

## Volume 1 Issue 1

Pan, L., Păun, Gh., Zhang, G. (2019). **Foreword: Starting JMC.** *Journal of Membrane Computing*, 1(1), 1-2. <https://doi.org/10.1007/s41965-019-00010-5>

Bottoni, P., Labella, A., & Rozenberg, G. (2019). **Reaction systems with influence on environment.** *Journal of Membrane Computing*, 1(1), 3–19. <https://doi.org/10.1007/s41965-018-00005-8>

Reaction systems, motivated by the functioning of the living cell, became a novel model of interactive computation. In this paper, we pursue this line of research. More specifically, we present a systematic investigation of possible interactions of a reaction system with its environment (context). While in the original definition this interaction is one-way, i.e., the behavior of a reaction system is influenced by its environment, we investigate now also the influence of the system on its environment, where a possible time delay of this influence is also considered. To understand the behavior of reaction systems when such a two-way interaction takes place, we establish its relationship to their context-independent behavior (i.e., the behavior which is not influenced by the environment).

Mayne, R., Phillips, N., & Adamatzky, A. (2019). **Towards experimental P-systems using multivesicular liposomes.** *Journal of Membrane Computing*, 1(1), 20–28. <https://doi.org/10.1007/s41965-018-00006-7>

P-systems are abstract computational models inspired by the phospholipid bilayer membranes generated by biological cells. Illustrated here is a mechanism by which recursive liposome structures (multivesicular liposomes) may be experimentally produced through electroformation of dipalmitoylphosphatidylcholine films for use in 'real' P-systems. We first present the electroformation protocol and microscopic characterisation of incident liposomes towards estimating the size of computing elements, level of internal compartment recursion, fault tolerance and stability. Following, we demonstrate multiple routes towards embedding symbols, namely modification of swelling solutions, passive diffusion, and microinjection. Finally, we discuss how computing devices based on P-systems can be produced and their current limitations.

Orellana-Martín, D., Valencia-Cabrera, L., Riscos-Núñez, A., & Pérez-Jiménez, M. J. (2019). **P systems with proteins: a new frontier when membrane division disappears.** *Journal of Membrane Computing*, 1(1), 29–39. <https://doi.org/10.1007/s41965-018-00003-w>

P systems with active membranes are usually defined as devices hierarchically structured that evolve through rewriting rules. These rules take the inspiration on the chemical reactions that happen within a cell and the role of both the inner and the plasma membranes as a “filter”, letting components pass or not. Classically, these systems are non-cooperative, that is, the left-hand side of the rules has at most one object. Using polarizations, dissolution or cooperation, these systems have been proved to have enough power to efficiently solve computationally hard problems, obtaining new complexity frontiers with respect to their non-cooperative counterparts. In this paper, division rules are interchanged by separation rules. While the first ones produce two new membranes and two new objects, duplicating the objects within the original one, separation rules distribute the objects of the original membrane into the two new created membranes, so no new objects are created in this way. To obtain new objects, a rule of the type  $[ a \rightarrow a^2 ]$  would be needed to accomplish that feature that seems to be necessary to obtain efficient solutions to NP-complete problems. Here, we present the limits when using separation rules instead of division rules.

Sánchez-Karhunen, E., & Valencia-Cabrera, L. (2019). **Modelling complex market interactions using PDP systems.** *Journal of Membrane Computing*, 1(1), 40–51. <https://doi.org/10.1007/s41965-019-00008-z>

Last decade has witnessed an increasing effort on modelling and simulation of phenomena within a wide range of areas such as Biochemistry, Ecology, Robotics or Engineering by using membrane computing, providing solutions for relevant problems (signalling pathways, population dynamics, robot control or fault diagnosis, among others). However, for no apparent reasons, other areas have not been investigated to such extent. This is the case of computational economics, where Gh. and R. Păun explored the so-called producer–retailer problem and, in a foundational paper, proposed an initial model making use of membrane

computing **modelling tools**. In the present paper, we design a solution based on **population dynamics** P systems for an **enriched version** of that problem. This enhanced model, closer to reality, takes into account several economic issues not considered in the initial model, including: **depreciation of production capacity**, **decision mechanism to increase manufacturing capability**, **dividends** payment and **costs associated to production** factors. Additionally, the model has been simulated making use of the framework provided by **P-Lingua and MeCoSim**, and delivering a **custom application** based on them to reproduce the virtual experiments. Finally, several scenarios have been **analysed** focusing on different elements included in the model.

