variants of P colonies where the environment of the agents is given

by a string and during functioning the agents change their own states and process the string similarly to automata. In case of the standard variant, the string is accepted if it can be reduced to the empty word. In this paper, we define APCol systems where the agents verify their environment, a model resembling multihead finite automata. In this case, a string of length n is accepted if during every halting computation the length of the environmental string in the configurations does not change and in the course of the computation every agent applies a rule to a symbol on position i of some of the environmental strings for every i, $1 \leq i \leq n$ at least once. We show that these verifying APCol systems simulate one-way multihead finite automata.

P Colony Automata with LL(k)-like Conditions

Erzsébet Csuhaj-Varjú, Kristóf Kántor, György Vaszil

Abstract. We investigate the possibility of the deterministic parsing (that is, parsing without backtracking) of languages characterized by (generalized) P colony automata. We define a class of P colony automata satisfying a property which resembles the LL(k) property of context-free grammars, and study the possibility of parsing the characterized languages using a k symbol lookahead, as in the LL(k) parsing method for context-free languages.

Testing Identifiable Kernel P Systems Using an X-machine Approach

Marian Gheorghe, Florentin Ipate, Raluca Lefticaru, Ana Ţurlea

Abstract. This paper presents a testing approach for kernel P systems (*kP systems*), based on the X-machine testing framework and the concept of cover automaton. The testing methodology ensures that the implementation conforms the specifications, under certain conditions, such as the *identifiably* concept in the context of kernel P systems.

Spiking Neural P Systems with Addition/Subtraction Computing on Synapses

Yun Jiang, Zhiqiang Chen

Abstract. Spiking neural P systems (SN P systems, for short) are a class of distributed and parallel computing models inspired from biological spiking neurons. In this paper, we introduce a variant called SN P systems with addition/subtraction computing on synapses (CSSN P systems). CSSN P systems are inspired and motivated by the shunting inhibition of biological synapses, while incorporating ideas from dynamic graphs and networks. We consider addition and subtraction operations on synapses, and prove that CSSN P systems are computationally universal as number generators, under a normal form (i.e. a simplifying set of restrictions).

Characterizing PSPACE with Shallow Non-Confluent P Systems

Alberto Leporati, Luca Manzoni, Giancarlo Mauri, Antonio E. Porreca, Claudio Zandron

Abstract. In P systems with active membranes, the question of understanding the power of non-confluence within a polynomial time bound is still an open problem. It is known that, for shallow P systems, that is, with only one level of nesting, non-confluence allows them to solve conjecturally harder problems than confluent P systems, thus reaching **PSPACE**. Here we show that **PSPACE** is not only a bound, but actually an exact characterization. Therefore, the power endowed by non-confluence to shallow P systems is equal to the power gained by *confluent* P systems when non-elementary membrane division and polynomial depth are allowed, thus suggesting a connection between the roles of non-confluence and nesting depth.

Limits on P systems with Proteins and Without Division

David Orellana-Martín, Luis Valencia-Cabrera, Agustín Riscos-Núñez, Mario J. Pérez-Jiménez

Abstract. In the field of Membrane Computing, computational complexity theory has been widely studied trying to find frontiers of efficiency by means of syntactic

or semantical ingredients. The objective of this is to find two kinds of systems, one non-efficient and another one, at least, *presumably* efficient, that is, that can solve **NP**-complete problems in polynomial time, and adapt a solution of such a problem in the former. If it is possible, then **P** = **NP**. Several borderlines have been defined, and new characterizations of different types of membrane systems have been published.

In this work, a certain type of P system, where proteins act as a supporting element for a rule to be fired, is studied. In particular, while division rules, the abstraction of cellular *mitosis* is forbidden, it seems that only problems from class **P** can be solved, in contrast to the result obtained allowing them.

Narrowing Frontiers of Efficiency with Evolutional Communication Rules and Cell Separation

David Orellana-Martín, Luis Valencia-Cabrera, Bosheng Song, Linqiang Pan, Mario J. Pérez-Jiménez

Abstract. In the framework of *Membrane Computing*, several efficient solutions to computationally hard problems have been given. To find new borderlines between families of P systems that can solve them and the ones that cannot is an important task to tackle the **P** versus **NP** problem. Adding syntactic and/or semantic ingredients can mean passing from non-efficiency to *presumably* efficiency. Here, we try to get narrow frontiers, setting the stage to adapt efficient solutions from a family of P systems to another one. In order to do that, a solution to the `SAT` problem is given by means of a family of tissue P systems with evolutional symport/antiport rules and cell separation with the restriction that both the left-hand side and the right-hand side of the rules have at most two objects.

