International Journal of Intelligent Information Systems
2017; 6(3): 25-35
http://www.sciencepublishinggroup.com/j/ijiis
doi: 10.11648/j.ijiis.20170603.11
ISSN: 2328-7675 (Print); ISSN: 2328-7683 (Online)

A Novel Approach of the Shortest Path Problem Using P System

Einallah Salehi

Department of Computer Science, Malayer Branch, Islamic Azad University, Malayer, Iran

Email address:
Salehi.utm@hotmail.com

To cite this article:
Einallah Salehi. A Novel Approach of the Shortest Path Problem Using P System. *International Journal of Intelligent Information Systems.* Vol. 6, No. 3, 2017, pp. 25-35. doi: 10.11648/j.ijiis.20170603.11

Received: June 9, 2017; **Accepted**: June 30, 2017; **Published**: July 20, 2017

Abstract: Membrane Computing is inspired from biological cell activities as a new distributed parallel computational framework, which can be used for decreasing the time complexity of problems with high execution time. Since the usual way to reduce the time complexity of Artificial Intelligence (AI) problems is using parallel algorithms, Membrane Computing can be extensively applied. On the other hand, shortest path problem (SPP) is the most broadly method to solve the problems in AI. There have been a variety of algorithms presented, that could find a solution in a desirable time but usually are not accurate, or they are accurate but they are too slow. In Membrane Computing technique, the normal way for reducing the time complexity is using P system with active membranes that the number of membranes increases during the computation; thus, the time is traded against the space. This paper presents the first Membrane Computing technique for solving the SPP using P system with membranes division by a breadth first exploring on a grid. The theorems show the run times for breadth-first search and SPP are significantly reduced.

Keywords: Breadth-First Search, Division Rule, P System, Shortest Path Problem, Time Complexity

1. Introduction

One of the areas of the study in computer science is Membrane Computing that has been proposed in [1]. Membrane Computing is a framework for the distributed parallel computing. This framework of computation has been derived from the views, the configurations, models and also living cell activities as well as the cells organized in a hierarchy. Through Membrane Computing, P system is gained and it is designed for this new parallel computing. In the initial configuration of a P system, there are several membranes that they are placed in the most outer membrane (skin). The membranes divide the Euclidean space into different and discrete regions that include several evolution rules and objects of alphabetical symbols. Through using these rules, it will become possible to change and/or transfer each object into the next region. Also the membranes could be dissolved, divided or created by some rules. The rules are usually applied in a maximally parallel and nondeterministic manner. The computation begins following the initial system configuration and it ends when it is not possible to apply any rule. The results of computation might be a multiset of objects which accumulated in output membrane or is released from the skin of the system.

There are different kinds of P systems such as P system with active membranes that, during the period of computation, through employing a number of division rules (membranes division), creation rules (membranes creation) or separation rules, the number of membranes are raised. The mentioned P systems are inspired from biological cell division or cell creation [2]. In P system with division rules, the membranes can be divided into k membranes that it is called P system with k-bounded division rules (2-bounded division rules called division rules) [3, 4]. Moreover, in P system with membranes creation, the new membranes will be created by applying creation rules on some objects [5]. As well as, P system with separation rules works such as P system with division rules respect to membranes division. P systems has been studied in different research which often many of them have reduced the time complexity from exponential to polynomial by making an exponential workspace, or at least the time complexity has been improved

by them [6-10].

The shortest path problem (SPP) is used in order to find a path possessing hold the minimum travel cost, from one or more start point to one or more goal in a linked network. Since the range of applications in transportations, robotics, computer and networks, is broad, this case is a significant issue. In order to solve the SPP different methods were suggested. In approximate and heuristic algorithms, the solution was found in a sufficient time, while they may not be accurate in this period of time. Even if the accuracy is high, they might not be executed fast enough. One of the algorithms that have been already designed and proposed is finding the shortest path using breadth-first search (BFS) algorithm.

In this paper, a new parallel approach for finding the shortest path between two vertices (origin and destination) is introduced in which P system with division rules is used. This P system is explored a directed grid with *m* rows and *n* columns of the nodes, which includes positive integer weights according to BFS. This method is capable to decrease the time complexity of BFS from $O(3mn)$ to $O[2(m+n)]$, and also reduces the time complexity of SPP from $O[mn(2L+1)]$ to $O[2L(m+n)]$ assuming L as the maximum size length of the edges.

The paper is organized as follows: In section 3, some basic definitions on P system with division rules are presented. Section 4 introduces the SPP using the BFS algorithm. The proposed method for SPP using BFS algorithm based on P system with membrane division is presented in section 5, in which some guidelines on the implementation of a P system with membrane division for SPP are presented using the BFS algorithm. In section 6, using three examples, experiments carried out in this study are presented. Finally, in section 7, conclusion and some ideas for future works are described.

2. Related Works

The first approaches on searching problems in the framework of Membrane Computing have been already done by Gutiérrez-Naranjo and Pérez-Jiménez [11, 12]. In these studies depth-first search (DFS) and local search have been investigated using transition P systems and speed of execution time for solving the N-Queen problem have been extremely increased in comparison to prior work [13]. Moreover, P systems were utilized for studying on DFS and BFS in [14-16] on disjoint paths. In these researches, P systems are used without division rules. The first study has been presented by applying well-known Membrane Computing techniques for implementation of BFS using P system with membrane division. In this study the BFS algorithm has been imposed on a random binary tree that the time complexity will be decreased from exponential to linear with regard to the depth of the tree [17].

3. P System with Membrane Division

P system with membranes division is a model which has been frequently studied on the Membrane Computing. This model is a well-known model in the P system community and one of the earliest models which emerged in 2001 [3].

Definition 1: A P system with division rules for elementary membranes (P system with active membrane using division rules for elementary membranes) is a construct of the form $\Pi = (\Gamma, I_M, I_N, O_M, O_N, H, \mu, W, R, i_{in}^{p_1}, i_{out}^{p_2})$, where:

(1) Γ denotes the alphabet of *symbol-objects*.

(2) $I_M \subseteq \Gamma$ is the *input alphabet*.

(3) I_N is the *numerical input*.

(4) $O_M \subseteq \Gamma$ is the *output alphabet*.

(5) O_N is the *numerical output*.

(6) H indicates a finite set of *labels* for membranes.

(7) μ stands for a membrane *structure*, consisting of finite membranes that labeled with elements of *H*.

(8) $W = \{m_h \mid h \in H\}$, where: $m_h \in \Gamma^*$ (Γ^* is the set of strings over Γ) describes the *initial multiset* of the objects placed in the membrane with label *h*.

(9) $i_{in}^{p_1}, p_1 \in \{+,-,0\}$, denotes the *input region* (membrane).

(10) $i_{out}^{p_2}, p_2 \in \{+,-,0\}$, determines the *output region* (membrane).

(11) *R* stands for a finite set of *rules* of the following forms:

(a) $[x \to u]_h^p$, for $h \in H;\ p \in \{+,-,0\};\ x \in \Gamma, u \in \Gamma^*$.