Román, G. (2019). **Inference of bounded L systems with polymorphic P systems**. *Journal of Membrane Computing*, 1(1), 52–57. <https://doi.org/10.1007/s41965-019-00007-0>

In this paper, we are going to solve the **inference** problem of **bounded L** systems, namely such L systems which work on **filaments having length** up to a fixed size. We will show that these bounded L systems have considerable computational power as they can **simulate linear-bounded automata**. To carry out the inference, we are going to construct a specific **polymorphic P** system with **target** indication, which can **reproduce the transitions** of the examined bounded L system, and which is of size  $O(n|G|^4)$ , where G is the alphabet of the bounded L system with n as the maximal size of the filaments.

Díaz-Pernil, D., Gutiérrez-Naranjo, M. A., & Peng, H. (2019). **Membrane computing and image processing: a short survey**. *Journal of Membrane Computing*, 1(1), 58–73.

<https://doi.org/10.1007/s41965-018-00002-x>

Membrane computing is a well-known research area in computer science inspired by the organization and behavior of live cells and tissues. Their computational devices, called P systems, work in parallel and distributed mode and the information is encoded by multisets in a localized manner. All these features make P systems appropriate for dealing with **digital images**. In this paper, some of the **open research lines** in the area are presented, focusing on **segmentation** problems, **skeletonization** and **algebraic-topological aspects** of the images. An extensive **bibliography about the application** of membrane computing to the study of digital images is also provided.

## Volume 1 Issue 2

Leporati, A., Manzoni, L., Mauri, G., Porreca, A. E., & Zandron, C. (2019). **Characterizing PSPACE with shallow non-confluent P systems**. *Journal of Membrane Computing*, 1(2), 75–84.

<https://doi.org/10.1007/s41965-019-00011-4>

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.

Orellana-Martín, D., Valencia-Cabrera, L., Riscos-Núñez, A., & Pérez-Jiménez, M. J. (2019). **Minimal cooperation as a way to achieve the efficiency in cell-like membrane systems**. *Journal of Membrane Computing*, 1(2), 85–92. <https://doi.org/10.1007/s41965-018-00004-9>

**Cooperation** is doubtless a relevant ingredient on **rewriting** rules based computing models. This paper provides an overview on both **classical and newest results** studying how cooperation among objects influences the ability of cell-like membrane systems to solve computationally hard problems in an efficient way. In this paper, two types of such membrane systems will be considered: (a) **polarizationless** P systems with active membranes **without dissolution** rules when **minimal cooperation** is permitted in object evolution rules; and (b) cell-like P systems with **symport/antiport** rules of **minimal** length. Specifically, assuming that **P** is not equal to **NP**, several **frontiers of the efficiency** are obtained in these two computing frameworks, in such manner that each borderline provides a tool to tackle the **P versus NP** problem.

Pérez-Hurtado, I., Orellana-Martín, D., Zhang, G., & Pérez-Jiménez, M. J. (2019). **P-Lingua in two steps: flexibility and efficiency**. *Journal of Membrane Computing*, 1(2), 93–102.