***

The list of participants as well as their email addresses are given below with the aim of facilitating further communication and interaction:

1. José A. Andreu-Guzmán, Universidad de Sevilla (Spain)
   `andreuguzman36@gmail.com`
2. Péter Battyányi, University of Debrecen (Hungary)
   `battyanyi.peter@zimbra.inf.unideb.hu`
3. Sandra M. Gómez-Canaval, Universidad Politécnica de Madrid (Spain)
   `sgomez@etsisi.upm.es`
4. Lucie Ciencialová, Silesian University in Opava (Czech Republic)
   `lucie.ciencialova@fpf.slu.cz`

5. Erzsébet Csuhaj-Varjú, Eötvös Loránd University (Hungary)
csuhaj@inf.elte.hu
6. Andrés Doncel-Ramírez, Universidad de Sevilla (Spain)
andrestp96@gmail.com
7. Rudolf Freund, Technological University of Vienna (Austria)
rudi@emcc.at
8. Carmen Graciani, Universidad de Sevilla (Spain)
cgdiaz@us.es
9. Miguel Á. Gutiérrez-Naranjo, Universidad de Sevilla (Spain)
magutier@us.es
10. Florentin Ipate, University of Bucharest (Romania)
florentin.ipate@ifsoft.ro
11. Sergiu Ivanov, IBISC, Université Évry, Université Paris-Saclay (France)
sergiu.ivanov@univ-evry.fr
12. Kristóf Kántor, University of Decebren (Hungary)
rapid90x@gmail.com
13. Raluca Lefticaru, University of Bradford (United Kingdom)
R.Lefticaru@bradford.ac.uk
14. Alberto Leporati, University of Milan-Bicocca (Italy)
leporati@disco.unimib.it
15. Luca Manzoni, University of Milan-Bicocca (Italy)
luca.manzoni@disco.unimib.it
16. Miguel Á. Martínez-del-Amor, Universidad de Sevilla (Spain)
mdelamor@us.es
17. David Orellana-Martín, Universidad de Sevilla (Spain)
dorellana@us.es
18. Gheorghe Păun, Universidad de Sevilla (Spain) and Academia Europaea
curteadelaarges@gmail.com, gpaun@us.es
19. Ignacio Pérez-Hurtado, Universidad de Sevilla (Spain)
perezh@us.es
20. Mario de J. Pérez-Jiménez, Universidad de Sevilla (Spain) and Academia Europaea
marper@us.es
21. Antonio E. Porreca, University of Milan-Bicocca (Italy)
porreca@disco.unimib.it
22. Agustín Riscos-Núñez, Universidad de Sevilla (Spain)
ariscosn@us.es
23. Daniel Rodríguez-Chavarría, Universidad de Sevilla (Spain)
danrodcha@gmail.com
24. Álvaro Romero-Jiménez, Universidad de Sevilla (Spain)
romero.alvaro@us.es
25. Zeyi Shang, University Southwest Jiaotong (China)
zeyi.shang@lacl.fr

26. Ana Turlea, University of Bucharest (Romania)
t_anacris@yahoo.com
27. Luis Valencia-Cabrera, Universidad de Sevilla (Spain)
lvalencia@us.es
28. György Vaszil, University of Decebren (Hungary)
vaszil.gyorgy@inf.unideb.hu
29. Jiang Yun, Chongqing University of Technology and Business (China)
jiangyun@email.ctbu.edu.cn

The meeting was partially supported by Universidad de Sevilla, more precisely by Department of Computer Science and Artificial Intelligence, *VI Plan Propio, Vicerrectorado de Investigación de la Universidad de Sevilla* and *Plan Andaluz de Investigación, Desarrollo e Innovación (PAIDI) 2020*.

Research Group on Natural Computing
Universidad de Sevilla
(March 2018)

A Summary of 2018 China Workshop on Membrane Computing (CWMC 2018)

Xiangxiang Zeng[1], Xiangrong Liu[1], Gexiang Zhang[2], Linqiang Pan[3]

[1] Department of Computer Science,
Xiamen University, Xiamen 361005, Fujian, China
[2] School of Electrical Engineering,
Southwest Jiaotong University, Chengdu 610031, China
[3] 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

The 2018 China Workshop on Membrane Computing (CWMC 2018) was held in Xiamen from 19-22 April 2018, hosted by Xiamen University, China. The aims of this meeting are to provide a platform for Chinese researchers in the community of membrane computing and related areas, and also to inspire some ideas for preparing collaborative papers to be submitted to The Asian Branch of International Conference on Membrane Computing (ACMC 2018) and Journal of Membrane Computing. The workshop focused on recent results and open problems from Chinese community of membrane computing from researchers in China, and several invited talks were given. About 40 participants from more than 10 universities participated in the workshop.