This is an object *evolution rule* that is accompanied with a membrane which is labeled with *h,* and it depends on the membrane polarity. The empty string is represented by $\lambda \in \Gamma^*$.

(b) $x[\ ]_h^{p_1} \to [y]_h^{p_2}$, for $h \in H, p_1, p_2 \in \{+,-,0\}, x, y \in \Gamma$.

An object is sent in from the immediate outside region of the membrane with the label *h*; at the same time, this object would be changed into another object, and, simultaneously, the membrane polarity can also be changed.

(c) $[x]_h^{p_1} \to y[\ ]_h^{p_2}$, for $h \in H, p_1, p_2 \in \{+,-,0\}, x, y \in \Gamma$.

Here, an object is sent out from the membrane with the label *h* to the immediate outside region. This object, at the same time, changes into another object, and, simultaneously, the membrane polarity can be changed.

(d) $[x]_h^p \to y$, for $h \in H;\ p \in \{+,-,0\};\ x, y \in \Gamma$.

A membrane labeled with *h* is dissolved in reaction with an object. The skin is never dissolved.

(e) $[x]_h^{p_1} \to [y]_h^{p_2}[z]_h^{p_3}$, for

$h \in H;\ p_1, p_2, p_3 \in \{+,-,0\};\ x, y, z \in \Gamma$.

A membrane by an object can be divided into two membranes with the same label, each of them containing one object (objects may be the same or different). And the polarities of these membranes can be altered.

The rules of a P system with membrane division are employed based on the following principles:

(1) All rules are used in a *parallel* (The priority for some rules could be determined) and in a *maximal* manner. At each

step, one object of a membrane could be utilized by only one rule that is chosen in a non deterministic way, but any object that is able to evolve by one rule of any form should do its evolution.

(2) With dissolution of a membrane, its contents (multiset and internal membranes) are left free in the surrounding region.

(3) All objects and membranes that have not been specified in a rule and those that have not evolved remain unchanged to the next step.

(4) Those rules that are associated with the membranes that are labeled with *h* are utilized for the all copies of this membrane. At one step, a membrane that is labeled with *h* could only be subjected to one rule of types (b)-(e), and rule (a) could also be applied with other rules. In this case, assumption is that the evolution rules of type (a) are used first, and after that, the other types of rules are applied. This process takes only one step to be performed.

(5) The skin membrane could not be divided. Like other membranes, it can have "electrical charge".

Note 1: In the above definition, some properties can be intended as follows;

(a). The non-necessary components of a P system can be omitted.

(b). Several inputs and outputs can be considered.

(c). The numerical inputs and outputs can be considered as a set of non-negative integer numbers, or vectors with non-negative integer components.

Definition 2: One says that rule r_i has priority (in the strong sense) over rule r_j; if r_i can be applied, then rule r_j cannot be applied. In this case one shows $r_i > r_j$.

Definition 3: Let i_{out}^{p} be an output membrane (region) of P system Π; one says that $i_{out}^{p_2}$ is a dynamic output if the P system rules can apply on $i_{out}^{p_2}$.

This definition allows finding the different solution (using division rule, more than one solution can be obtained) of the problem. Also the number of steps will decrease (there is no need to transfer the output alphabets).

Definition 4: The number of objects in string *u* is the length of the string, and it is denoted by $|u|$, also the number of object *a* in string *u* is denoted by $|u|_a$.

Remark 1: Some remarks are taken as follows:

(a). P system Π configuration in step *i* is determined by $C^i(\Pi)$ (in short C^i); also, $C^0(\Pi)$ and $C^{T_\Pi}(\Pi)$ are initial and final (halting) configurations, respectively.

(b). The execution time (steps) of P system Π from $C^i(\Pi)$ until final configuration is indicated by $T(C^i(\Pi))$, and therefore, $T(C^0(\Pi)) = T_\Pi$.

(c). P system Π with set of rules *R* is denoted by $\Pi|_R$.

Note 2: According to Remark 1(c):

$$T(C^0(\Pi)) = k + T(C^k(\Pi)),\ 0 \le k \le T_\Pi.$$

It should be noted that in a P system with membranes division, the number of membranes can be increased applying rules of type (e). Due to maximal parallelism, 2^n copies from the membranes of the same labels can be obtained after *n* steps. Therefore, hard problems such as NP-complete problems or problems with high time complexity can be solved in this framework in a polynomial or even linear time.

4. The Shortest Path Problem Based on Breadth-First Search Algorithm

The SPP deals to find the shortest distance between start and final vertices in a graph in which shown by a set of edges and nodes. The SPP is a most basic network optimization problem that in a plenty of network optimization algorithms, this problem may happen and be considered as sub problem. Different types of SPP algorithms are being introduced and improved constantly.

There are several algorithms to solve the SPP. One of the basic methods is the BFS algorithm, in which each level of the nodes will be expanded in one stage. Thus, if a node is placed in depth *n*, then it will be explored after *n* stages [18]. Hence, using breadth first can obtain the goal in the minimum number of stages. The approximate running time of BFS is $O(|V|+|E|)$, where $|V|$ is the number of vertices and $|E|$ is the number of edges, also the time complexity for SPP using BFS is $O(|V|+L|E|)$, in which *L* indicates the highest magnitude length of the edges [18, 19].

5. Proposed Method

This section provides the mentioned approach to solving the SPP based on BFS through P system with membranes division. This approach intends to prove that Membrane Computing can offer components required for finding a solution for any problem which as shown as a space of states; thus, it has a high capability to solve a wide range of the real-life problems. The aim of this approach is looking for a shortest path solution in the search space (weighted, directed grid) by using a class of P systems with membranes division according to the BFS.

5.1. Shortest Path Problem Formulation and Its Ingredients

In the present research, the SPP was executed based on BFS while P systems with membranes division were employed on a 2-dimentional directed grid with integer positive weights. Grids are an important restricted class of graphs which can reach a wide range of real-life problems using them.

Definition 5: A 2-dimentional directed grid is a connected graph with $m \times n$ vertices $v_{i,j}$ ($1 \le i \le m$, and $1 \le j \le n$), and with the horizontal edges directed to the right and the vertical edges directed to the bottom, such that each vertex $v_{i,j}$ is

connected to vertex $v_{i+1,j}$ ($1 \le i \le m-1$, and $1 \le j \le n$), or vertex $v_{i+1,j}$ ($1 \le i \le m$, and $1 \le j \le n-1$), or both of them.

A 2-dimentional directed grid graph is presented as $\vec{G}_{m\times n}$ and weighted graph $\vec{G}_{m\times n}$ is considered as case study. The goal is finding the shortest path from origin ($v_{1,1}$) to destination ($v_{m,n}$). In this case, length of each edge is considered weight of it. Also length of each path from origin to destination is the summation of whole weights of the edges in each path, and the shortest path is a path with minimum length among the existing paths from origin to destination.