<https://doi.org/10.1007/s41965-019-00014-1>

Membrane computing is a bio-inspired computing paradigm that lacks in vivo implementation. That is why **software or hardware implementations** have to be used to **validate models**. Several **tools** have been created for this purpose; some of them are created for **specific purposes**, such as solving a computationally hard problem; and others are more **generic**, to cover a broad spectrum of possible models. The former have the advantage of being very **efficient, crucial for solving large** instances of certain problems; however, this efficiency leads to a **loss of generality**, since algorithms are usually **hard-coded** and they do **not allow other models**. On the contrary, the latter are perfect **tools for researchers**, given that new **models can be checked** without much **effort by defining them in the framework**; since these algorithms have to simulate as **many models** as possible, they lack **specificities to improve the performance**. P-Lingua has been widely used to **simulate membrane systems**, having integrated both **a language and a simulator**. To obtain better results in terms of **time used to simulate** models defined in this language, a **new perspective** is studied. The model defined in P-Lingua will be **compiled into C++ source code** that will **implement an ad hoc simulator**. This code will consider **specifications about how rules have to be executed**, that is, some **simple specifications of the semantics**. To show how it works, some **examples of specifications of models** will be presented, which can be simulated using the new-developed GNU GPLv3 **command-line tool** *pcc*.

Nash, A., & Kalvala, S. (2019). **A P system model of swarming and aggregation in a Myxobacterial colony**. *Journal of Membrane Computing*, 1(2), 103–111.

<https://doi.org/10.1007/s41965-019-00015-0>

**Bacterial communities** provide an interesting subject for the study of **emergence** and complexity as the consequence of many **local interactions**. In particular, the **soil-dwelling social bacterium Myxobacteria** demonstrates two distinct types of motility, **social motility** via the **sensing of bacterial slime** deposits and **adventurous motility**. Both modes of motility are governed by local interactions. Using P systems, a membrane computing methodology based on compartmental rewrite rules for modelling computational processes; this work demonstrates how **minimal set of rules** can model **swarming and aggregating behaviour** in **Myxobacteria** bacterial populations. Our model uses a **multi-environment P system** similar a 2D **cellular automaton** to represent the **substrate environment** whilst **stochastic rule selection** dictates Myxobacterial motion according to behaviour observed **in vitro**. The rules account for both mechanisms of motility, the **deposit and detection of slime**, a **change in direction** due to **C-signal induction** and the **mixing of population numbers**. Simulations demonstrate an extensible computational framework for the modelling of bacterial behaviour, with the potential for extension into **additional emergent behaviours**.

Cooper, J., & Nicolescu, R. (2019). **Alternative representations of P systems solutions to the graph colouring problem**. *Journal of Membrane Computing*, 1(2), 112–126.

<https://doi.org/10.1007/s41965-019-00013-2>

This paper first presents a **simulation of the simple kernel P systems** solution to the **graph 3-colouring** problem presented in a previous paper by Gheorghe et al., implemented in a programming style named **Concurrent ML**, which is based on the concept of **synchronous communication** between logical processing elements. This paper then presents and informally analyses an **alternative compact single-cell solution** to the same problem using P systems with **compound objects (cP systems)**, which has the benefit of **naturally adapting to the use of any number of colours** greater than zero—only the specified colour symbols need to be changed. **Successful and failing examples** of the latter solution are also presented.

Mitrana, V. (2019). **Polarization: a new communication protocol in networks of bio-inspired processors**. *Journal of Membrane Computing*, 1(2), 127–143. <https://doi.org/10.1007/s41965-018-0001-9>

This work is a survey of the most recent results regarding the computational **power of the networks of bio-inspired processors** whose communication is based on a new protocol called **polarization**. In the former models, the communication amongst processors is based on **filters** defined by some **random-context** conditions, namely the **presence** of some symbols and the **absence** of other symbols. In the new protocol discussed here, a polarization (negative, neutral, and positive) is associated with each **node**, while the **polarization of data** navigating through the network is **computed in a dynamical way** by means of a

valuation function. Consequently, the protocol of communication amongst processors is naturally based on the compatibility between their polarization and the polarization of the data. We consider here three types of bio-inspired processors: evolutionary processors, splicing processors, and multiset processors. A quantitative generalization of polarization (evaluation sets) is also presented. We recall results regarding the computational power of these networks considered as accepting devices. Furthermore, a solution to an intractable problem, namely the 0 / 1 Knapsack problem, based on the networks of splicing processors with evaluation sets considered as problem solving devices, is also recalled. Finally, we discuss some open problems and possible directions for further research in this area.