At the opening ceremony, Prof. Gexiang Zhang, the President of International Society of Membrane Computing (IMCS), reported the recent advancement of IMCS including IMCS Bulletin, several Festschrift volumes or special issues dedicate to big names of our community, for their 70th birthday, and Call-For-Papers of ACMC 2018. He distributed 40 hard copies of Bulletin (Issue 4) to the participants. He also introduced the *Journal of Membrane Computing*, on behalf of Prof. Linqiang Pan, Editor-in-Chief.

A very short description on the invited talks is as follows:

Invited Talk 1 - Some Remarks on the Research of Membrane Computing
Hong Peng
Professor, Xihua University, China
E-mail: ph.xhu@hotmail.com

In reference, membrane computing is described in two forms: (1) membrane computing is a class of distributed parallel computing systems, inspired from ; (2) membrane computing is a class of distributed parallel computing models, inspired from . The two descriptions may indicate two research lines: (1) computing systems and (2) computing models. According to the two research lines, the talk review their research contexts and further topics, respectively.

1. membrane computing as computing systems The talk briefly discusses some remarks for the following topics: system construction, theoretical computing problems, implementation and direct applications.
2. membrane computing as computing models In the research line, membrane computing is viewed as a class of modeling models and tools, which can used as modeling, representation and validation tools for real-world problems. The talk describes some remarks of its topics, including model extension, state equation, structural properties and indirect applications.

Invited Talk 2- Open Problems About Spiking Neural P Systems with Communication on Request

Tingfang Wu
Ph.D student, Huazhong University of Science & Technology
E-mail: tfwu@hust.edu.cn

Spiking neural P system with communication on request (abbreviated as SNQ P system) is a type of spiking neural P system, which is inspired by both the communication between neurons through spikes and the communication pattern by request in parallel-cooperating grammar systems. In such a system, the spikes are transmitted from one neuron to another one only when the receiving neuron makes a query to request spikes from the sending neuron; in other words, the receiving neuron has the initiative for communication. At the beginning the motivation of SNQ P systems is introduced. Then, the formal definition of SNQ P systems is reviewed. Moreover, some theoretical results about the universality of SNQ P systems are analyzed in detail. Finally, several research topics about SNQ P systems are pointed out:

1. Consider an interesting kind of queries the form $(a^{\infty-s}, j)$: request all but s spikes a from neuron σ_j.
2. Consider other semantics of rule application in SNQ P systems, e.g., the local check mode.
3. Can the request-response communication pattern improve the application potential of SNQ P systems?

Ten short talks listed below:

Presentation 1 – Optimization Spiking Neural P System, Jianping Dong, Southwest Jiaotong University, Chengdu

Presentation 2 – Optimization of Low Carbon Smart Logistics System Driven by Big Data, Jianhua Xiao, Nankai University, Tianjin

Presentation 3 – Application of Membrane Computing in Image Processing, Jianying Yuan, Chengdu University of Information Technology, Chengdu

Fig. 1. Group photo from China Workshop on Membrane Computing.

Presentation 4 – Tissue-like P systems with evolutional symport/antiport rules, Bosheng Song, Huazhong University of Science and Technology, Wuhan

Presentation 5 – Spiking Neural P Systems with Computing on Synapses, Yun Jiang, Chongqing Technology and Business University, Chongqing

Presentation 6 – Membrane Controller and P System Hardware Implementation, Zeyi Shang, Southwest Jiaotong University, Chengdu

Presentation 7 – Application of Membrane Computing in Microgrid Energy Management, Tao Wang, Xihua University, Chengdu

Presentation 8 – Spiking Neural P Systems with Multiple Channels, Xiaoxiao Song, Xihua University, Chengdu

Presentation 9 – Membrane Computing Simulation System, Changsheng Quan, Chongqing University, Chongqing

Presentation 10 – Homeostasis Tissue P Systems with Object Evolutional Rules, Yueguo Luo, Yangtze Normal University, Chongqing

Contents of Previous Volumes

June 2016 Volume

December 2016 Volume

June 2017 Volume

December 2017 Volume