Remark 2: Some remarks of the case study (weighted graph $\vec{G}_{m\times n}$) are considered as follows:

(a). Graph $\vec{G}_{m\times n}$ is weighted by a positive integer numbers.

(b). Weight of edge $(v_{i-1,j}, v_{i,j})$ is shown by $w_{i,j,1}$, where $2 \le i \le m$, and $1 \le j \le n$.

(c). Weight of edge $(v_{i,j-1}, v_{i,j})$ is shown by $w_{i,j,2}$, where $1 \le i \le m$, and $2 \le j \le n$.

(d). The connection between nodes $v_{i-1,j}$ and $v_{i,j}$ ($2 \le i \le m$, and $1 \le j \le n$) is considered by

$$y_{i,j,1}, y_{i,j,1} = \begin{cases} 1, & \text{if there exist the connection} \\ 0, & \text{if there exist no the connecttion} \end{cases}.$$

(e). The connection between nodes $v_{i,j-1}$ and $v_{i,j}$ ($1 \le i \le m$, and $2 \le j \le n$) is considered by

$$y_{i,j,2}, y_{i,j,2} = \begin{cases} 1, & \text{if there exists the connection} \\ 0, & \text{if there exists no the connection} \end{cases}.$$

(f). The origin (start state) and destination (final state) vertices are $O = v_{1,1}$ and $D = v_{m,n}$, respectively.

Note 3: Each path from O to D is in the form $P(O,D) = (O = v_{i_1,j_1}, v_{i_2,j_2}, ..., v_{i_k,j_k}, v_{i_{k+1},j_{k+1}}, ..., v_{i_l,j_l} = D)$, such that the connection between vertices v_{i_k,j_k} and $v_{i_{k+1},j_{k+1}}$ is according to Definition 5. Without prejudice to the generality of the definition, $P(O,D)$ can be considered such as $v_{i_1,j_1} v_{i_2,j_2} ... v_{i_k,j_k} v_{i_{k+1},j_{k+1}} ... v_{i_l,j_l}$.

5.2. General View of the Proposed Method

This subsection is given an overview design of a P system family with membrane division for solving the SPP according to BFS on the case study. The computation has two phases; phase one is to find the paths from O to D according to BFS, and phase two is finding the shortest path among the discovered paths. The gist of the research is as follows:

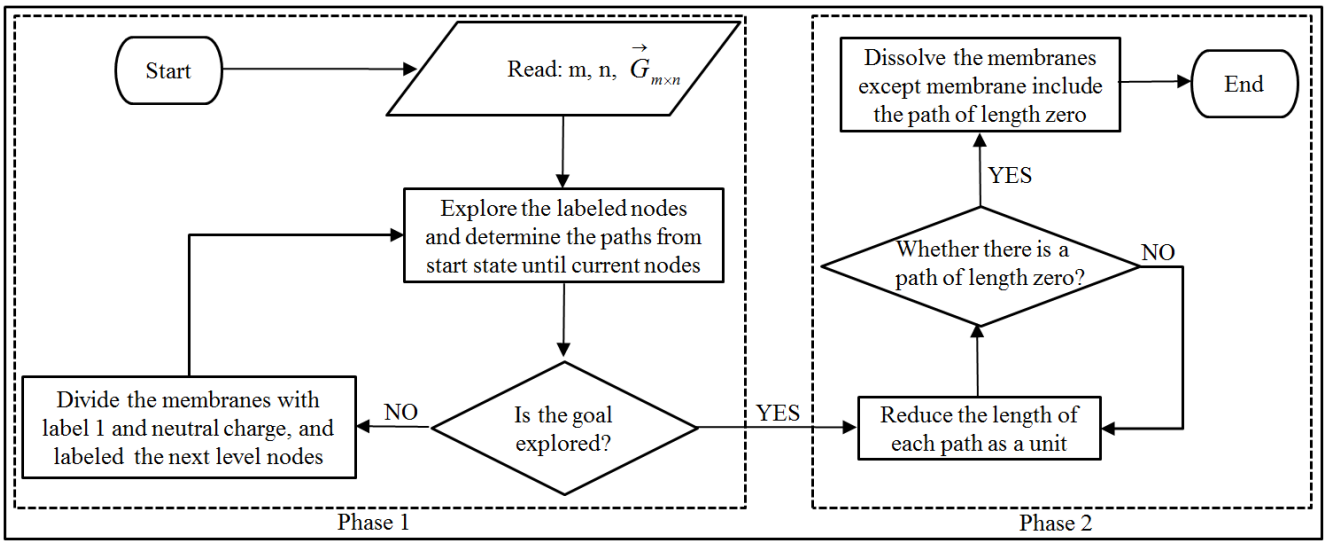

Figure 1. *Logical scheme.*

Phase 1: At each stage of the computation, the system (P system) explores the labeled nodes (in initial configuration the origin is explored), next, it checks to find the goal (destination). If the goal is explored then the computation will be entered to phase two, and if the goal is not explored then the membranes with label 1 (elementary membrane that located in the skin) will be divided and the next level nodes are labeled and the process will be continued until the goal is found. Considering the fact that each level of graph is explored in one stage, and if the goal exists at the depth *d*, the goal will be discovered after *d+1* stages. In section 5.3, it will be demonstrated that each stage takes 2 steps. After finding the goal, phase two will begin.

Phase 2: At this phase, in each step, the P system reduces one unit length of the each path, then checks the paths to find a path of length zero (for finding the shortest path). After obtaining shortest path, whole of the membrane will be dissolved except membranes including the shortest path, and

computation will be halted. In this case the remaining multisets in membrane labeled 1 show the shortest path from *O* to *D* (see the proposed method in Figure 1).

5.3. P System Algorithm for Solving the Shortest Path Problem

As all know algorithms that are compatible with common architectures are executed sequentially, thus, many problems in artificial intelligence are implementable using these architectures consecutively with high time complexity. P system is a new parallel framework which allows solving the hard problems in a feasible time.

Here is a solution to solve the SPP using P system with membranes division that the aim is to solve the SPP according to BFS on $\vec{G}_{m\times n}$. In this framework the time complexities for BFS and SPP are $O[2(m+n)]$ and $O[2L(m+n)]$, respectively, in which L is the maximum size length of the edges; whiles these complexities in sequential algorithm are $O\left(|V|+|E|\right)=O\left(3mn\right)$ and $O\left(|V|+L|E|\right)=O[mn(2L+1)]$, respectively. The details for obtaining this idea and P system structure are given bellow.

Proposed P system: P system $\Pi=(\Gamma,I_N,O_M,O_N,H,\mu,W,R,i_{out}^p)$ is considered with dynamic output and with the strong priorities $r_1>r_2$ and $r_7>r_8$ where:

(a) Numerical input $I_N=\{m,n\}\ \cup\{0,1,2,3,...,L\}$, where $m,n\in N$.