### Volume 1 Issue 3

Jimenez, Z. B., Cabarle, F. G. C., de la Cruz, R. T. A., Buño, K. C., Adorna, H. N., Hernandez, N. H. S., & Zeng, X. (2019). **Matrix representation and simulation algorithm of spiking neural P systems with structural plasticity.** *Journal of Membrane Computing*, 1(3), 145–160.

<https://doi.org/10.1007/s41965-019-00020-3>

In this paper, we create a matrix representation for spiking neural P systems with structural plasticity (SNPSP, for short), taking inspiration from existing algorithms and representations for related variants. Using our matrix representation, we provide a simulation algorithm for SNPSP systems. We prove that the algorithm correctly simulates an SNPSP system: our representation and algorithm are able to capture the syntax and semantics of SNPSP systems, e.g. plasticity rules, dynamism in the synapse set. Analyses of the time and space complexity of our algorithm show that its implementation can benefit using parallel computers. Our representation and simulation algorithm can be useful when implementing SNPSP systems and related variants with a dynamic topology, in software or hardware.

Cruz, R. T. A. D. L., Cabarle, F. G., & Adorna, H. N. (2019). **Generating context-free languages using spiking neural P systems with structural plasticity.** *Journal of Membrane Computing*, 1(3), 161–177. <https://doi.org/10.1007/s41965-019-00021-2>

Spiking neural P system (SNP system) is a model of computation inspired by networks of spiking neurons. An SNP system is a network of neurons that can send an object, known as a spike, to each other. Spiking neural P system with structural plasticity (SNPSP system) is a variant of the classical SNP system. SNPSP system that incorporates the ideas of synaptogenesis (creating new synapses) and synaptic pruning (deletion of existing synapses), collectively known as structural plasticity, as features of the model. This gives SNPSP systems the ability to change their own structure/topology. In this work, we use SNPSP systems to generate context-free languages. We create a procedure for constructing an SNPSP system given a context-free grammar in Greibach normal form (GNF). The resulting SNPSP system essentially simulates the way in which a context-free grammar in GNF is used to generate languages. We use modules known as arithmetic-memory modules, also created using SNPSP systems, to perform arithmetic operations which are needed for the simulation.

Cencialová, L., Csuhaĵ-Varjú, E., Cenciala, L., & Sosík, P. (2019). **P colonies.** *Journal of Membrane Computing*, 1(3), 178–197. <https://doi.org/10.1007/s41965-019-00019-w>

P colonies are abstract computing devices modelling communities of very simple reactive agents living and acting in a joint shared environment. The concept was motivated by so-called colonies, grammar systems based on interplay of very simple agents, on one hand, and by membrane systems, massively parallel computational models inspired by cell biology, on the other hand. Some variants of P colonies also allow the environment to participate actively in the system's evolution. In this paper we summarize the most important results on P colonies, present open problems concerning these constructs, and suggest new research directions in their study.

Sosík, P. (2019). **P systems attacking hard problems beyond NP: a survey.** *Journal of Membrane Computing*, 1(3), 198–208. <https://doi.org/10.1007/s41965-019-00017-y>

In the field of membrane computing, a great attention is traditionally paid to the results demonstrating a theoretical possibility to solve NP-complete problems in polynomial time by means of various models of P

systems. A bit **less common** is the fact that almost all models of P systems with this capability are actually **stronger**: some of them are able to solve **PSPACE-complete** problems in polynomial time, while others have been recently shown to characterize the class **P<sup>#P</sup>** (which is conjectured to be strictly included in **PSPACE**). A large part of these results has appeared quite **recently**. In this **survey**, we focus on **strong models** of membrane systems which have the potential to solve hard problems belonging to classes containing NP. These include P systems with **active membranes**, P systems with **proteins on membranes** and **tissue** P systems, as well as P systems with **symport/antiport and membrane division** and, finally, **spiking** neural P systems. We provide a survey of **computational complexity** results of these membrane models, pointing out some **features** providing them with their computational strength. We also mention a sequence of **open** problems related to these topics.

