P Systems Implementation: A Model of Computing for Biological Mitochondrial Rules using Object Oriented Programming

Mohammed M. Nasef[1]
Mathematics and Computer Science Department
Faculty of Science, Menoufia University
Shebin El-Koom,32511, Egypt

Bishoy El-Aarag[2]
Biochemistry Division, Chemistry Department
Faculty of Science, Menoufia University
Shebin El-Koom,32511, Egypt
Division of Chemistry and Biotechnology
Graduate School of Natural Science and Technology
Okayama University, Okayama 7008530, Japan

Amal Hashim[3]
Mathematics and Computer Science Department
Faculty of Science, Menoufia University
Shebin El-Koom,32511
Information Systems Department
Higher Institute of Advanced Studies, Haram, Giza, Egypt

Passent M. El Kafrawy[4]
Mathematics and Computer Science Department
Faculty of Science, Menoufia University
Shebin El-Koom,32511
School of Information Technology and Computer Science
Nile University, Egypt

***Abstract*—Membrane computing is a computational framework that depends on the behavior and structure of living cells. P systems are arising from the biological processes which occur in the living cells' organelles in a non-deterministic and maximally parallel manner. This paper aims to build a powerful computational model that combines the rules of active and mobile membranes, called Mutual Dynamic Membranes (MDM). The proposed model will describe the biological mechanisms of the metabolic regulation of mitochondrial dynamics made by mitochondrial membranes. The behaviors of the proposed model regulate the mitochondrial fusion and fission processes based on the combination of P systems variants. The combination of different variants in our computational model and their high parallelism lead to provide the possibility for solving problems that belong to NP-complete classes in polynomial time in a more efficient way than other conventional methods. To evaluate this model, it was applied to solve the SAT problem and calculate a set of computational complexity results that approved the quality of our model. Another contribution of this paper, the biological models of mitochondrial is presented in the formal class relationship diagrams were designed and illustrated using Unified Modeling Language (UML). This mechanism will be used to define a new specification of membrane processes into Object-Oriented Programming (OOP) to add the functionality of a common programming methodology to solve a large category of NP-hard problems as an interesting initiative of future research.**

Keywords*—Computational biology; P systems; membranes fusion – fission; mitochondria; Mutual Dynamic Membranes (MDM); NP- complete problems

I. INTROUDCTION

A lot of ideas in Bioinformatics have attracted the attention of researchers. These ideas have been inspired by living cells, their organics, and their membranes that have been used to compute and model a large number of problems in computational science. Natural Computing is a wide field that describes computational methods and processes inspired by natural environments. The natural computing frameworks that are known to solve computational problems are, for example: DNA computing [1], automata computing [2], quantum computing [3], genetic algorithms [4], and neural networks [5]. Similarly, membrane computing is abstracted to solve hard problems in Artificial Intelligence (AI) [6].

Membrane computing is introduced by Păun in 1998 and was inspired by paradigmatic computation based on the structure, behavior, and functions of living cells [7]. One of such outcomes is the P systems, a category of various biological parallel computing models which can be considered as universal computing paradigms [8]. P systems projected efficient computational models by combining the structural and dynamic properties of biological systems [9]. As a result, many fields such as the formal languages [10], theory of computability [11], and theory of computational complexity [12] with intractable problems were solved by the biological model, called P systems [13].

Some of the biological phenomena of the living cells were incorporated into membrane computing as an influential kind of computational concepts, cell-like P systems. According to these phenomena, the framework of active membranes in polarizationless P systems [14] are developed and evaluated by using rewriting rules [7] which can be merged, divided, separated, or passed through membranes [15]. Another variant of cell-like P system was inspired by cells movement as exocytosis and endocytosis rules that are expressed of the outside and inside processes of neighboring membranes, this variant is called mobile membranes [16].

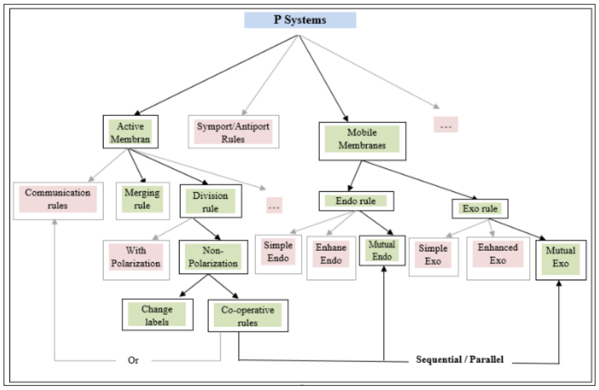


Fig. 1. P Systems Variants.

A general form of cell-like P systems variants is shown in Fig. 1. Each variant has a set of rules describing the basic structure of any model designed to solve computational problems. The main components used in the proposed model are represented by green arrows and compartments in Fig. 1.

The living cells have organelles called mitochondria; mitochondrion is a dynamic organelle capable of interacting with each other. It is responsible for producing energy molecules in the living cell, Adenosine Triphosphate, (ATP) [17]. The function of mitochondria depends on four processes, fusion, fission, motility, and Mitophagy [17]. In the current work, the focus is on the first two operations, fusion and fission that will be introduced in a powerful computational model. The influential division rule is cooperative with mutual exo and mutual endo rules in (MDM) model. All rules are applied in parallel and non-deterministically selecting the membranes, the rules, and the objects. In the case a set of rules can be executed in each step, but no other rules can be added to this set and no membranes and objects can be evolve at the same time; this is indicated to the parallelism mechanism is maximal for a final solution.

The basic idea is to develop a set of rules that will be the bases of a model to solve NP hard problems through the interactions of the objects, those set of rules are an abstraction of the natural biological generation of energy in human cells. From which conclusions can be achieved. The benefit is to develop a new set of AI models that do not need a priori knowledge nor learning, and moreover to model standardization in smart application development.

Our target is to represent the computability and complexity of an unconventional computing system as a theoretical solvent for NP-complete problems in Polynomial/ linear time. In the current work, the combination of variants of P systems in the model, using real operations, leads to solve a number of problems belonging to different classes further than NP in a more efficient manner than other classical computing systems. A semi-uniform linear time solution has been introduced to SAT problem using (MDM), with 3 membranes only as described in the initial configuration of the fission model.

Finally, another side is the need for alternative mechanisms and unconventional methods to describe and illustrate complex biomolecular processes which have actions, behaviors, and functions are vitality to the health of human. Accordingly, to this aspect, our biological models are considered a computational method for the development of object relationship diagrams. Then the objects structure and their actions are represented by UML diagrams. These diagrams can be implemented using object-oriented programming to be used as an analysis tool for complex data of mitochondria neural diseases to extract the diseases' characteristics, causes, and insights as an initiation in huge applications of data science. The object relationship diagrams will be used to develop machine learning algorithms in the future work.

This paper is organized as follows: Section II introduces the concepts of active and mobile membranes, as well as their basic conceptions of P systems. The third section is dedicated to the elaboration of the related work. The proposed model, (MDM) is introduced and modeled in Section IV. Efficient Solutions to different classes of problems by the proposed model and a semi-uniform linear time solution have been introduced to SAT problem using the proposed model (MDM), with 3 membranes only as presented in the initial configuration of the fission model is explained in Section V. UML diagrams representations will be introduced in Section VI. A final conclusion and some future researches lines are given in Section VII.

II. Definitions of P Systems with Active and Mobile Membranes

The distributed parallel computing models included P systems class arisen from the membrane computing framework. In the present work, the important processes in mitochondria are recruitment. The combination of two important classes of rules described in P systems models, active and mobile membranes are presented as a new model, (MDM).

The first class is the active membranes variant. It has evolution, merging, division, and communication rules. The second class is the mobile membranes variant, which is inspired by biological events such as cell membrane mobility, Endocytosis, and Exocytosis processes. It has four types: simple, enhanced, mutual mobile membranes and mutual mobile membranes with objects on the surface [18]. We will focus on the third type, mutual mobile membranes. It's worth noting that the rules permit a membrane to move independently of the other membranes involved. According to this property, the mutual mobile membrane is completely suitable for fusion and fission mitochondria.

The fundamental notions of P systems with active and mutual mobile membranes are constructed as follows:

Π= (V, H, µ, W, R, i), where:

1) *V* is the alphabet (non-empty and finite) objects;

2) *H* is a finite set of labels for membranes, labeled h_1, h_2, …, h_n; ($n \geq 1$ the degree of the system);

3) *µ* is the membrane structure, composed of ***n*** membranes;

4) *W* is string, w_1, w_2, …, w_n on *V* which represent the multisets of objects present in regions h_1, h_2, …, h_n of the membrane structure *µ*;

5) *R* is finite sets of evolutionary rules related to regions $h_1, h_2, ..., h_n$ of membrane structure, for the following forms;

6) $i \in H \cup \{environment\}$, is either one of the regions $h_1, h_2, ..., h_n$ and the respective region are the output range of the system, or it is ***0***, where *i = environment.*

Fig. 2 presents the class of P systems with active and mobile membrane rules as follows:

a) $[a \rightarrow v]_{h1}$, where $a \in V$; $v \in V^*$; $h_1 \in H$.

Object evolution rule; associated with the membrane and depending on the label, but not directly involving the membrane, in the sense that the membrane is neither taking part in the application of this rule nor modified by it [7, 19, and 20].

b) $a[\ ]_{h1} \rightarrow [b]_{h2}$,where a ,b $\in$ V; $h_1, h_2 \in H$.

In-Communication rule; an object is sent in the membrane $\boldsymbol{h_2}$ [7, 19, and 20].

c) $[b]_{h2} \rightarrow a[\ ]_{h1}$,where a ,b $\in$ V; $h_1, h_2 \in H$.

Out-Communication rule; an object is sent out of the membrane $\boldsymbol{h_2}$ [7, 19, and 20].

d) $[a]_{h1} \rightarrow [b]_{h2}\ [c]_{h3}$, where a, b, c $\in$ V; $h_1, h_2, h_3 \in H$.

Division rule for elementary membrane h_1; in reaction with an object a, the membrane is divided into two new membranes with the different labels h_2 and h_3, the object specified in the rule is replaced in the two new membranes by possibly new objects b and c; the new objects may evolve in the same step [7, 20]. The object specified in this rule may be replaced in the two new membranes by possibly new objects or they are duplicated and may evolve in the same step by rule of type (a) [7].

e) $[b]_{h2}\ [c]_{h3} \rightarrow [a]_{h1}$, where a, b, c$\in$ V; $h_1, h_2, h_3 \in H$.

Merging rule for elementary membranes h_2 and h_3; in reaction with their objects b and c, they are merged into a single membrane h_1; the object a is produced in the new membrane h_1 [7, 20].