(b) Working alphabet

$$\Gamma=\{d,g,p,s,w\}\cup\left\{a_{i,j},v_{i,j}\middle|1\le i\le m,\ 1\le j\le n\right\}$$
$$\cup\left\{c_{i,j,1}\middle|2\le i\le m\ ,\ 1\le j\le n\right\}\cup\left\{c_{i,j,1}\middle|i=m+1,\ 1\le j\le n-1\right\}$$
$$\cup\left\{c_{i,j,2}\middle|1\le i\le m\ ,\ 2\le j\le n\right\}\cup\left\{c_{i,j,2}\middle|j=n+1,\ 1\le i\le m-1\right\}$$

(c) Output membrane $i_{out}^p=1^+$.

(d) Output alphabet $O_M=\left\{v_{i,j}\middle|1\le i\le m\ ,\ 1\le j\le n\right\}$.

(e) Numerical output $O_N=\left\{\ |u|_w\right\}$, where, u is the mightiest in the output region.

(f) The set of labels $H=\{0,1\}$.

(g) Initial membrane structure $\mu=[\ [\ ]_1^0\ ]_0^0$.

(h) The set of initial multisets $W=\{m_0,m_1\}$, where, $m_0=g$, $m_1=a_{1,1}v_{1,1}$.

The set of rules *R* are also considered as follows:

$$r_1\equiv[a_{m,n}]_1^0\rightarrow a_{m,n}[]_1^-$$

After applying this rule, the object $a_{m,n}$ will be sent outside, and the polarity of membrane 1 is changed from neutral to negative. This rule will be applied when there is the final state in membrane 1, and after that phase two will be started.

$$r_2\equiv[a_{i,j}]_1^0\rightarrow[c_{i+1,j,1}]_1^0[c_{i,j+1,2}]_1^0\ ;1\le i\le m\ ,\ 1\le j\le n$$

Object $a_{i,j}$ divides the membrane 1 and sends $c_{i+1,j,1}$ and $c_{i,j+1,2}$ to the core of each new membrane, that they labeled next level nodes, and they will explore the next level nodes in the next step (see r_5).

$$r_3\equiv[c_{m+1,j,k}]_1^0\rightarrow d\ ;1\le j\le n\ ,\ 1\le k\le 2$$

$$r_4\equiv[c_{i,n+1,k}]_1^0\rightarrow d\ ;1\le i\le m\ ,\ 1\le k\le 2$$

By applying rules of type r_3 and r_4, the membranes that they observe outside of the graph structure will be dissolved.

$$r_5\equiv[c_{i,j,k}\rightarrow w^{w_{i,j,k}}p^{w_{i,j,k}}v_{i,j}a_{i,j}^{y_{i,j,k}}s^{(1-y_{i,j,k})}]_1^0\ ;1\le i\le m\ ,\ 1\le j\le n\ ,\ 1\le k\le 2,w_{i,j,k}\in\{0,1,2,...,L\},y_{i,j,k}\in\{0,1\}$$

From $c_{i,j,k}$, rule r_5 makes $w^{w_{i,j,k}}$ (weight of the edge), $p^{w_{i,j,k}}$ (for finding the shortest path by rule r_7), $v_{i,j}$ (explored node), and one copy of $a_{i,j}$ or s (not both). If $y_{i,j,k}=1$, then there is one copy $a_{i,j}$ that labels the next level nodes in the next step by rules r_2 and the process will continue. If $y_{i,j,k}=0$, then there exist not connection between current and next level nodes, in this case one copy of object s will be appeared and the computation will be stopped in this path in the next step by rule r_6.

$$r_6\equiv[s]_1^0\rightarrow d$$

This rule is applied when there exists no connection between current and next level nodes (see r_5). Using this rule, the membrane with label 1 is dissolved, and computation will be stopped in this path.

$$r_7\equiv[p]_1^-\rightarrow p[]_1^-$$

By this rule, a copy of p is sent out from the membrane with the label 1 and negative charge in each step. Beginning of working this rule is start of phase two.

$$r_8\equiv g[]_1^-\rightarrow[g]_1^+$$

Object g is sent into membrane 1, and, simultaneously, the membrane polarity is changed from negative to positive. This membrane contains the shortest path from origin to destination.

Note 4: In this paper, for the notational convenience, multiset $v_{i_1,j_1}v_{i_2,j_2}...v_{i_t,j_t}v_{i_{t+1},j_{t+1}}...v_{i_l,j_l}$ is illustrated by A_{i_l,j_l},

where $1 \le i_t \le m$, $1 \le j_t \le n$, and $i_{t+1} = i_t, j_{t+1} = j_t + 1$, or $i_{t+1} = i_t + 1, j_{t+1} = j_t$. Also, the summation $w_{i_1,j_1,k_1} + w_{i_2,j_2,k_2} + ... + w_{i_t,j_t,k_t} + w_{i_{t+1},j_{t+1},k_{t+1}} + ... + w_{i_l,j_l,k_l}$ and the sequence $y_{i_1,j_1,k_1}, y_{i_2,j_2,k_2}, ..., y_{i_t,j_t,k_t}, y_{i_{t+1},j_{t+1},k_{t+1}}, ..., y_{i_l,j_l,k_l}$ are shown by δ_{i_l,j_l} and λ_{i_l,j_l} , respectively, where $1 \le i_t \le m$, $1 \le j_t \le n, \delta_{1,1} = 0$, and $i_{t+1} = i_t, j_{t+1} = j_t + 1, k_t = 2$, or $i_{t+1} = i_t + 1, j_{t+1} = j_t, k_t = 1$. Note that in all of above relations $m, n \in N$.

Lemma 1: Let Π be the proposed P system with mentioned components such that $y_{i,j,k} = 1$, for all $y_{i,j,k}$, and let $A_{i,j}$ and $\delta_{i,j}$ be parameters according to Note 4. If C is an arbitrary computation of the system and $2 \le K \le m + n$, then the multiset associated with membrane 1^0 (membrane with label 1 and neutral charge) is in the form $w^{\delta_{i,j}} p^{\delta_{i,j}} A_{i,j} a_{i,j}$ in the configuration $C^{2(K-2)}$, and in this case $i + j = K$.

Proof: See Appendix A.

Lemma 2: Let C be an arbitrary computation of the proposed P system. Consider the parameters $A_{i,j}$, $\delta_{i,j}$, and $\lambda_{i,j}$ according to Note 4, and let the terms of sequence $\lambda_{i,j}$ and the objects of multiset $A_{i,j}$ have matched indexes. For each K ($2 \le K \le m + n$) the following holds:

(a). If each term of the sequence $\lambda_{i,j}$ is equal 1, then for configuration $C^{2(K-2)}$, there exists a unique membrane 1^0 whose associated multiset is in the form $w^{\delta_{i,j}} p^{\delta_{i,j}} A_{i,j} a_{i,j}$, such that $i + j = K$.

(b). If $y_{i,j,k} = 0$ ($i + j = K$) is the first term of sequence $\lambda_{i,j}$ that it is equal 0, then the membrane 1^0 containing multiset $A_{i,j}$ will be dissolved in configuration $C^{2(K-2)+1}$.