Valencia-Cabrera, L., Orellana-Martín, D., Martínez-del-Amor, M. Á., & Pérez-Jiménez, M. J. (2019). **An interactive timeline of simulators in membrane computing**. *Journal of Membrane Computing*, 1(3), 209–222. <https://doi.org/10.1007/s41965-019-00016-z>

As with any fast-emerging research front in computer science, the proliferation of theoretical and practical results within Membrane computing since its appearance in 1998 was astonishing. As a consequence, it became necessary during the subsequent years to produce several surveys collecting the main achievements from a theoretical point of view, along with some specific **surveys about simulation tools** for this paradigm. As the discipline has reached a certain degree of maturity, more **practical applications** have arisen, and new collective works are summarising the new **software products** appeared. However, while these recapitulation efforts remain useful for details about new simulators, they cannot act as **exhaustive updated listings**, as they become obsolete as soon as new tools are developed. Thus, we considered that it was necessary to provide an **interactive tool** showing an **updated timeline** (<https://www.gcn.us.es/SimulationMC>) about the simulation of the computational devices of membrane computing (a.k.a P systems), aiming to stay updated whenever any new practical work comes out in the discipline. This paper recalls the main stages and milestones within the **evolution of simulation tools** for different **types and variants of P** systems, along with their main related applications. In addition, it describes the interactive **web tool** with the timeline mentioned, where all the references related here have been incorporated. Unlike other survey papers, it is the intent of this work to reinforce this initial collective effort with the web endpoint kept alive and updated.

Manca, V. (2019). **Metabolic computing**. *Journal of Membrane Computing*, 1(3), 223–232. <https://doi.org/10.1007/s41965-019-00012-3>

The paper reviews some aspects of **MP grammars**, a particular type of P systems (M stands for Metabolic) consisting of multiset rewriting rules, which were introduced in the context of Membrane Computing, for **modeling biological dynamics**. The main **features of MP theory** are recalled, such as the **control mechanisms** based on **regulation functions**, **MP graphs**, representation of **oscillatory dynamics**, **regression algorithms**, and MP modeling. Finally, the computational **universality** of MP grammars is proved by means of Minsky's register machines.

## Volume 1 Issue 4

Aman, B., & Ciobanu, G. (2019). **Synchronization of rules in membrane computing**. *Journal of Membrane Computing*, 1(4), 233–240. <https://doi.org/10.1007/s41965-019-00022-1>

We modify the most used evolution strategy in membrane systems (namely that of maximal parallelism) by imposing a **synchronization between rules**. A synchronization over a set of rules can be applied **only if each rule of the set can be applied** at least once. For membrane systems working in the **accepting** mode, this synchronization is powerful enough to provide the **computational completeness** without any other ingredient (no catalysts, promoters, inhibitors, etc). The **modeling power** of synchronization is described by **simulating the basic arithmetic** operations (addition, subtraction, multiplication and division).

Turlea, A., Gheorghe, M., Ipate, F., & Konur, S. (2019). **Search - based testing in membrane computing**. *Journal of Membrane Computing*, 1(4), 241–250. <https://doi.org/10.1007/s41965-019-00027-w>

Search-based testing is widely used for generating test sets. It is also applied in the case of model-based testing, especially for (extended) finite state machines. In this paper, we define such an approach for kernel P system models. We consider a specific kernel P system model and define a search-based testing method. The test set generated consists of input sequences producing a given computation defined by the model. An example illustrates the use of the introduced method.