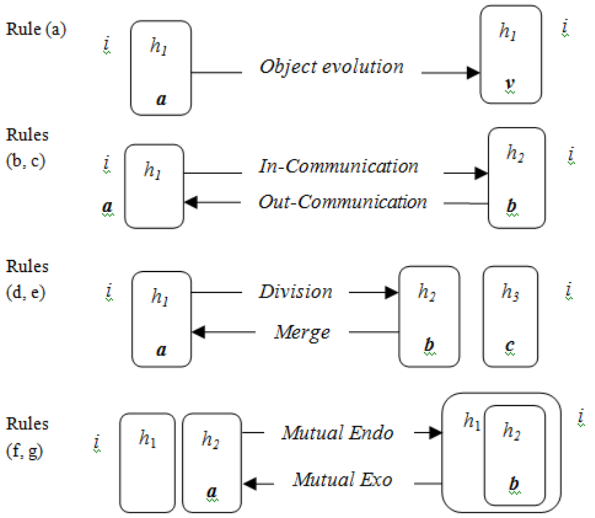


Fig. 2. Membranes Handling Operations.

f) $[\ ]_{h1}\ [a]_{h2} \rightarrow [[b]_{h2}]_{h1}$, where a, b$\in$ V; $h_1, h_2 \in H$.

Mutual Endocytosis (mendo): Under the management of object a, an elementary region labeled h_2 enters the next region named h_1; the labels h_1 and h_2 stay the same during this process. But the object a may be adapted to b during the process; h_1 isn't always seen as a fundamental region [7, 19].

g) $[[b]_{h2}]_{h1} \rightarrow [\ ]_{h1}[a]_{h2}$,where a ,b $\in$ V; h_1 , $h_2 \in H$.

Mutual Exocytosis (mexo): an elementary region labeled h_2 is sent out of a region labeled h_1, The labels h_1 and h_2 remain unchanged during this procedure, as they are under the control of object b, however, the object b may be changed to a during this operation; region h_1 is not always an elementary region [7, 19].

In [21] one observed that the electrical charges (polarization) are not adapting to biological rules. In fact, using the polarization features with mobile rules as exo and endo rules were not common in membranes systems. Although of that, P systems with mobile membranes have given computational power and universality properties. Their efficiency and ability are used to solve NP-complete problems without using the polarization [22].

The rules of type (a) are applied in parallel (all objects that may be developed by such rules must evolve). However, the rules of types (b) through (g) are implemented in such a way that only one rule of these kinds may be utilized at a time on one membrane. Overall, the rules are applied in a non-deterministic, maximally parallel fashion; all objects and membranes that may evolve should do so. These rules modify the label of the membrane, but they must be implemented in a sequential manner to avoid label conflicts.

Rules (b) and (c) are considered as a simple form without polarization, those two powerful rules are used for in and out communication as illustrated in [23]. Even in the limited instance where no polarization is utilized, membrane labels are modified. While in rules (d) and (e), P systems are used without polarization and the membrane division and merging rules are allowed to change the label of a membrane (Theorem 2 in [21]). This condition is to accomplish P system efficiency and universality without polarization.

Finally, under the control of object a, the elementary membrane labeled h_2 enters the next membrane named h_1; the labels h_1 and h_2 stay unchanged during this operation. Hence, object a may be changed to b throughout the endocytosis process, rule (f). Similarly, under the control of object b, an elementary membrane labeled h_2 is expelling from a membrane labeled h_1; the two membranes labels stay the same. However, the object b from membrane h_2 can be modified throughout the exocytosis process, rule (g).

In all of the variants, the number of membranes can only be reduced during computation by merging membranes as a result of applying evolution rules to the objects represented in the system. A natural possibility is to allow increases in the number of membranes also to increase during a computation. For example, by division, as it is well known in biology. Actually, the membranes from biochemical are not passive. Where, the majority of chemical compound passage across a membrane is accomplished by direct interaction with the membrane itself (through the membrane's so-called protein channels or protein gates). For the duration of this interaction, the chemical compounds that pass during membrane can be modified, while the membrane itself can in this way be modified at least locally.

III. Related Work

Researches of recent literature are well presented with grateful studies in P systems models that support the computing paradigm known as membrane computing. In addition to simulating the biomolecular processes as fusion and fission of mitochondria as we will introduce in this work the same behavior with P system variants to improve solutions for NP-complete problems. We mention a few reviews of the membrane computing concepts and their notions [24-29]. Membrane computing is similar to quantum computing, is a new unconventional computing model that is applied in linguistics, sociology, optimization design, and a large number of fields [30]. The researcher can read [27, 28, 31, 32, and 19] to cover widely part of P systems variants.

All of the studies are represented and developed to contribute to solving different kinds of problems from formal languages theory [24] to power systems blunder diagnosis [34], passing through a wide range of different research fields as in chemical engineering, scheduling of gasoline blending [35] by membrane computing with its application. Also, in the pattern recognition letters, an automatic clustering algorithm by P systems has been applied [36]. All of the mentioned applications used different P systems models; developed according to the problem nature whether it is theoretical or practical. Here, the focus is on the literary works that addressing computationally NP-complete problems using active and mobile membrane systems and their efficiency and computational power. The efficient variants of Membrane systems and the formal definitions of P systems are investigated from computational complexity theory [34]. Elegant works for NP-complete problems such as Boolean Satisfiability (SAT) problem, has been shown with the polynomial solutions with active membrane principles [26, 16], The Hamiltonian Cycle Problem (HCP) [37], and Travelling Salesman Problem (TSP) [38]. SAT solution is also set by P systems with active membranes by division rule for elementary membranes without dissolution rule. It's verified in [39]. Separation rules are utilized instead of division rules in [40], where it created two new membranes each of them has one new object and the rest of them are replicating. These rules had restrictions that created an exponential workspace in membranes terms in polynomial time, not in objects [41, 42]. The Turing completeness is achieved by using elementary division rules for nine membranes [16, 43].

Regarding the computational complexity of mobile membranes, four mobile membranes are obtained to prove a Turing machine power using evolution rules with exo and endo rules [44].

P automata variant is used to describe a mitochondrial fusion model [45]. It represented the procedure of specific protein production that is necessary for mitochondrial fusion [17]. In [33], Giannakis and Andronikos referred to use six biological operations described in [46]. The most important process of them is the exocytosis operation used in the mitochondrial fusion model that concerns a mitochondrial fusion mechanism with communication orders and specific actions. The comparison between our model and their study [33] will be introduced in Table I.

All of the studies are represented and developed to contribute to solving different kinds of problems from formal languages theory [24] to power systems blunder diagnosis [34], passing through a wide range of different research fields as in chemical engineering, scheduling of gasoline blending [35] by membrane computing with its application. Also, in the pattern recognition letters, an automatic clustering algorithm by P systems has been applied [36]. All of the mentioned applications used different P systems models; developed according to the problem nature whether it is theoretical or practical. Here, the focus is on the literary works that addressing computationally NP-complete problems using active and mobile membrane systems and their efficiency and computational power. The efficient variants of Membrane systems and the formal definitions of P systems are investigated from computational complexity theory [34]. Elegant works for NP-complete problems such as Boolean Satisfiability (SAT) problem, has been shown with the polynomial solutions with active membrane principles [26, 16], The Hamiltonian Cycle Problem (HCP) [37], and Travelling Salesman Problem (TSP) [38]. SAT solution is also set by P systems with active membranes by division rule for elementary membranes without dissolution rule. It's verified in [39]. Separation rules are utilized instead of division rules in [40], where it created two new membranes each of them has one new object and the rest of them are replicating. These rules had restrictions that created an exponential workspace in membranes terms in polynomial time, not in objects [41, 42]. The Turing completeness is achieved by using elementary division rules for nine membranes [16, 43].

TABLE I. Comparison between our Model and Related Work [33]

Model in [33]	**Our Model**
Only one operation (fusion operation) is modeled.	Two operations (fusion and fission) are modeled.
Considered that the outer and inner membranes are two individuals in skin membranes; each of them has three nested elementary membranes. This is not matched with the structure of biological model.	Biological model is precisely designed to describe the operations occurred in the two nested mitochondria membranes and regulates the fusion and fission processes.
Depended on Bio Ambients rules with Mobile membranes to design the model.	Depended on combination of active membranes rules with mobile membranes to design the model.
Used an exo behavior to achieve the final configuration.	Used a combination of division, mexo and mendo behaviors to achieve the final configuration.
Proteins have been produced from the outer and inner membranes in the final configuration. This is not matched with mitochondria function and their membranes.	New mitochondria organelles have been produced from fusion and fission processes. These mitochondria organelles are capable to adapt its metabolic and produce **ATP** molecules.
	Rules of our model have been given computational efficiency for many **NP** complete problems.

Regarding the computational complexity of mobile membranes, four mobile membranes are obtained to prove a Turing machine power using evolution rules with exo and endo rules [44].

P automata variant is used to describe a mitochondrial fusion model [45]. It represented the procedure of specific protein production that is necessary for mitochondrial fusion [17]. In [33], Giannakis and Andronikos referred to use six biological operations described in [46]. The most important process of them is the exocytosis operation used in the mitochondrial fusion model that concerns a mitochondrial fusion mechanism with communication orders and specific actions. The comparison between our model and their study [33] will be introduced in Table I.

IV. Proposed Model

To present our model, firstly we describe the biological background for mitochondria actions (fusion and fission models), after that we will describe the biological actions by Mutual Dynamic Membranes (MDM) with P Systems.

A. Mitochondrial Fusion Model

A mitochondria fusion mechanism includes specific actions and communications motivatedby the structure and the functioning of the distribution of mitochondrial fusion in the renewal of the mitochondrial population within an eukaryotic cell.

We will demonstrate the proposed scheme using an actual biochemical model of mitochondrial fusion processes. The proposed model consists of five actual membranes, the cell which acts as an "environment" in the P systems. The skin membrane has two non-elementary membranes called "Outer Membranes" with each one has one elementary membrane called "Inner Membranes". Both inner and outer membranes have operations "rules" executed by proteins that are considered as "objects" in the P systems. Then, a new mitochondrion is produced in the environment.

1) Fusion biological model: As illustrated in Fig. 3, the biological description of a mitochondrial fusion model is designed by Cell Designer software. In the fusion model, the joining of two organelles into one was applied through two basic operations, Outer and Inner membrane fusion.

a) First operation: The fusion of outer membrane: There are several proteins involved in the fusion process of the outer membrane such as mitofusin 1 and 2 (MFN1& MFN2). Oxidative stress and the presence of high levels of oxidized glutathione (GSSG) are inducer signals for outer membrane fusion. These signals induced serial of constitutive steps as the following: oxidative stress and the excess formation of glutathione oxidized form (GSSG) induce mitofusins complexes in the outer membrane to act in trans. This induction of trans complexes of mitofusin proteins was mediated by disulfide bonds owing to the binding of organelle and causing fusion of the outer membranes [47].