Proof: See Appendix B.

Lemma 3: Let R' be a set of rules such as $R' = \{r_1, r_2, r_3, r_4, r_5, r_6\}$, and let $A_{i,j}$, $\delta_{i,j}$ and $\lambda_{i,j}$ be according Note 4. If Π is the proposed P system with mentioned ingredients, then:

$$T(C^0(\Pi|_{R'})) \le 2(m + n - 2) + 1 .$$

Also, if exists a sequence $\lambda_{i,j}$ with terms of equal 1, then:

$$T(C^0(\Pi|_{R'})) = 2(m + n - 2) + 1$$

Proof: See Appendix C.

Theorem 1: If $\vec{G}_{m \times n}$ is the case study, then the BFS algorithm runs in $O[2(m+n)]$ on $\vec{G}_{m \times n}$ by a P system with division rules.

Proof: See Appendix D.

Lemma 4: Let $A_{i,j}$, $\delta_{i,j}$ and $\lambda_{i,j}$ be according to Note 4, and let exists a sequence $\lambda_{i,j}$ that its terms are equal 1. If C is an arbitrary computation of the proposed P system with the mentioned ingredients, then the following holds:

(c). A membrane 1^+ containing multiset $gw^{\delta_{m,n}} A_{m,n}$ will be obtained, in configuration $C^{2(m+n-2)+\underset{1^-}{Min}\{\delta_{m,n}\}+2}$, where $\delta_{m,n} = \underset{1^-}{Min}\{\delta_{m,n}\}$.

(d). $T(C^{2(m+n-2)+1}(\Pi)) = \begin{cases} \underset{1^-}{Max}\{\delta_{m,n}\} & , \quad \underset{1^-}{Max}\{\delta_{m,n}\} > \underset{1^-}{Min}\{\delta_{m,n}\} \\ \underset{1^-}{Min}\{\delta_{m,n}\} + 1 & , \quad \underset{1^-}{Max}\{\delta_{m,n}\} = \underset{1^-}{Min}\{\delta_{m,n}\} \end{cases}$.

Proof: See Appendix E.

Lemma 5: Let $\delta_{i,j}$ and $\lambda_{i,j}$ be according to Note 4, and let exists a sequence $\lambda_{i,j}$ that its terms are equal to 1. If Π is the proposed P system with the mentioned ingredients, then:

$$T(C^0(\Pi)) = \begin{cases} 2(m+n-2) + \underset{1^-}{Max}\{\delta_{m,n}\} + 1, & \underset{1^-}{Max}\{\delta_{m,n}\} > \underset{1^-}{Min}\{\delta_{m,n}\} \\ 2(m+n-2) + \underset{1^-}{Min}\{\delta_{m,n}\} + 2, & \underset{1^-}{Max}\{\delta_{m,n}\} = \underset{1^-}{Min}\{\delta_{m,n}\} \end{cases}$$

Proof: See Appendix F.

Theorem 2: If $\vec{G}_{m \times n}$ is the case study, then the SPP algorithm runs in $O[2(m+n)]$ on $\vec{G}_{m \times n}$ by a P system with division rules, in which L is the maximum magnitude length of the edges.

Proof: See Appendix G.

6. Simulations

In this section, the ability of this technique is shown by following the computation of the examples that were shown in Figures 2 and 5.

Example 1: The goal is to find the shortest path from $v_{1,1}$ to $v_{3,4}$ in the weighted graph $\vec{G}_{3 \times 4}$, which it is shown in Figure 2 by proposed P system. In this case, $m = 3$, $n = 4$, $\underset{1^-}{Min}\{\delta_{m,n}\} = 8$, and $\underset{1^-}{Max}\{\delta_{m,n}\} = 11$. By Lemma 3, whole of paths will be obtained in configuration $C^{2(m+n-2)+1} = C^{11}$, and from Lemma 5, the shortest path will be achieved in configuration $C^{2(m+n-2)+\underset{1^-}{Max}\{\delta_{m,n}\}+1} = C^{22}$. The paths from $v_{1,1}$ (start) to $v_{3,4}$ (goal) are available in the membranes 1^- and 1^+ , which the shortest path is accessible in membrane 1^+ . Numerical output (power of w) shows the weight of path and output objects show the path. For verification, the program has run in the P-Lingua [20]. The final configuration (Configuration 22) is shown by P-Lingua simulator in Figure 3.

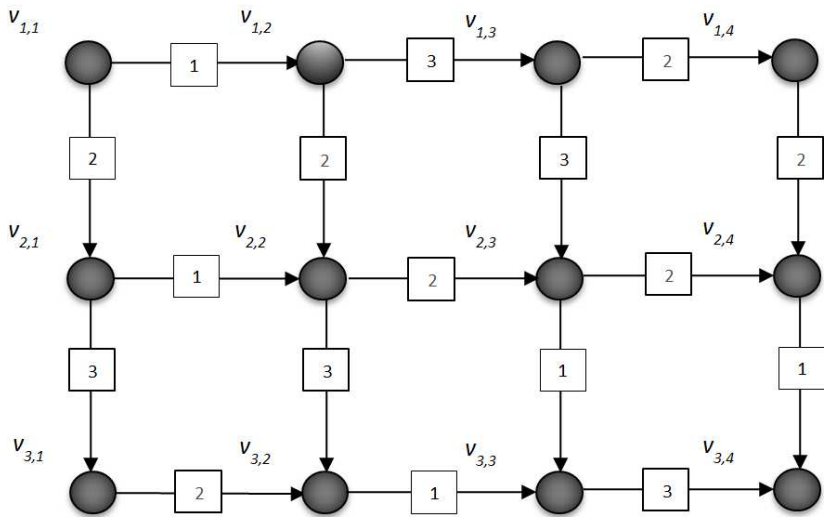

Figure 2. *Weighted graph* $\vec{G}_{3\times4}$ *, with positive integer weights mentioned in Example 1.*