Gazdag, Z., & Kolonits, G. (2019). **A new method to simulate restricted variants of polarizationless P systems with active membranes.** *Journal of Membrane Computing*, 1(4), 251–261. <https://doi.org/10.1007/s41965-019-00024-z>

According to the P conjecture by Gh. Păun, polarizationless P systems with active membranes cannot solve NP-complete problems in polynomial time. The conjecture is proved only in special cases yet. In this paper we consider the case where only elementary membrane division and dissolution rules are used and the initial membrane structure consists of one elementary membrane besides the skin membrane. We give a new approach based on the concept of object division polynomials introduced in this paper to simulate certain computations of these P systems. Moreover, we show how to compute efficiently the result of these computations using these polynomials.

Buiu, C., & George, A. (2019). **Membrane computing models and robot controller design , current results and challenges.** *Journal of Membrane Computing*, 1(4), 262–269. <https://doi.org/10.1007/s41965-019-00029-8>

Designing membrane controllers for single- and multi-robot systems is an application area initiated in 2011 at the Laboratory of Natural Computing and Robotics (natuRO) of the Politehnica University of Bucharest. In this paper, an overview of natuRO's research on the design of robot controllers based on various models of membrane systems is given. After an introduction to robotics and natural computing, this paper follows multiple directions. Firstly, a description of three membrane system simulators is given, taking into account their evolution, comparative capabilities, and application areas for each of them. Secondly, the main part of this paper is a synopsis of the applications of membrane systems in robot control, while an emphasis is paid to the new membrane computing models introduced at natuRO, Enzymatic Numerical P Systems and XP colonies, and their specific use in single- and multiple-robot applications. Thirdly, the paper continues with a critical overview of the performances of membrane controllers as compared to traditional ways to control single- and multiple-robot systems, current challenges and possible ways to overcome these. Example avenues for future related works are given in the conclusion.

Jiang, Y., Su, Y., & Luo, F. (2019). **An improved universal spiking neural P system with generalized use of rules.** *Journal of Membrane Computing*, 1(4), 270–278. <https://doi.org/10.1007/s41965-019-00025-y>

Taken inspiration from biological phenomenon that neurons communicate via spikes, spiking neural P systems (SN P systems, for short) are a class of distributed and parallel computing devices. So far firing rules in most of the SN P systems usually work in a sequential way or in an exhaustive way. Recently, a combination of the two ways mentioned above is considered in SN P systems. This new strategy of using rules, which is called a generalized way of using rules, is applicable for both firing rules and forgetting rules. In SN P systems with generalized use of rules (SNGR P systems, for short), if a rule is used at some step, it can be applied any possible number of times, nondeterministically chosen. In this work, the computational completeness of SNGR P systems is investigated. Specifically, a universal SNGR P system is constructed, where each neuron contains at most 5 rules, and for each time each firing rule consumes at most 6 spikes and each forgetting rule removes at most 4 spikes. This result makes an improvement regarding to these related parameters, thus provides an answer to the open problem mentioned in original work. Moreover, with this improvement we can use less resources (neurons and spikes involved in the evolution of system) to construct universal SNGR P systems.

Andonie, R. (2019). **Hyperparameter optimization in learning systems.** *Journal of Membrane Computing*, 1(4), 279–291. <https://doi.org/10.1007/s41965-019-00023-0>

While the training parameters of machine learning models are adapted during the training phase, the values of the hyperparameters (or meta-parameters) have to be specified before the learning phase. The goal is to find a set of hyperparameter values which gives us the best model for our data in a reasonable amount of

time. We present an integrated view of methods used in hyperparameter optimization of learning systems, with an emphasis on computational complexity aspects. Our thesis is that we should solve a hyperparameter optimization problem using a combination of techniques for: optimization, search space and training time reduction. Case studies from real-world applications illustrate the practical aspects. We create the framework for a future separation between parameters and hyperparameters in adaptive P systems.