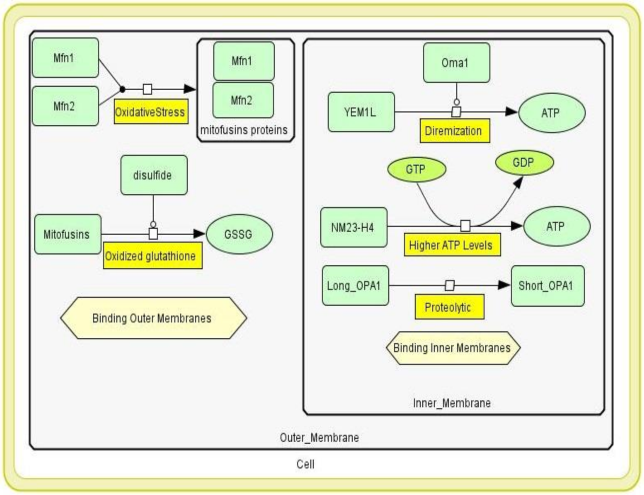


Fig. 3. Fusion Operations for Mitochondria Membranes.

b) Second operation: The fusion of inner membrane: A particular protein termed Optic Atrophy 1 (OPA1) has participated in the fusion of the inner membrane. The fusion was initiated by oxidative phosphorylation (OXPHOS) and advanced ATP levels as inducer signals. Concerning to OXPHOS signal these steps have occurred as follows:

- The presence of OXPHOS increases Yme1L metalloprotease. Yme1L and Oma1 activate the Proteolytic processing of Opa1 from the long-form to the soluble short form leading to inner membrane fusion.
- With regards to higher ATP levels, this signal was linked with GTP-loading and hydrolysis by the action of the kinase (NM23-H4) leading to conversion of OPA1 to short-form causing inner membrane fusion [47].

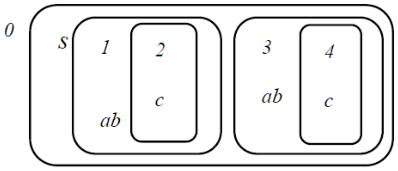
2) Fusion membranes using Mutual Dynamic Membranes (MDM) with P Systems: Active and mobile membranes with P systems are used to obtain the behavior of cooperation rules such that using more than one object on the left-hand side of the rule. The biochemical reactions in mitochondria organelles are usually created by the reaction of two or more proteins represented as normal objects that can improve through the computation to get the final configuration.

In the following configuration, proteins MFN1, MFN2, and OPA1 are used as "activators" of rules; these proteins are engaged in the fusion process and are important for the regulation of certain chemical processes. These proteins are also responsible for determining which rules apply in each phase leading to a final configuration.

$\Pi_1 = (V, H, \mu, W, R, \boldsymbol{i}_0)$ be a **P** system with the initial degree of the system is $\boldsymbol{n = 5}$.

- The system can be viewed as a set of five membranes labeled by elements of $\boldsymbol{H = \{s,1,2,3,4\}}$ arranged in a hierarchical structure which including skin membrane '***s***' contains two non-elementary membranes '***1***' and '***3***' each one of them has one elementary membrane, ‘**2**’ in ‘**1**’ and ‘**4**’ in ‘**3**’.

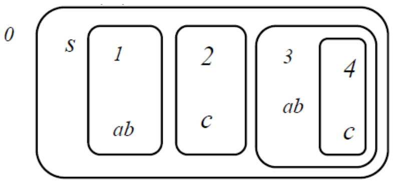
- The initial configuration, $\boldsymbol{\mu}$ given by $\mathbf{[[[]_2]_1[[]_4]_3]_s}$
- The finite multisets of objects are represented by $\boldsymbol{w_1, w_2, w_3, w_4, w_s}$ *where* $\boldsymbol{w_s = \phi}$*,* $\boldsymbol{w_1=w_3}$ have proteins of outer membranes, *MFN1* and *MFN2*. And $\boldsymbol{w_2=w_4}$ have proteins of inner membranes, *OPA1*. These are placed in membranes of the initial system. For simplicity, we will be considered *MFN1=a, MFN2=b, OPA1=c, new_MFN1= a`, new_ MFN2= b`* and *new_OPA1=c`.* Such that, *a, b, c, a`, b`, c`∈ V*
- ***R*** is a finite set of the development rules over ***V*** of the following forms, **(*R1, R2, …, R6).***
- ***i*** is *0*, refer to the environment, “*cell*” in the system.



Initial Configuration

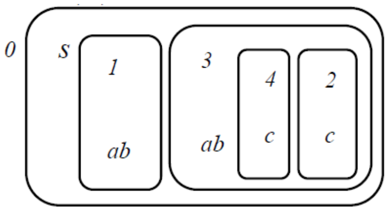
1) Apply mexo rule on membranes 2 and 1.

R1: $[\,[\,a\,b\,[\,c\,]_2]_1[\,a\,b\,[\,c\,]_4\,]_3]_s \rightarrow [\,[\,a\,b]_1\,[\,c]_2\,[\,a\,b\,[c\,]_4\,]_3\,]_s$, *where a, b, c ∈V.*



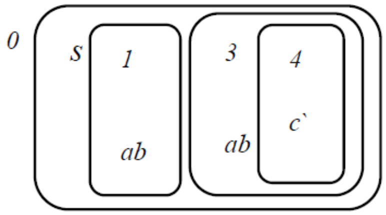
2) Apply mendo rule on membranes 2 and 3.

R2: $[\,[\,a\,b\,]_1[\,c]_2\,[\,a\,b\,[c\,]_4\,]_3\,]_s \rightarrow [\,[\,a\,b\,]_1\,[a\,b\,[\,c\,]_4\,[\,c\,]_2\,]_3\,]_s$, *where a, b, c ∈V.*



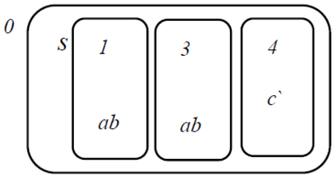
3) Apply merging rule on membranes 2 and 4 (the new membrane has label 4)

R3: $[\,[\,a\,b\,]_1\,[\,a\,b\,[\,c\,]_4\,[\,c\,]_2\,]_3]_s \rightarrow [\,[\,a\,b\,]_1\,[\,a\,b\,[\,c^{\backprime}]_4\,]_3]$ s, *where a, b, c, c`∈V.*



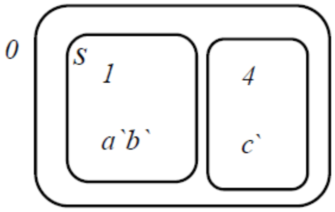
4) Apply mexo rule on membranes 4 and 3

R4: $[\,[\,a\,b\,]_1\,[\,a\,b\,[\,c^{\backprime}\,]_4\,]_3]_s \rightarrow [\,[\,a\,b\,]_1\,[a\,b\,]_3\,[c^{\backprime}]_4]_s$, *where a, b, c, c`∈V.*



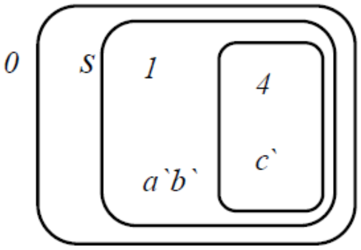
5) Apply merging rule on membranes 1 and 3(the new membrane has label 1)

R5: $[\,[\,a\,b\,]_1\,[a\,b\,]_3\,[\,c^{\backprime}]_4]_s \rightarrow [\,[a^{\backprime}b^{\backprime}\,]_1\,[\,c^{\backprime}]_4\,]_s$, *where a, b, a`, b`, c`∈V.*



6) Apply mendo rule on membranes 4 and 1

R6: $[\,[a^{\backprime}b^{\backprime}\,]_1\,[\,c^{\backprime}]_4]_s \rightarrow [\,[a^{\backprime}b^{\backprime}\,[\,c^{\backprime}]_4\,]_1]_s$, *where a`, b`, c`∈V.*



Final Configuration

B. Mitochondrial Fission Model

A mitochondria fission mechanism is described by variants of active and mobile membranes using realistic biological operations. The proposed model consists of three actual membranes, the skin membrane which is a surface membrane, and the cell that acts as an “environment” in the P systems. The skin membrane has one non-elementary “Outer Membrane” that has one “Inner Membrane”. Both inner and outer membranes have operations “rules” executed by proteins considered “objects” in the P systems. After that, two mitochondria are produced in the environment.

1) Fission biological model: Fig. 4 presented the biological description of mitochondrial fission regulation model. It designed by cell designer software. According to division operation, the fission model was included dividing one mitochondria organelle into two new ones. It should be taken in consideration that a dynamin-related protein 1 (Drp1) is the principal controller protein complicated in fission process. The Drp1 mobilization from cytosol onto the surface of mitochondria is essential and established via many proteins found on the outer membrane of mitochondria. They included mitochondrial fission factor (Mff) and mitochondrial dynamics proteins 49 and 51 (MiD49 and MiD51). The regulation of fission process is completed by one of the following four basic modes [47]:

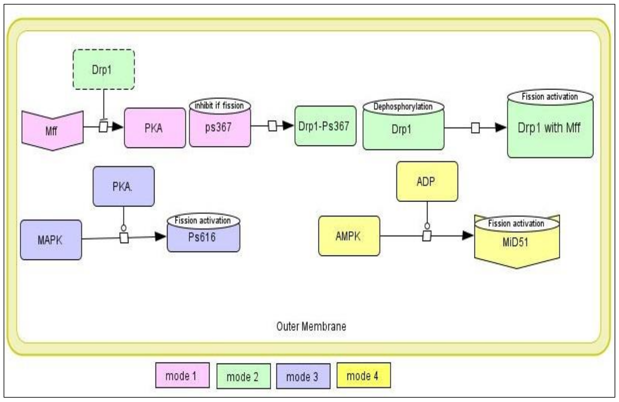


Fig. 4. Fission Operations for Mitochondria Membranes.

Mode (1): showed the inhibition of mitochondrial fission. It induced by signals included exercise and nitrogen starvation. The mode was done through the following sequencing steps:

1) Exercise and nitrogen starvation induce protein kinase A (PKA) activation.

2) The activated PKA causes phosphorylation of Drp1 at amino acid serine no. 637 (Ser637).

3) Phosphorylated Drp1 keep hold of Drp1 in cytosol preventing the fission process.

Mode (2): revealed the induction of mitochondrial fission. It initiated via signals such as metabolic uncoupling of the organelle and calcineurin.

The mode was done through the following sequencing steps:

1) Metabolic uncoupling of the organelle activates calcineurin.

2) The activated calcineurin dephosphorylates Drp1 at Ser637 (i.e. convert Drp1 from the phosphorylated to dephosphorylated form).

3) Dephosphorylated Drp1 allows mobilization of Drp1 to mitochondria resulted in activation of mitochondria fission.

Mode (3): exhibited the induction of mitochondrial fission. It began through signals such as cold exposure and oncogenic RasG12V. The mode was done through the following sequencing steps:

1) Cold exposure activates PKA, as well as oncogenic RasG12V activates mitogen-activated protein kinase (MAPK).