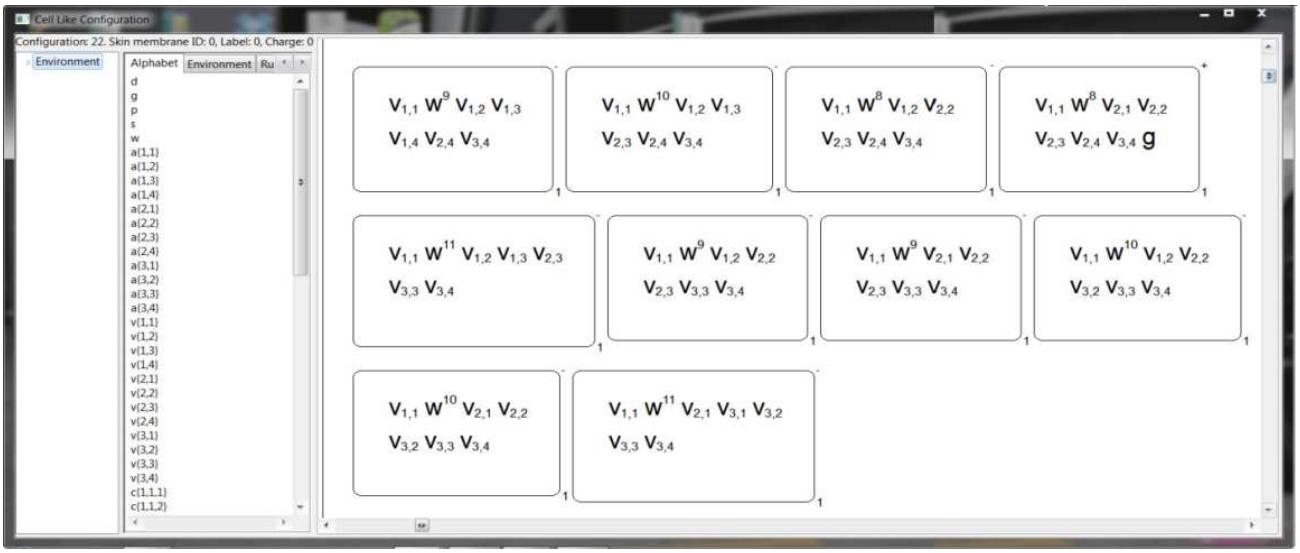

Figure 3. *The final configuration of the proposed method for finding the shortest path in weighted graph* $\vec{G}_{3\times4}$ *mentioned in Figure 2 by P-Lingua simulator.*

Example 2: The goal is to find two paths with a minimum length among the whole of paths in Example 1. It is enough to consider $m_0 = g^2$ as the initial multiset in region 0. In this case, there will be two membranes 1^+ in the final configuration that they show two paths. The simulation is carried out using P-Lingua shown in Figure 4.

Figure 4. *The final configuration of the proposed method for finding two shortest paths in weighted graph* $\vec{G}_{3\times4}$ *mentioned in Figure 2 by P-Lingua simulator.*

Example 3: The goal is to find the shortest path from $v_{1,1}$ to $v_{3,4}$ in non-complete weighted graph $\vec{G}_{3\times4}$, which it is shown in Figure 5 by the proposed P system. The whole process is according to Example 1, but with fewer paths. The final configuration of the example simulation using P-Lingua is visible in Figure 6.

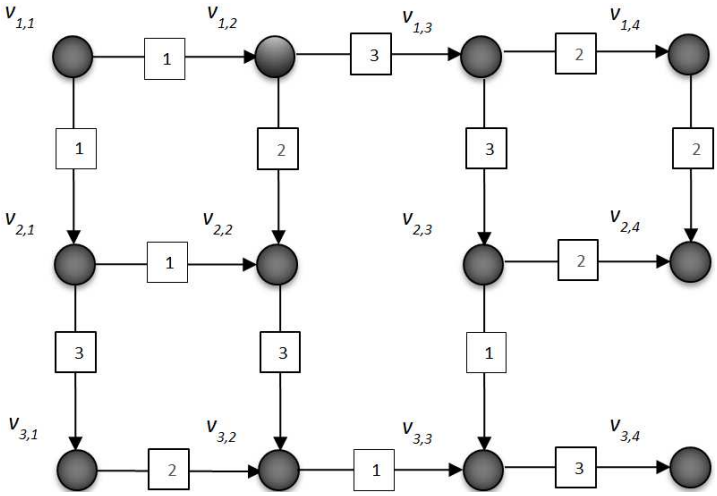

Figure 5. *Non-complete weighted graph $\vec{G}_{3\times4}$, with positive integer weights mentioned in Example 3.*

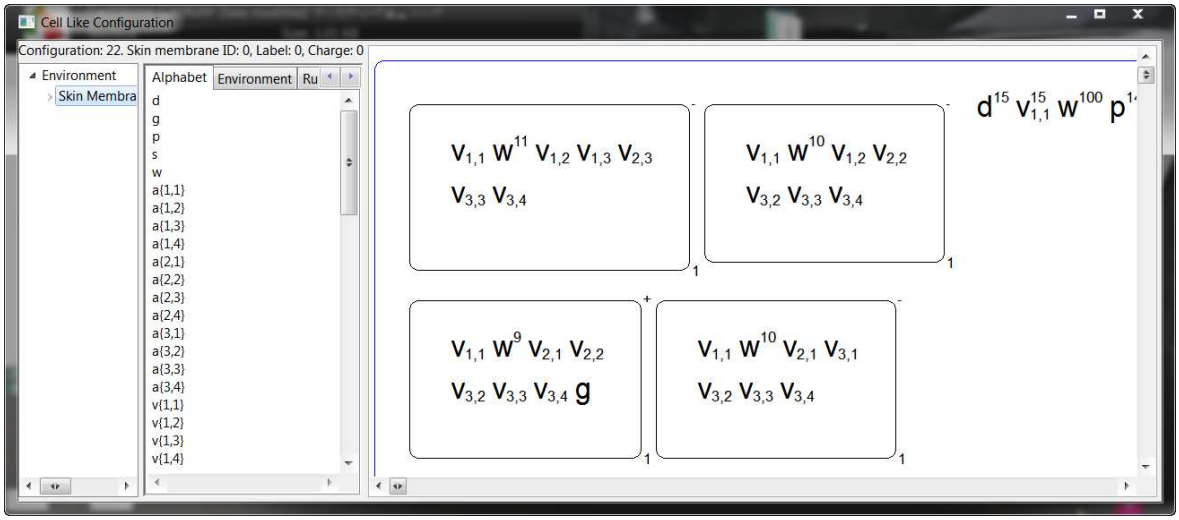

Figure 6. *The final configuration of the proposed method for finding the shortest path in non-complete weighted graph $\vec{G}_{3\times4}$ mentioned in Figure 5 by P-Lingua simulator.*

7. Conclusion and Future Works

In this study, a novel parallel approach based on P system has been presented to find the shortest path between two points according to the breadth-first search (BFS) in a grid with m rows and n columns of the nodes. The proposed P system works based on membranes division, and it increases the workspace during the computation and decreases the time complexity. This method reduces the time execution of BFS from $O(3mn)$ to $O[2(m+n)]$ and the running time of SPP from $O[mn(2L+1)]$ to $O[2L(m+n)]$ on the mentioned grid, where L is as the maximum length of the edges. As the future works, improvement of the proposed P system is suggested, considering the design and its ingredients. Also, investigation of the problem on a new case study such as arbitrary graphs, and implementation of the study on the graphics processing unit (GPU) is recommended.

Appendix A - Proof of Lemma 1

The lemma is proved by induction on K.

Case *1* ($K=2$): In initial configuration (C^0), there exist only one membrane 1^0, such that its associated multiset is $v_{1,1}a_{1,1} = w^{\delta_{1,1}} p^{\delta_{1,1}} A_{1,1}a_{1,1}$, thus the proposition is true for $K=2$.