Manca, V. (2019). **From biopolymer duplication to membrane duplication and beyond.** *Journal of Membrane Computing*, 1(4), 292–303. <https://doi.org/10.1007/s41965-019-00018-x>  
The relationship between biopolymer duplication and membranes compartmentation is analyzed, by stressing their intertwining and the different monomers that determine their forms of aggregation (linear and circular) with their functions. Both polymers and membranes are prebiotic forms of molecular assemblies, but in their integration the seed of life emerges. From membranes hosting a replicative metabolism cells stem as living unities, where an almost perfect synthesis is realized between metabolism and duplication. What is missing to perfection becomes the basis of evolution. The whole logic of the processes at the origin of life is reconstructed in general terms, in the line of Manca (Infobiotics: information in biotic systems. Springer, New York, 2013), Manca (J Proteom Bioinform 11(7), 135–137, 2018) and Manca and Santagata (Un meraviglioso accidente. La nascita della vita. Mondadori, Italy, 2018), with no biochemistry detail, but only on the basis of the needs for representing, conserving, developing, and transmitting biological information.

**Online First** (as of 15.02.2020, check <https://link.springer.com/journal/41965/onlineFirst> for updates)

Andreu-Guzmán, J. A., Valencia-Cabrera, L. **A novel solution for GCP based on an OLMS membrane algorithm with dynamic operators.** *Journal of Membrane Computing*, <https://doi.org/10.1007/s41965-019-00026-x>

Graph coloring problem (GCP) is an NP-complete combinatorial optimization problem. Its computational complexity motivated many efforts to get approximate solutions through different meta-heuristics, such as several variants of evolutionary algorithms. On the other hand, membrane algorithms have appeared as alternative hybrid techniques merging together the structure and operators of membrane systems, along with the capabilities of optimization algorithms inside each membrane. This paper explores the ability of a new variants of one-level membrane systems using a recent variant of evolutionary algorithm dynamically using different genetic operators depending on the best fitness found. The experimental results presented show that this new algorithm, called DOGAPS, outperforms the dynamic evolutionary algorithm, with the extra value provided by the membrane system. Additionally, the role of some parameters involved in our algorithm are analyzed, including the number of membranes, iterations per membrane or mutation rate.

Freund, R. **How derivation modes and halting conditions may influence the computational power of P systems.** *Journal of Membrane Computing*, <https://doi.org/10.1007/s41965-019-00030-1>

In the area of P systems, besides the standard maximally parallel derivation mode, many other derivation modes have been investigated, too. In this overview paper, many variants of hierarchical P systems using different derivation modes are considered and the effects of using different derivation modes, especially the maximally parallel derivation modes and the maximally parallel set derivation modes, on the generative and accepting power are illustrated. Moreover, an overview on some control mechanisms used for P systems is given. Furthermore, besides the standard total halting, we also consider different halting conditions such as unconditional halting and partial halting and explain how the use of different halting conditions may considerably change the computing power of P systems.

Calude, C. S., Dinneen, M. J., Hua, R. **Quantum solutions for densest k-subgraph problems.** *Journal of Membrane Computing*, <https://doi.org/10.1007/s41965-019-00030-1>

In this paper, we present, for the first time, quantum annealing solutions for densest k-subgraph problems

which have many applications in computational biology. Our solutions are formulated as solutions for quadratic unconstrained binary optimization (QUBO) and integer programming problems, proved to be equivalent with the densest k-subgraph problems and then tested on an D-Wave 2X machine. The QUBO formulations are optimal in terms of the number of logical qubits, but require the highest number of physical qubits after embedding. Experimental work reported here suggests that the D-Wave 2X model cannot handle problems of this difficulty. The next generation of D-wave hardware architecture—the Pegasus architecture—is much denser than the current one, so dense QUBOs will be easier to embed. The experimental work also suggests that the current built-in post-processing optimization method does not work very well for some problems and the default setting (post-processing optimization on) should be avoided (or at least tested before being turned on).