2) The activated PKA and MAPK phosphorylates Drp1 at Ser616.

3) The phosphorylatedDrp1 links with both MiD51 and Mff leading to activation of mitochondrial fission.

Mode (4): illustrated the induction of the mitochondrial fission through high adenosine diphosphate (ADP) and adenosine monophosphate (AMP) levels as induction signals. The following sequencing steps were included in the fission mode.

I- Steps regards to high ADP level signal:

1) The presence of high levels of ADP initiates the binding of ADP with MiD51 receptor.

2) The bounded ADP with receptor induces MiD51 activation.

3) The activated MiD51 associates with Drp1.

4) The activated and associated MiD51 with Drp1 causes Drp1 mobilization to mitochondria resulted in activation of mitochondria fission.

II-Steps regards to high AMP level signal:

1) The elevated AMP levels are detected by AMP-activated protein kinase (AMPK).

2) AMPK causes the phosphorylation of Mff protein receptor.

3) The phosphorylated Mff receptor binds to Drp1.

4) Drp1 bounded to phosphorylate Mff organizes Drp1 to mitochondria and activates mitochondria fission.

2) Fission membranes using Mutual Dynamic Membranes (MDM) with P system: The chemical reactions in mitochondria organelles are mainly caused by the reaction of two or more proteins. These are depicted as normal objects that can evolve by division computation, mexo, and mendo rules to get the final configuration.

In the following computation, proteins DRP1, MFF, MID51, PKA, Calcineurin, AMPK, and MAPK are used in P systems as "activators" of rules, where proteins are regulated certain biochemical reactions to the fission process occur.

Π2= (V, H, μ, W, R, i) be a P system with active and mobile membranes with the initial degree of the system is n = 3.

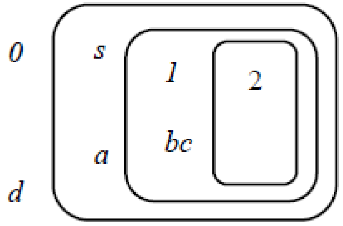
- The system can be viewed as a set of three membranes labeled by elements of H= {s, 1, 2} arranged in a hierarchical structure which including the skin membrane 's' contains two nested membranes '1' and '2'.

- The initial configuration μ given by [[[]2]1] s.

- The finite multisets of objects are represented by w0, ws, w1, w2 where w0 has proteins of the environment, ws has proteins of the surrounding of the outer membrane, w1 has proteins of an outer membrane and w2 has proteins of an inner membrane. These are placed in membranes of the initial system. For simplicity, we will be considered DRP1=a, DRP1-PS637 = as637, DRP1-PS616 = as616, MFF=b, Phospho-MFF=bp, MID51=c, PKA=d, Calcineurin=e, AMPK=f, MAPK=g. Then, w0=d, f, g, ws=a, e, w1=b, c and w2=ϕ. Such that, a, as637, as616, b, bp, c, d, e, f, g∈ V.

- R is a finite set of the development rules in V of the following forms associated with the labels including evolution, communication, division, mexo, and mendo rules, (R7, R8, ..., R33).

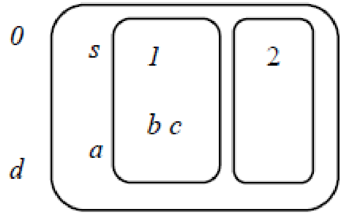
- i is 0, refer to the environment, “cell” in the system.



Initial Configuration of Mode 1

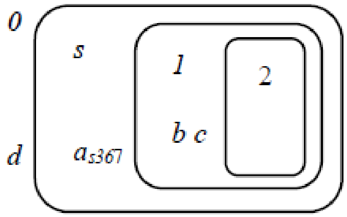
1) Apply mexo rule on membranes 2 and 1.

R7: $d\,[a\,[b\,c\,[\;]_2\,]_1]_s \rightarrow d\,[a\,[\,b\,c]_1\,[\;]_{2\;s}$, *where a, b, c, d ∈V.* (mode1)

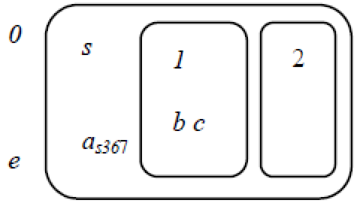


2) Apply object evolution rule in the same region [a → a_{s367}]

R8: d $[a\,[b\,c]_1[\;]_2]_s \rightarrow$ d $[a_{s367}\,[\,b\,c]_1\,[\;]_2]_s$, where a, b, c, d, a_{s367}



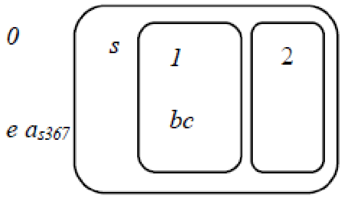
Note: a_{s367} object leads to inhibition of mitochondrial fission, then it interaction with e object to activate fission process in mode.



Initial Configuration of Mode 2

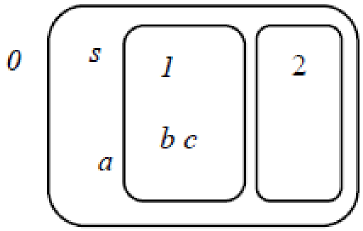
3) Apply communication rule; a_{s367} object is sent out from the skin membrane s.

R9: $e\,[a_{s367}\,[b\,c]_1\,[\;]_2]_s \rightarrow e\,a_{s367}\,[\,[\,b\,c]_1\,[\;]_2]_s$, *where a, b, c, e, a_{s367} ∈V.* (mode2)



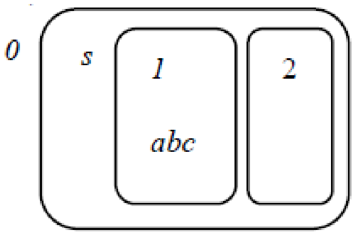
4) Apply object evolution rule [$a_{s367} \rightarrow a$]

R10: $e\,a_{s367}\,[\,[b\,c]_1\,[\;]_2]_s \rightarrow [\,a\,[b\,c]_1\,[\;]_2]_s$, *where a, b, c, e, a_{s367} ∈V.*



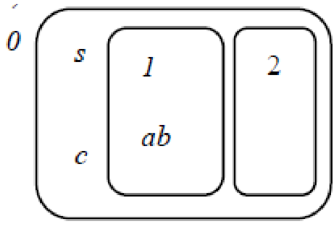
5) Apply communication rule; a object is sent in the membrane 1

R11: $[a\,[b\,c]_1\,[\;]_2]_s \rightarrow [\,[a\,b\,c]_1\,[\;]_2]_s$, *where a, b, c, ∈V.* (mode2)



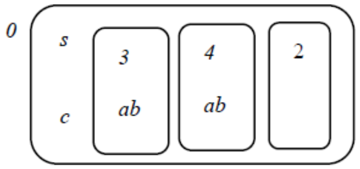
6) Apply communication rule; c object is sent out from the membrane 1

R12: $[\,[a\,b\,c]_1\,[\;]_2]_s \rightarrow [c\,[a\,b]_1\,[\;]_2]_s$, *where a, b, c, ∈V.* (mode2)



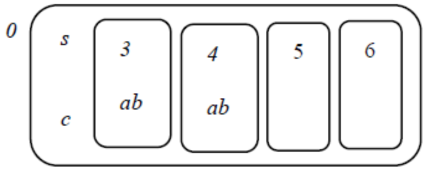
7) Apply division rule on membrane 1 (the different membranes labels 3, 4 are results)

R13: $[c\,[a\,b]_1\,[\;]_2]_s \rightarrow [c\,[a\,b]_3\,[a\,b]_4\,[\;]_2]_s$, where *a, b, c∈ V.* (mode2)



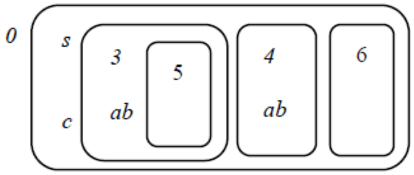
8) Apply division rule on membrane 2 (the different membranes labels 5, 6 are results).

R14: $[c\ [a\ b]_3\ [a\ b]_4\ [\ ]_2\ \ ]_s \rightarrow [c\ [a\ b]_3\ [a\ b]_4\ [\ ]_5\ [\ ]_6\ ]_s$, where $a, b, c \in V$. (mode2).



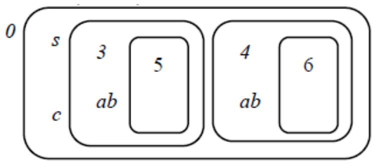
9) Apply mendo rule on membranes 5 and 3.

R15: $[c\ [a\ b]_3\ [a\ b]_4\ [\ ]_5\ [\ ]_6]\ _s \rightarrow [c\ [a\ b\ [\ ]_5\ ]_3[a\ b\ ]_4\ [\ ]_6]\ _s$, *where* $a, b, c \in V$. (mode2)

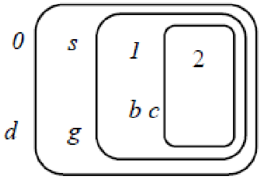


10)Apply mendo rule on membranes 6 and 4.

R16: $[c\ [a\ b\ [\ ]_5\ ]_3[a\ b\ ]_4\ [\ ]_6]\ _s \rightarrow [c\ [ab\ [\ ]_5\ ]_3[a\ b\ [\ ]_6]_4]\ _s$, *where* $a, b, c \in V$. (mode2)

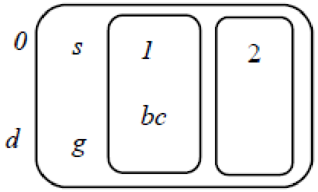


Rules of mode3:



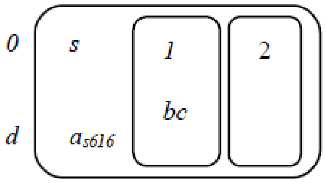
1) Apply mexo rule on membranes 2 and 1

R17: $d\ [g\ [b\ c\ [\ ]_2\ ]_1]\ _s \rightarrow d\ [g\ [b\ c]_1[\ ]_2]\ _s$, *where* $b, c, d, g\ \in V$.



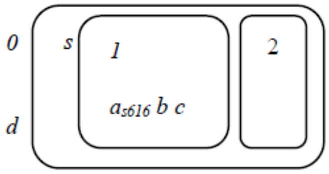
2) Apply object evolution rule in the same region, [g→a_{s616}].

R18: $d\ [g\ [b\ c\ ]_1[\ ]_2]\ _s \rightarrow d\ [a_{s616}\ [b\ c\ ]_1[\ ]_2]\ _s$, where $b, c, d, g, a_{s616} \in V$.