Case 2 ($K<m+n \rightarrow K+1$): Let $i<m$ and $j<n$, by inductive hypothesis, each membrane 1^0 contains a multiset

in the form $w^{\delta_{i,j}} p^{\delta_{i,j}} A_{i,j} a_{i,j}$ in configuration $C^{2(K-2)}$, and in this case $i+j=K$. In the next step rules r_2 can be applied and membranes 1^0 will be duplicated; half of them including $w^{\delta_{i,j}} p^{\delta_{i,j}} A_{i,j} c_{i+1,j,1}$, and the remaining including $w^{\delta_{i,j}} p^{\delta_{i,j}} A_{i,j} c_{i,j+1,2}$, in this time, configuration $C^{2(K-2)+1}$ will be obtained. In the step $2(K-2)+2=2((K+1)-2)$, objects $c_{i+1,j,1}$ and $c_{i,j+1,2}$ are changed to $w^{w_{i+1,j,1}} p^{w_{i+1,j,1}} v_{i+1,j} a_{i+1,j}$, and $w^{w_{i,j+1,2}} p^{w_{i,j+1,2}} v_{i,j+1} a_{i,j+1}$, respectively, by rules r_5. Therefore, in configuration $C^{2((K+1)-2)}$, each membrane 1^0 contains a multiset such as follows:

$$w^{\delta_{i,j}+w_{i+1,j,1}} p^{\delta_{i,j}+w_{i+1,j,1}} A_{i,j} v_{i+1,j} a_{i+1,j} = w^{\delta_{i+1,j}} p^{\delta_{i+1,j}} A_{i+1,j} a_{i+1,j} = w^{\delta_{I,J}} p^{\delta_{I,J}} A_{I,J} a_{I,J},\ I=i+1,\ J=j \quad (1)$$

$$w^{\delta_{i,j}+w_{i,j+1,2}} p^{\delta_{i,j}+w_{i,j+1,2}} A_{i,j} v_{i,j+1} a_{i,j+1} = w^{\delta_{i,j+1}} p^{\delta_{i,j+1}} A_{i,j+1} a_{i,j+1} = w^{\delta_{I,J}} p^{\delta_{I,J}} A_{I,J} a_{I,J},\quad I=i, J=j+1 \quad (2)$$

in this case $I+J=K+1$.

Now, let $i=m, j<n$ or $i<m$, $j=n$ therefore, some membranes 1^0 include $a_{m,j}$ or $a_{i,n}$, in configuration $C^{2(K-2)}$. For these membranes, configuration $C^{2(K-2)+1}$ are obtained by applying rules r_2, and these membranes will be divided; each of them containing an object $c_{m+1,j,1}$ or $c_{i,n+1,2}$, that in the next configuration ($C^{2((K+1)-2)}$) will be dissolved by rules r_3 and r_4, and thus the lemma holds for *K*.

Appendix B - Proof of Lemma 2

(a) If $y_{i_t,j_t,kt}=1$, for each $y_{i_t,j_t,kt} \in \lambda_{i,j} (1 \le i_t \le i, 1 \le j_t \le j, 1 \le k_t \le 2)$, then by considering the proof of Lemma 1 it is deducted that there exists a unique membrane 1^0 whose associated multiset is in the form $w^{\delta_{i,j}} p^{\delta_{i,j}} A_{i,j} a_{i,j}$, such that $i+j=K$.

(b) By hypothesis, if $y_{i_t,j_t,kt} \in \lambda_{i,j}$, then $y_{i_t,j_t,kt}=1$ and $y_{i,j,k}=0$, where $1 \le i_t < i_l = i$, $1 \le j_t < j_l = j$, $1 \le k_t \le 2$. Tues, by part (a), there exists a unique membrane 1^0 whose associated multiset is in the form $w^{\delta_{i_l-1,j_l-1}} p^{\delta_{i_l-1,j_l-1}} A_{i_{l-1},j_{l-1}} a_{i_{-1},j_{-1}}$ in configuration $C^{2((K-1)-2)}$, such that $i_l + j_l = K-1$. In this configuration rules r_2 and after that rules r_5 can be applied. In this case, membrane 1^0 with multiset associated $w^{\delta_{i,j}} p^{\delta_{i,j}} A_{i,j} s$ will be obtained, in configuration $C^{2(K-2)}$. Now, for this membrane rule r_6 can be applied and membrane will be dissolved, in configuration $C^{2(K-2)+1}$. Therefore, membrane 1^0 containing multiset $A_{i,j}$ will be dissolved in configuration $C^{2(K-2)+1}$.

Appendix C - Proof of Lemma 3

From Lemmas 1 and 2, and their proofs, if exist some membranes 1^0 in configuration $C^{2(m+n-2)}(\Pi)$, then each of these membranes contain a multiset in the form $w^{\delta_{m,n}} p^{\delta_{m,n}} A_{m,n} a_{m,n}$ or $w^{\delta_{m,n}} p^{\delta_{m,n}} A_{m,n} s$. The next configuration is obtained by applying rules r_1 and r_6 simultaneously. In this case the membranes containing multiset $w^{\delta_{m,n}} p^{\delta_{m,n}} A_{m,n} s$ will be dissolved by applying rule r_6; and objet $a_{m,n}$ will be sent out of membranes 1^0 containing multiset $w^{\delta_{m,n}} p^{\delta_{m,n}} A_{m,n} a_{m,n}$ by applying rule r_1, and the charge of membrane 1^0 is changed to negative, simultaneously, and configuration $C^{2(m+n-2)+1}(\Pi)$ is obtained. Thus, considering $R' = \{r_1, r_2, r_3, r_4, r_5, r_6\}$ computation will be halted for P system $\Pi\big|_{R'}$, in configuration $C^{2(m+n-2)+1}(\Pi\big|_{R'})$.

Appendix D - Proof of Theorem 1

Let $\vec{G}_{m\times n}$ and its ingredients be according to Definition 5 and Remark 2, and let $\Pi\big|_{R'}$ be mentioned P system in Lemma 3 with output membrane 1^-. If $P(v_{1,1}, v_{m,n}) = (v_{1,1} = v_{i_1,j_1}, v_{i_2,j_2}, ..., v_{i_k,j_k}, v_{i_{k+1},j_{k+1}}, ..., v_{i_l,j_l} = v_{m,n})$ is a path from origin to destination, then according to Remarks 2(iv) and 2(v), the terms of sequence $y_{i_1,j_1,k_1}, y_{i_2,j_2,k_2}, ..., y_{i_t,j_t,k_t}, y_{i_{t+1},j_{t+1},k_{t+1}}, ..., y_{i_l,j_l,k_l}$ are equal to 1. From Lemma 3 and its proof using P system $\Pi\big|_{R'}$ with numerical input according to the ingredients of $\vec{G}_{m\times n}$, exists a membrane 1^- containing multiset $w^{\delta_{m,n}} p^{\delta_{m,n}} A_{m,n}$ in configuration $C^{2(m+n-2)+1}$. Considering $A_{m,n} \equiv P(v_{1,1}, v_{m,n})$, it is deduced that, P system $\Pi\big|_{R'}$ explores path $P(v_{1,1}, v_{m,n})$ in $O[2(m+n)]$. Also by Lemma 2, each path from $v_{1,1}$ until $v_{i,j}$ will be explored in $2(K-2)$ steps, where $K=i+j$, it means the searching is based on BFS.