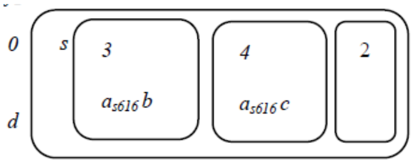
3) Apply communication rule; a_{s616} sent in the membrane 1,

R19: $d\ [a_{s616}\ [b\ c\ ]_1[\ ]_2\ _s \rightarrow d\ [\ [a_{s616}\ b\ c\ ]_1[\ ]_2]\ _s$, where $b, c, d, a_{s616} \in V$.



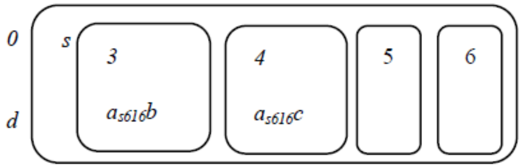
4) Apply division rule on membrane 1 (the different membranes labels 3, 4 are results).

R20: $d\ [\ [a_{s616}\ b\ c]_1[\ ]_2]\ _s \rightarrow d\ [\ [a_{s616}\ b]_3[a_{s616}\ c]_4\ [\ ]_2]\ _s$, where $b, c, d, a_{s616} \in V$.



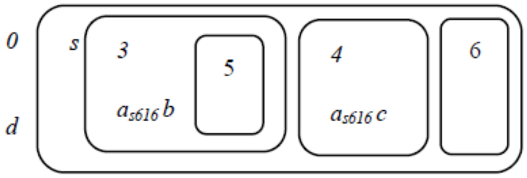
5) Apply division rule on membrane 2 (the different membranes labels 5, 6 are results).

R21: $d\ [\ [a_{s616}b]_3[a_{s616}c]_4\ [\ ]_2]\ _s \rightarrow d\ [\ [a_{s616}b]_3\ [a_{s616}c]_4\ [\ ]_5\ [\ ]_6]\ _s$, where $b, c, d, a_{s616} \in V$.



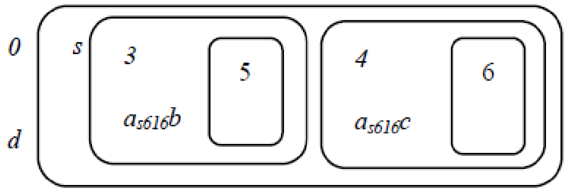
6) Apply mendo rule on membranes 5 and 3

R22: $d\ [\ [a_{s616}\ b]_3[a_{s616}\ c]_4\ [\ ]_5\ [\ ]_6]\ _s \rightarrow d\ [\ [a_{s616}\ b\ [\ ]_5\ ]_3[a_{s616}\ c]_4\ [\ ]_6]\ _s$, *where* $b, c, d, a_{s616}\ \in V$.



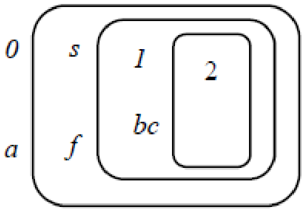
7) Apply mendo rule on membranes 6 and 4

R23: $d\ [\ [a_{s616}\ b\ [\ ]_5\ ]_3[a_{s616}\ c]_4\ [\ ]_6]\ _s \rightarrow d\ [\ [a_{s616}\ \ b\ [\ ]_5\ ]_3[a_{s616}\ c\ [\ ]_6\ ]_4]\ _s$, where $b, c, d, a_{s616} \in V$.



Final Configuration of Mode3

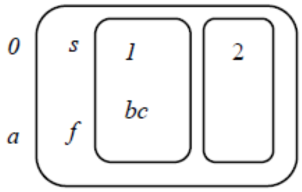
Rules of mode 4:



Initial Configuration of mode 4

1) Apply mexo rule on membranes 2 and 1.

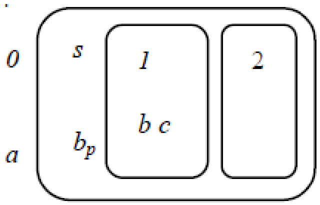
R24: $a\ [\ f\ \ [b\ c\ [\ ]_2\ ]_1]_s \rightarrow a\ [\ f\ \ [b\ c]_1\ [\ ]_2\ ]_s$, *where a, b, c, f ∈V.*



2) Apply object evolution rule, $[\ f \rightarrow b_p]$.

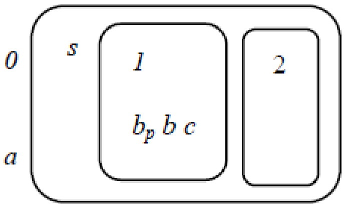
R25: $a\ [\ f\ [b\ c]_1\ [\ ]_2]\ _s \rightarrow a\ [\ b_p\ [b\ c]_1\ [\ ]_2]_s$, *where a, b, c, f, b_p∈ V.*

.



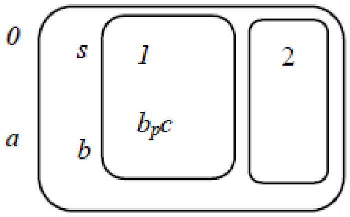
3) Apply communication rule; b_p object is sent in membrane 1.

R26: $a\ [b_p\ [b\ c]_1\ [\ ]_2\ ]_s \rightarrow\ \ a\ [\ [b_p b\ c]_1\ [\ ]_2\ ]_s$, *where a, b, c, b_p ∈V.*



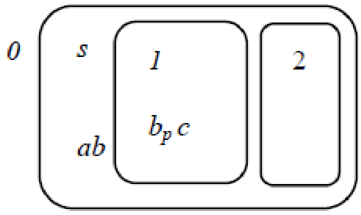
4) Apply communication rule; b object is sent out from membrane1.

R27: $a\ [\ [b_p\ b\ c]_1\ [\ ]_2\ ]_s \rightarrow a[b\ [b_p c]_1\ [\ ]_2\ ]_s$, *where a, b, c, b_p∈ V.*



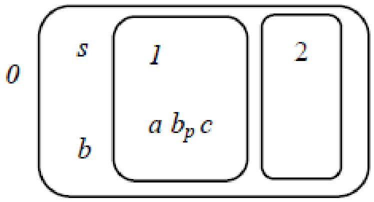
5) Apply communication rule; a object is sent in skin membrane.

R28: $a\ [b\ [b_p\ c]_1\ [\ ]_2]\ _s \rightarrow [a\ b\ [b_p c]_1\ [\ ]_2]\ _s$, *where a, b, c, b_p ∈V.*



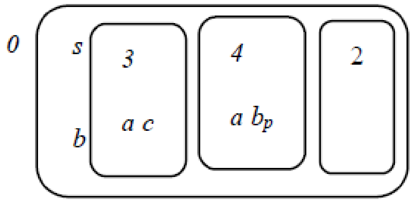
6) Apply communication rule; a object is sent in membrane 1.

R29: $[a\ b\ [b_p\ c]_1\ [\ ]_2]\ _s \rightarrow [b\ [a\ b_p\ c]_1\ [\ ]_2]\ _s$, where a, b, c, b_p ∈V.



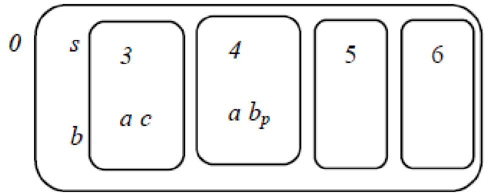
7) Apply division rule on membrane 1 (the different membranes labels 3, 4 are results).

R30: $[b\ [a\ b_p\ c]_1\ [\ ]_2]\ _s\ \rightarrow [b\ [a\ c]_3\ [a\ b_p\ ]_4[\ ]_2]_s$, *where a, b, c, b_p ∈V.*



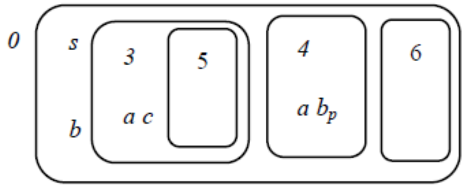
8) Apply division rule on membrane 2 (the different membranes labels 5, 6 are results).

R31: $[\ b\ [a\ c]_3\ [a\ b_p\ ]_4[\ ]_2]\ _s \rightarrow [b\ [a\ c]_3\ [a\ b_p\ ]_4[\ ]_5[\ ]_6]\ _s$, *where a, b, c, b_p ∈V.*



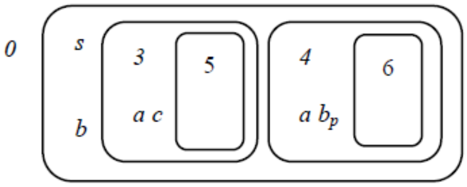
9) Apply mendo rule on membranes 5 and 3.

R32: $[\ b\ [a\ c]_3\ [a\ b_p\ ]_4 [\ ]_5 [\ ]_6]_s \rightarrow [b\ [a\ c\ [\ ]_5]_3\ [a\ b_p\ ]_4 [\ ]_6]_s$, *where* $a, b, c, b_p \in V$.



10)Apply mendo rule on membranes 6 and 4.

R33: $[b\ [a\ c[\ ]_5]_3\ [a\ b_p\ ]_4 [\ ]_6\ ]_s \rightarrow [b\ [a\ c[\ ]_5]_3\ [a\ b_p[\ ]_6\ ]_4]_s$, *where* $a, c, b_p \in V$.



Final Configuration of Mode 4

C. Computational Properties of Proposed Model

There are several features of our model and are really appropriate for many problems (solvable) that will be discussed:

1) Distribution: the behavior of interactive systems in the mitochondrial model depends on nonlinearly results from protein composition to get a new configuration.

2) Algorithmically: Our computability model is defined by active and mobile variants, which deal with the Turing machine's computational power or other classic representations of algorithms, making full computational models as decidability devices and efficient algorithms to solve NP-complete problems in polynomial time easy to simulate (and exponential space).

3) Transparency: the rules applied in this model are nothing else than reaction equations as illustrated in the previous subsection, biological fusion, and fission models without any mysterious notation and mysterious behavior.

4) Non-determinism: our model is viewed as a collection of instructions/rules, with the sole structure being that imposed by membrane localization, but structure inside each membrane consisting of rigid sequences of instructions of programs written in common programming languages.

V. Efficient Solutions to Different Classes of Problems using Proposed Model

A. Computational Complexity Classes with Proposed Model

The types of rules used in the proposed model are shown in the first column of Table II, while the applied rules for each type are listed in the second column. The basic classes of computational complexity problems have been shown in the second row are matched to the set of rules in the third column presenting solvability of each rule type.

TABLE II. Computational Power of Rules Recognizer in Proposed Model

Rules Types	Rules Applied in our Models	Classes of Problems		
		P	**NP**	**PSPACE**
Evolution rule (a)	R8, R10, R18, R25	√	*	√
In-Communication rule (b)	R11, R19, R26, R28, R29	-	√	√
Out-Communication rule (c)	R9, R12, R27	-	√	√
Membranes Division rule (d)	R13, R14, R20, R21, R30, R31	*	√	√

√ denotes allowed types of rules; * denotes disallowed types and - Rules with no impact on the computational power

Details of the notations P, NP, and PSPACE with definitions are presented in [48]. P is referred to Problems that had been solved in polynomial time with the evolution and rewriting rules for only one membrane [18, 49, and 50]. For instance, the ranking problem for three numbers had been solved by conventional computers. It has a computational power with P systems with an evolution rule. SUBSET SUM problem is belonging to a class of NP-complete problems that have been solved by division rule for elementary membranes with communication rules. In [51], the efficiency proof of its solution is demonstrated in uniform linear time. The QSAT problem is belonging to a class of PSPACE problems that are called "second class computers". QSAT has been solved by different families of active membranes. PSPACE class has been confirmed as an upper bound of the computations in polynomial time with P systems in [41].

Briefly, Table II is the summary of computational complexity power of rules that belong to active variant and used in this model. Presenting accepted classes of problems in polynomial time. The rules of types from (b, c, and d) can solve NP-complete problems in polynomial time as shown in the fifth column in Table II. Where the division rules for elementary membranes are unrestricted which leads to an increase of computational power [52, 53]. While the rules of types from (a) to (d) are suitable to address PSPACE problems in polynomial time. In [54] highlighted the role of different restrictions/extensions of evolution, communication rules.

B. Solving SAT Problem using MDM

- The Satisfiability of a propositional logic formula in conjunctive normal form is known as the SAT problem. It is classified as a strong problem in categories of NP-complete problems.
  - For any instance of SAT, Consider the following expression:

    $\beta = C_1 \wedge C_2 \wedge ... \wedge C_m$ over $X = \{x_1, x_2, ..., x_n\}$ such x_i ,where $1 \leq i \leq n$. In this case, (n = 3).
  - For each C_i, $1 \leq i \leq m$. In this case (m=3), a disjunction of the form $C_i = y_1 \vee y_2 \vee ... \vee y_r$ ($r \leq n$), where each y_j indicates to variable x_k or negation of x_k, $\bar{x}_k$

- The working over the alphabet: V = {$z, \bar{z}, q, \bar{q}, g, g_0, \gamma, \bar{\gamma}$, yes, no, $b_i, t_i, f_i, \gamma_i, \bar{\gamma}_i| 1 \leq i \leq n$}
- We construct the system by use the fission model configuration that has MDM rules to generate all potential variables assignments, $\{x_1, . . ., xn\}$. By rules of MDM to solve the SAT problem as follows:

1) $[b_i]_s \rightarrow [t_i\ b_{i+1}]_s\ [f_i\ b_{i+1}]_s$, for $1 \leq i \leq n - 1$ (division rule)

$[b_n]_s \rightarrow [t_n\ \gamma_1]_s\ [f_n\ \gamma_1]_s$ (division rule)

$[g]_1 \rightarrow [\ ]_1[\ ]_1$ (division rule)

$[g_0]_1 \rightarrow [\bar{\gamma}_1]_1\ [\bar{\gamma}_1]_1$ (division rule)

- $_2{}^{\boldsymbol{n}}$ membranes are created from the first two rules and labeled by ***s*** contain all potential variables assignments, $\{x_1, . . ., x_n\}$. Each membrane labeled by ***s*** is assigned by object $\boldsymbol{\gamma_1}$.
- $_2{}^{\boldsymbol{n}}$ membranes labeled by 1 are created from the following two rules. Each one of them contains object $\bar{\gamma}_1$. The correct assignments for C_1 are determined by using symbols γ_1 and $\bar{\gamma}_1$ in two steps.

2) $[t_j\gamma_i]_s[\bar{\gamma}_i]_1 \rightarrow [[t_j\gamma_i]_s\,\bar{\gamma}_i]_1$ (mendo)

$[[tj\gamma i]s\bar{\gamma}i]1 \rightarrow [tj\ \gamma i+1]s[\bar{\gamma}i+1]1$ (mexo)

(in case Ci includes the literal xj)

$[fj\gamma i]s[\bar{\gamma}i]1 \rightarrow [[fj\gamma i]s\bar{\gamma}i]1$ (mendo)

$[[fj\gamma i]s\bar{\gamma}i]1 \rightarrow [fj\ \gamma i+1]s[\bar{\gamma}i+1]1$ (mexo)

(in case Ci includes the literal $\neg$xj)

$[tj\gamma m]s[\bar{\gamma}m]1 \rightarrow [[tj\gamma m]s\bar{\gamma}m]1$ (mendo)

$[[tj\gamma m]s\ \bar{\gamma}m]1 \rightarrow [tj\ z]s[\bar{\gamma}m]1$ (mexo)

(in case Cm includes the literal xj)

$[fj\gamma m]s[\bar{\gamma}m]1 \rightarrow [[fj\gamma m]s\bar{\gamma}m]1$ (mendo)

$[[fj\gamma m]s\ \bar{\gamma}m]1 \rightarrow [fj\ z]s[\bar{\gamma}m]1$ (mexo)

(in case Cm includes the literal $\neg$xj)

The objects $\boldsymbol{\gamma_i}$ from the identical membrane ***s*** are exchanged by $\boldsymbol{\gamma_{i+1}}$ if the assignments fulfilled the clause $\boldsymbol{C_i}$. The assignments from the membranes which contain $\boldsymbol{\gamma_{i+1}}$ satisfy the clauses $\boldsymbol{C_1,\ldots,C_i}$, then, the object $\boldsymbol{\gamma_{i+1}}$ indicate the result in next step the clause $\boldsymbol{C_{i+1}}$ is checked. If all clauses are satisfied, then the membranes which have these assignments will be marked by object $\boldsymbol{\bar{z}}$.

3) $[c_i\ \gamma]_2[\bar{\gamma}]_3 \rightarrow [[c_{i+1}\ \gamma]_2\bar{\gamma}]_3$ (mendo)

$[[ci\ \gamma]2\bar{\gamma}]3 \rightarrow [ci+1\gamma]2\ [\bar{\gamma}]3$ (mexo)

$[[cn+2m+1\gamma]2\ \bar{\gamma}]3 \rightarrow [d\ \gamma]2[\bar{\gamma}]3$ (mexo)

$[cn+2m+1\ \gamma]2[\bar{\gamma}]3 \rightarrow [[cn+2m+1\ \gamma]2\ \bar{\gamma}]3$ (mendo)

The mentioned rules in (*iii*) are used to determine the number of steps is applied in rules of type (*ii*). If the number of steps is larger than $n + 2m + 1$, this is indicated to clauses are not satisfy. Then an object ***q*** is generated, that will create an object ***no***. The number $n + 2m + 1$ correspond to the following steps: generating space in n steps, validate assignments in $2m$ steps, generating a ***yes*** object in one step. Additionally, step can be applied; membrane *2* including the object $\boldsymbol{c_{n+2m+1}}$ becomes sibling with membrane *3*, thus increasing the number of steps needed to generate ***d*** object to $n + 2m + 2$.

4) iv. $[\bar{z}]_s[z]_4 \rightarrow [[yes]_4]_s$ (mendo)

$[q]_3[\bar{q}]_4 \rightarrow [[no]_3]_4$ (mendo).

- When membrane 4 enters membrane s in (2m+n+1) step, a yes object is created. But a no object is created when no membrane s contains an object q. When membrane 3 enters membrane 4, one of these two rules cannot be applied anymore. Finally, the computation result is either a yes or a no object in the system.
- The size of the working alphabet is 4n+4m+ 13. The number of used rules is computed as follows: the number of rules from type (i) is n + 2, the number of rules from type (ii) is 4nm, the number of rules from type (iii) is n + 2m + 3 and the number of rules from type (iv) is 2 rules.
- As a result, the size of the computing system to solve SAT is O(mn). Because n + 2m is an odd number, the computation ends in n + 2m + 3. As a result, we had to do an extra step before obtaining q object from c_{n+2m+1}. If n+2m is an even integer, then after n + 2m+ 2 steps, q object is generated.

VI. Representation Model using UML

Different diagrams of UML are used to analyze the behavior of mitochondria fusion-fission models. The first diagram represents the modality of normal mitochondria organelles in the living cell using an object diagram as shown in Fig. 5.

Fig. 5 revealed that the object diagram is combined from a superclass titled cell that is considered as “environment” in P systems. The super class is consisting of many objects that represent components of a living cell that contains at least two mitochondria. Our proposed design is interested in mitochondria object which is a sub class from the superclass. The mitochondria object has some attributes such as DNA, RNA, and the mitochondria functional operations. The basic operations of mitochondria objects are fusion, fission, motility, and mitophagy, while in the current study, the authors focused on the fusion and fission operations. The outer and inner membranes objects are composited from mitochondria objects, where each mitochondrion has one outer membrane and one inner membrane, and all of them, have some attributes that are represented by proteins that involved in the fusion and fission processes.

Fig. 5 revealed that the object diagram is combined from a superclass titled cell that is considered as “environment” in P systems. The super class is consisting of many objects that represent components of a living cell that contains at least two mitochondria. Our proposed design is interested in

mitochondria object which is a sub class from the superclass. The mitochondria object has some attributes such as DNA, RNA, and the mitochondria functional operations. The basic operations of mitochondria objects are fusion, fission, motility, and mitophagy, while in the current study, the authors focused on the fusion and fission operations. The outer and inner membranes objects are composited from mitochondria objects, where each mitochondrion has one outer membrane and one inner membrane, and all of them, have some attributes that are represented by proteins that involved in the fusion and fission processes.

The diagram is illustrated in Fig. 6, the activity diagram and represents the interactions that occurred in two mitochondria objects to keep their metabolic regulation using fusion operations. The events that occurred to complete fusion operations of the two normal mitochondria organelles were analyzed in Fig. 6. First, both outer membranes object with their contents such as proteins, enzymes, and ATP molecules are merged by chemical interactions to complete the binding process owing to the formation of a new outer membrane object. Second, both inner membranes object was merged using other chemical operations involved in the binding process followed by the production of a new inner membrane object. Lastly, both objects are joined producing new outer and new inner membranes leading to the production of new mitochondrion object which contains inherits behaviors from the original one. In Fig. 7, actions and events are analyzed to complete fission operations to the one normal mitochondrion organelle. First, the object of Drp1 protein is called to mitochondrion object to interact with proteins in the outer membrane, then one action from four actions will be activated the fission operation according to the required signal. Finally, the mitochondrion divided into two mitochondria organelles.

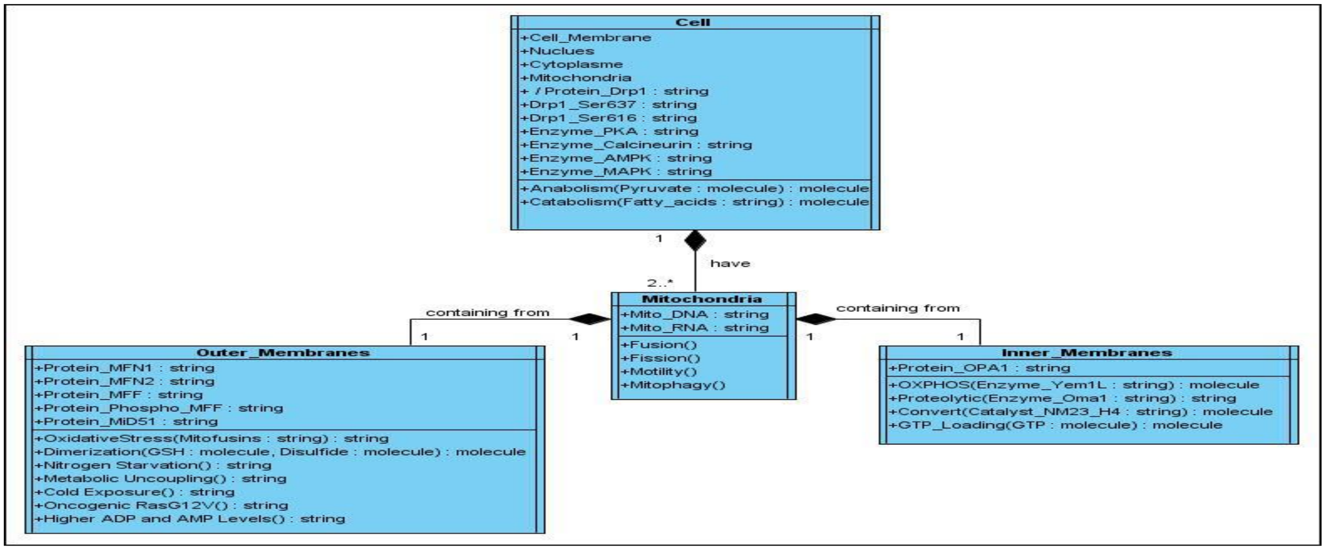

Cell
+Cell_Membrane
+Nuclues
+Cytoplasme
+Mitochondria
+ / Protein_Drp1 : string
+Drp1_Ser637 : string
+Drp1_Ser616 : string
+Enzyme_PKA : string
+Enzyme_Calcineurin : string
+Enzyme_AMPK : string
+Enzyme_MAPK : string
+Anabolism(Pyruvate : molecule) : molecule
+Catabolism(Fatty_acids : string) : molecule
1
have
2..*
Mitochondria
+Mito_DNA : string
+Mito_RNA : string
+Fusion()
+Fission()
+Motility()
+Mitophagy()
containing from
1
1
containing from
1
1
Outer_Membranes
+Protein_MFN1 : string
+Protein_MFN2 : string
+Protein_MFF : string
+Protein_Phospho_MFF : string
+Protein_MiD51 : string
+OxidativeStress(Mitofusins : string) : string
+Dimerization(GSH : molecule, Disulfide : molecule) : molecule
+Nitrogen Starvation() : string
+Metabolic Uncoupling() : string
+Cold Exposure() : string
+Oncogenic RasG12V() : string
+Higher ADP and AMP Levels() : string
Inner_Membranes
+Protein_OPA1 : string
+OXPHOS(Enzyme_Yem1L : string) : molecule
+Proteolytic(Enzyme_Oma1 : string) : string
+Convert(Catalyst_NM23_H4 : string) : molecule
+GTP_Loading(GTP : molecule) : molecule

Fig. 5. Object Diagram of Mitochondria Components.

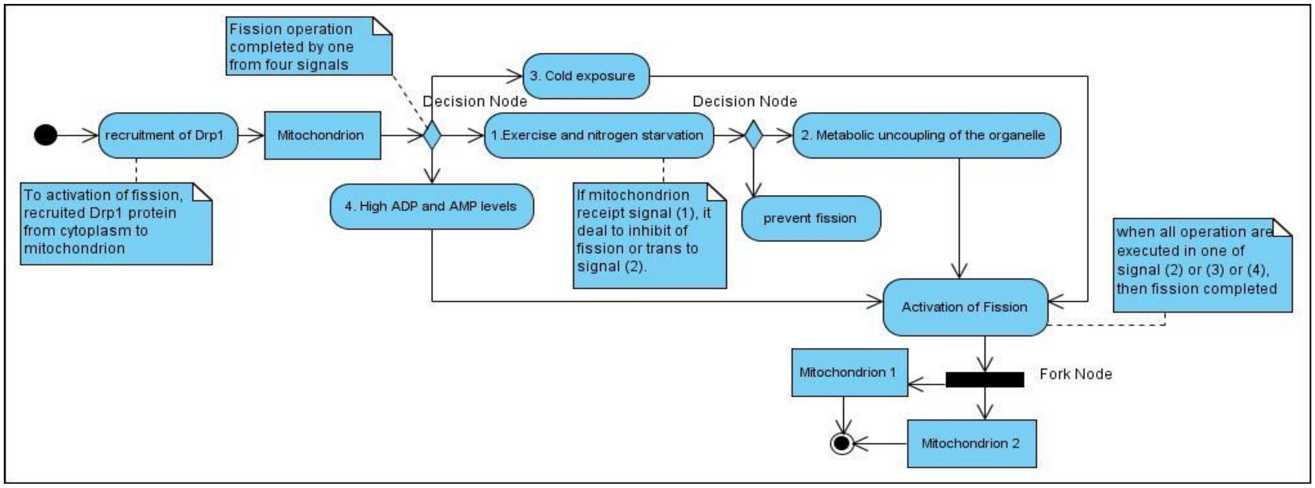

Fission operation completed by one from four signals
recruitment of Drp1
Mitochondrion
Decision Node
3. Cold exposure
1.Exercise and nitrogen starvation
Decision Node
2. Metabolic uncoupling of the organelle
To activation of fission, recruited Drp1 protein from cytoplasm to mitochondrion
4. High ADP and AMP levels
If mitochondrion receipt signal (1), it deal to inhibit of fission or trans to signal (2).
prevent fission
when all operation are executed in one of signal (2) or (3) or (4), then fission completed
Activation of Fission
Mitochondrion 1
Fork Node
Mitochondrion 2

Fig. 6. Activity Diagram for Actions Fusion Model.

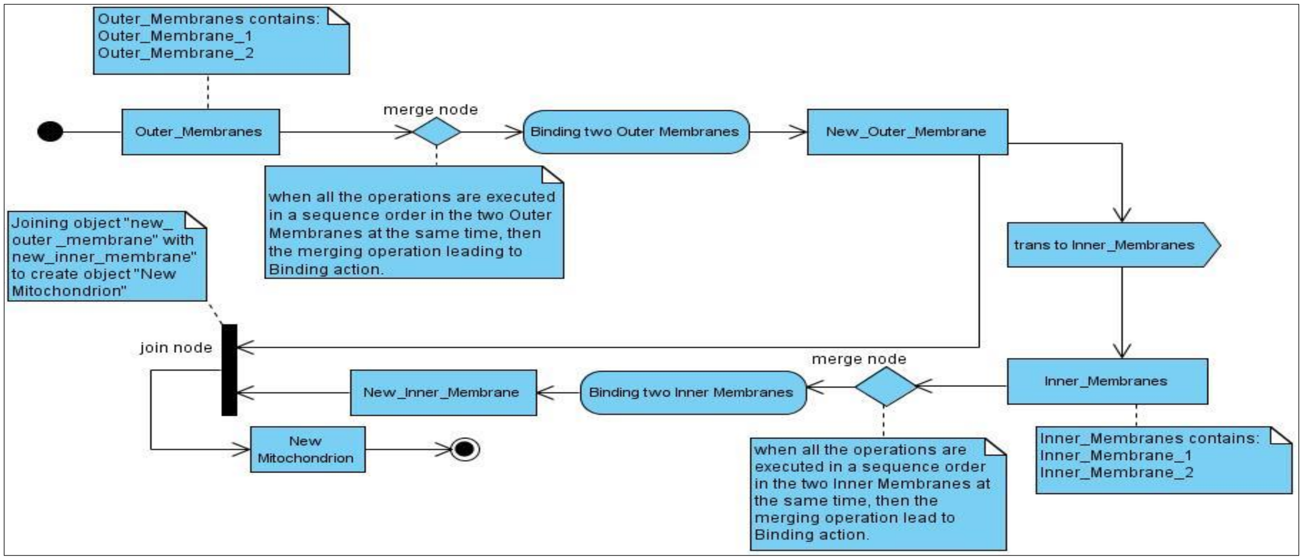

Outer_Membranes contains:
Outer_Membrane_1
Outer_Membrane_2
Outer_Membranes
merge node
Binding two Outer Membranes
New_Outer_Membrane
when all the operations are executed in a sequence order in the two Outer Membranes at the same time, then the merging operation leading to Binding action.
Joining object "new_ outer _membrane" with new_inner_membrane" to create object "New Mitochondrion"
trans to Inner_Membranes
join node
merge node
New_Inner_Membrane
Binding two Inner Membranes
Inner_Membranes
New Mitochondrion
when all the operations are executed in a sequence order in the two Inner Membranes at the same time, then the merging operation lead to Binding action.
Inner_Membranes contains:
Inner_Membrane_1
Inner_Membrane_2

Fig. 7. Activity Diagram for Actions Fission Model.

VII. Conclusion and Future Work

P systems can be used to model biological and physiological processes. This paper proposed a model of two actual biological functions of important organelle in a living cell, mitochondria fusion, and fission. This model, MDM, combined a set of rules from different variants of P systems, division rule with mexo and mendo rules that provided higher efficiency in terms of space complexity rather than using rules of active membranes only. The conclusion of this work can be summarized in terms of three aspects. First, P is a computational power set of evolution rules in polynomial time. While a set of PSPACE problems are solved in polynomial time using polynomial uniform of active membranes families with the operation of elementary membrane division. This is one of the most well-studied methods for getting an exponential working space in order to exchange space for time and solve computationally difficult problems (commonly NP-complete problems) in a timely manner (typically polynomial or even linear). Second, this research introduced a semi-uniform linear time solution to SAT problem using MDM rules with 3 membranes. From the results, we proofed that only three membranes suffice in solving the SAT problem. MDM is faster than other models that used 9 membranes to obtain computational universality. Finally, using biological models and development of their object relationship diagrams have been implemented using object-oriented programming (OOP) to be used as an analysis tool for complex data of mitochondria neurodegenerative diseases.

Future research topics may include:

1) Understanding of the remaining two operations of mitochondrial regulation (motility and mitophagy) and simulated them by P systems variants.

2) Designing a complete model for metabolic regulation of mitochondrial dynamics.

3) By different kinds of cell-like P systems.

4) Complete understanding of physiological processes for studying different diseases and malfunctions involved in human neurodegenerative system mediated by irregular mitochondria.

5) Investigating and studying other biological operations for modeling other echo systems using different P systems variants.

6) We hope to define and develop a new strategy in **OOP** using P systems with active and dynamic rules to solve nondeterministic problems in polynomial time specifically in Machine Learning algorithms.

References

[1] Paun G, Rozenberg G, Salomaa A. DNA computing: new computing paradigms. Springer Science & Business Media; 2005 Feb 4.

[2] Román G. Inference of bounded L systems with polymorphic P systems. Journal of Membrane Computing. 2019 Mar;1(1):52-7.

[3] Ambainis A, Yakaryılmaz A. Automata and quantum computing. arXiv preprint arXiv:1507.01988. 2015 Jul 7.

[4] Haldurai L, Madhubala T, Rajalakshmi R. A study on genetic algorithm and its applications. International Journal of Computer Sciences and Engineering. 2016 Oct;4(10):139.

[5] Albawi S, Mohammed TA, Al-Zawi S. Understanding of a convolutional neural network. In2017 International Conference on Engineering and Technology (ICET) 2017 Aug 21 (pp. 1-6). IEEE.

[6] Sempere JM. Modeling of decision trees through P systems. New Generation Computing. 2019 Sep;37(3):325-37.

[7] Ciobanu G, Păun G. Applications of membrane computing. Pérez-Jiménez MJ, editor. Berlin: Springer; 2006 Oct.

[8] Zhang, Gexiang, et al. "An overview of hardware implementation of membrane computing models." ACM Computing Surveys (CSUR) 53.4 (2020): 1-38.

[9] Song, B., Li, K., Orellana-Martín, D., Valencia-Cabrera, L., & Pérez-Jiménez, M. J. (2020). Cell-like P systems with evolutional symport/antiport rules and membrane creation. Information and Computation, 275, 104542.

[10] Păun G. Languages in membrane computing: some details for spiking neural p systems. In International Conference on Developments in Language Theory 2006 Jun 26 (pp. 20-35). Springer, Berlin, Heidelberg.

[11] Freund R, Păun G. On the number of non-terminal symbols in graph-controlled, programmed and matrix grammars. InInternational

Conference on Machines, Computations, and Universality 2001 May 23 (pp. 214-225). Springer, Berlin, Heidelberg.

[12] Jiménez MJ, Jiménez ÁR, Caparrini FS. Complexity classes in models of cellular computing with membranes. Natural Computing. 2003 Sep 1;2(3):265-85.

[13] García-Quismondo M, Graciani C, Riscos-Núñez A. Membrane computing as a modelling tool: looking back and forward from Sevilla. InEnjoying Natural Computing 2018 (pp. 114-129). Springer, Cham.

[14] Macías ‐ ramos LF, Song B, Valencia ‐ Cabrera L, Pan L, Pérez ‐ jiménez MJ. Membrane fission: A computational complexity perspective. Complexity. 2016 Jul;21(6):321-34.

[15] Orellana-Martín D, Valencia-Cabrera L, Riscos-Núñez A, Pérez-Jiménez MJ. Minimal cooperation as a way to achieve the efficiency in cell-like membrane systems. Journal of Membrane Computing. 2019 Jun 1;1(2):85-92.

[16] Aman B, Ciobanu G. Mobile membranes: Computability and complexity. InInternational Colloquium on Theoretical Aspects of Computing 2013 Sep 4 (pp. 59-75). Springer, Berlin, Heidelberg.

[17] Alexiou AT, Psiha MM, Rekkas JA, Vlamos PM. A stochastic approach of mitochondrial dynamics. World AcadSciEng Technol. 2011 Jul 29;55:77-80.

[18] Aman B, Ciobanu G. Mobile Membranes. IEEE Access. 2020 Jul 24;8:147439-50.

[19] Krishna SN, Păun G. P systems with mobile membranes. Natural Computing. 2005 Sep 1;4(3):255-74.

[20] Macías-Ramos LF, Pérez-Jiménez MJ, Riscos-Núñez A, Valencia-Cabrera L. Membrane fission versus cell division: When membrane proliferation is not enough. Theoretical Computer Science. 2015 Dec 10; 608:57-65.

[21] Alhazov, A., & Pan, L. Trading Polarizations for Labels in P Systems with Active Membranes, submitted, 2004, 7(1), 141:159.

[22] Alhazov A, Pan L, Păun G. Trading polarizations for labels in P systems with active membranes. ActaInformatica. 2004 Dec 1;41(2-3):111-44.

[23] Mishra P, Chan DC. Metabolic regulation of mitochondrial dynamics. Journal of Cell Biology. 2016 Feb 15;212(4):379-87.

[24] Păun G. Computing with membranes. Journal of Computer and System Sciences. 2000 Aug 1;61(1):108-43.

[25] Păun A. On P systems with active membranes. InUnconventional Models of Computation, UMC'2K 2001 (pp. 187-201). Springer, London.

[26] Păun G. Computing with membranes: Attacking NP-complete problems. In Unconventional models of Computation, UMC'2K 2001 (pp. 94-115). Springer, London.

[27] Martín-Vide C, Păun G, Pazos J, Rodríguez-Patón A. Tissue P systems. Theoretical Computer Science. 2003 Mar 8;296(2):295-326.

[28] Păun G. Computing with membranes: Attacking NP-complete problems. In Unconventional models of Computation, UMC'2K 2001 (pp. 94-115). Springer, London.

[29] Margenstern M, Martin-Vide C, Paˇun G (2002) Computing with membranes: variants with an enhanced membrane handling. DNA computing. Springer, New York, pp 340–349

[30] Wang T, Wang J, Ming J, Sun Z, Wei C, Lu C, Pérez-Jiménez MJ. Application of neural-like P systems with state values for power coordination of photovoltaic/battery microgrids. IEEE Access. 2018 Aug 13;6:46630-42.

[31] Paˇun G, Pe´rez-Jime´nez MJ et al (2012) Languages and P systems: recent developments. ComputSci 20(2):59

[32] Barbuti R, Maggiolo-Schettini A, Milazzo P, Pardini G, Tesei L (2011) Spatial P systems. Nat Comput 10(1):3–16

[33] Giannakis K, Andronikos T. Membrane automata for modeling biomolecular processes. Natural Computing. 2017 Mar 1;16(1):151-63.

[34] Peng H, Wang J, Ming J, Shi P, Pérez-Jiménez MJ, Yu W, Tao C. Fault diagnosis of power systems using intuitionistic fuzzy spiking neural P systems. IEEE Transactions on Smart Grid. 2017 Feb 16;9(5):4777-84.

[35] Zhao J, Wang N. A bio-inspired algorithm based on membrane computing and its application to gasoline blending scheduling. Coputers & chemical engineering. 2011 Feb 9;35(2):272-83.

[36] Peng H, Wang J, Shi P, Riscos-Núñez A, Pérez-Jiménez MJ. An automatic clustering algorithm inspired by membrane computing. Pattern Recognition Letters. 2015 Dec 15;68:34-40.

[37] Cooper J, Nicolescu R. The Hamiltonian cycle and travelling salesman problems in cP systems. FundamentaInformaticae. 2019 Jan 1;164(2-3):157-80.

[38] Zandron C, Ferretti C, Mauri G. Solving NP-Complete Problems Using P Systems. In Unconventional Models of Computation, UMC'2K: Proceedings of the Second International Conference on Unconventional Models of Computation,(UMC'2K) 2012 Dec 6 (p. 289). Springer Science & Business Media.

[39] Gutiérrez–Naranjo MA, Pérez–Jiménez MJ, Riscos–Núñez A, Romero–Campero FJ. On the power of dissolution in P systems with active membranes. InInternational workshop on membrane computing 2005 Jun 18 (pp. 224-240). Springer, Berlin, Heidelberg.

[40] Pan L, Ishdorj TO. P Systems with Active Membranes and Separation Rules. J. UCS. 2004 May 28;10(5):630-49.

[41] Sosík P, Păun A, Rodríguez-Patón A. P systems with proteins on membranes characterize PSPACE. Theoretical Computer Science. 2013 Jun 3;488:78-95.

[42] Song B, Pérez-Jiménez MJ, Pan L. An efficient time-free solution to QSAT problem using P systems with proteins on membranes. Information and Computation. 2017 Oct 1;256:287-99.

[43] Aman B, Ciobanu G. Turing completeness using three mobile membranes. InInternational Conference on Unconventional Computation 2009 Sep 7 (pp. 42-55). Springer, Berlin, Heidelberg.

[44] Krishna SN. The power of mobility: Four membranes suffice. InConference on Computability in Europe 2005 Jun 8 (pp. 242-251). Springer, Berlin, Heidelberg.

[45] Csuhaj-Varjú E, Vaszil G. On the power of P automata. InInternational Conference on Unconventional Computing and Natural Computation 2013 Jul 1 (pp. 55-66). Springer, Berlin, Heidelberg.

[46] Cardelli L, Paun G. An universality result for a (mem) brane calculus based on mate/drip operations. International Journal of Foundations of Computer Science. 2006 Feb;17(01):49-68.

[47] Mishra P, Chan DC. Metabolic regulation of mitochondrial dynamics. Journal of Cell Biology. 2016 Feb 15;212(4):379-87.

[48] Valiant LG, van Leeuwen J. Handbook of theoretical computer science. InAlgorithms and Complexity, chapter General Purpose Parallel Architectures 1990 (pp. 943-971). Elsevier Science.

[49] Mauri G, Paun G, Pérez-Jiménez MJ, Rozenberg G, Salomaa A, editors. Membrane Computing: 5th International Workshop, WMC 2004, Milan, Italy, June 14-16, 2004, Revised Selected and Invited Papers. Springer Science & Business Media; 2005 Mar 7.

[50] Gheorgue M, Paun G, Pérez Jiménez MD. Frontiers of membrane computing: Open problems and research topics. Proceedings of the Tenth Brainstorming Week on Membrane Computing, 171-250. Sevilla, ETS de IngenieríaInformática, 30 de Enero-3 de Febrero, 2012.

[51] Song B, Pérez-Jiménez MJ, Pan L. Efficient solutions to hard computational problems by P systems with symport/antiport rules and membrane division. BioSystems. 2015 Apr 1;130:51-8.

[52] Leporati A, Manzoni L, Mauri G, Porreca AE, Zandron C. Simulating elementary active membranes. InInternational Conference on Membrane Computing 2014 Aug 20 (pp. 284-299). Springer, Cham.

[53] Zandron C, Leporati A, Ferretti C, Mauri G, Pérez-Jiménez MJ. On the computational efficiency of polarizationless recognizer P systems with strong division and dissolution. Fundamenta Informaticae. 2008 Jan 1;87(1):79-91.

[54] Song B, Song T, Pan L. Time-free solution to SAT problem by P systems with active membranes and standard cell division rules. Natural Computing. 2015 Dec;14(4):673-81.