Appendix E - Proof of Lemma 4

(a) From proofs of Lemmas 1 and 2, the membrane structure of P system Π includes a skin and some membranes 1^0, whose associated multisets are in the form $w^{\delta_{m,n}} p^{\delta_{m,n}} A_{m,n} a_{m,n}$, in configuration $C^{2(m+n-2)}$, and by proof of Lemma 3, the membranes will be changed to 1^- including multiset in the form $w^{\delta_{m,n}} p^{\delta_{m,n}} A_{m,n}$, in configuration $C^{2(m+n-2)+1}$. In the next step, rules r_7 and r_8 can be applied but

with priority $r_7 > r_8$; thus, rule r_7 will be applied for $\underset{1^-}{Min}\{\delta_{m,n}\}$ times, and one copy of object p will be sent out from each membrane 1^- in each step. In this time, configuration $C^{2(m+n-2)+\underset{1^-}{Min}\{\delta_{m,n}\}+1}$ is obtained, and some membranes 1^- are empty from object p, these membranes are considered with label $(1^-)'$. Next, rules r_7 and r_8 will be applied for membranes 1^- and $(1^-)'$, simultaneously. By applying rule r_7, an object p will be sent out of membrane 1^-, and by applying rule r_8, object g will be sent in of one of the membranes $(1^-)'$, and changes the polarity of membrane $(1^-)'$ from negative to positive, and configuration $C^{2(m+n-2)+\underset{1^-}{Min}\{\delta_{m,n}\}+2}$ will be obtained with a membrane 1^+, containing multiset $gw^{\delta_{m,n}}A_{m,n}$, where $\delta_{m,n} = \underset{1^-}{Min}\{\delta_{m,n}\}$.

(b) From part (a), if $\underset{1^-}{Max}\{\delta_{m,n}\} > \underset{1^-}{Min}\{\delta_{m,n}\}$, then rule r_7 can be applied on some of the membranes 1^-, from configuration $C^{2(m+n-2)+1}$ until configuration $C^{2(m+n-2)+\underset{1^-}{Max}\{\delta_{m,n}\}+1}$, and computation will be halted. Also, if $\underset{1^-}{Max}\{\delta_{m,n}\} = \underset{1^-}{Min}\{\delta_{m,n}\}$, then according to part (a), computation is halted in configuration $C^{2(m+n-2)+\underset{1^-}{Min}\{\delta_{m,n}\}+2}$.

Appendix F - Proof of Lemma 5

By Lemma 4(a), $T(C^0(\Pi)) > 2(m+n-2)+1$; thus, considering Note 2:

$$T(C^0(\Pi)) = [2(m+n-2)+1] + T(C^{2(m+n-2)+1}(\Pi))$$

then, the proof follows from Lemma 4(b).

Appendix G - Proof of Theorem 2

Let $\vec{G}_{m\times n}$ and its ingredients be according to Definition 5 and Remark 2. On one hand, from the proof of Theorem 1, whole of paths from origin to destination are explored in $\vec{G}_{m\times n}$ based on BFS using proposed P system Π in configuration $C^{2(m+n-2)+1}$. On the other hand, each $\delta_{m,n}$ is the length of a path from origin to destination, and after finding the shortest path as mentioned in Lemma 4(a), the computation will be halted. Also, by Lemma 5,

$$T(C^0(\Pi)) = \begin{cases} 2(m+n-2) + \underset{1^-}{Max}\{\delta_{m,n}\} + 1, & \underset{1^-}{Max}\{\delta_{m,n}\} > \underset{1^-}{Min}\{\delta_{m,n}\} \\ 2(m+n-2) + \underset{1^-}{Min}\{\delta_{m,n}\} + 2, & \underset{1^-}{Max}\{\delta_{m,n}\} = \underset{1^-}{Min}\{\delta_{m,n}\} \end{cases}$$

and in the worst case $\delta_{m,n} = (m+n-2)L$, where L is the maximum magnitude length of edges; then,

$$T(C^0(\Pi)) = 2(m+n-2)L + 2 ,$$

which concludes the proof.

References

[1] Păun, G., *Computing with Membranes.* Journal of Computer and System Sciences, 2000. 61(1): p. 108-143.

[2] Păun, G., *Membrane computing: an introduction*. 2012: Springer Science & Business Media.

[3] Păun, G., *P systems with active membranes: attacking NP complete problems.* Journal of Automata, Languages and Combinatorics, 2001. 6(1): p. 75-90.

[4] Krishna, S. N. and R. Rama, *A variant of P systems with active membranes: Solving NP-complete problems.* Romanian Journal of Information Science and Technology, 1999. 2(4): p. 357-367.

[5] Mutyam, M. and K. Krithivasan, *P Systems with Membrane Creation: Universality and Efficiency*, in *Machines, Computations, and Universality*, M. Margenstern and Y. Rogozhin, Editors. 2001, Springer Berlin Heidelberg. p. 276-287.

[6] Frisco, P., M. Gheorghe, and M. J. Pérez-Jiménez, *Applications of Membrane Computing in Systems and Synthetic Biology*. 2014: Springer.

[7] Peng, H., et al., *An automatic clustering algorithm inspired by membrane computing.* Pattern Recognition Letters, 2015. 68: p. 34-40.

[8] Chen, H., et al., *On trace languages generated by (small) spiking neural P systems.* Theoretical Computer Science, 2016.

[9] Wang, J., P. Shi, and H. Peng, *Membrane computing model for IIR filter design.* Information Sciences, 2016. 329: p. 164-176.

[10] Song, B., T. Song, and L. Pan, *A time-free uniform solution to subset sum problem by tissue P systems with cell division.* Mathematical Structures in Computer Science, 2017. 27(1): p. 17-32.

[11] Gutiérrez-Naranjo, M. and M. Pérez-Jiménez, *Depth-First Search with P Systems*, in *Membrane Computing*, M. Gheorghe, et al., Editors. 2011, Springer Berlin Heidelberg. p. 257-264.

[12] Gutiérrez-Naranjo, M. A. and M. J. Pérez-Jiménez, *Implementing local search with Membrane Computing.* 2011.

[13] Gutiérrez-Naranjo, et al. *Solving the N-Queens puzzle with P systems*. in *7th Brainstorming Week on Membrane Computing*. 2009. Sevilla, España: Fénix Editora.

[14] Michael J. Dinneen, Y. -B. K., and Radu Nicolescu, *Edge- and node-disjoint paths in P systems*, in *Electronic Proceedings in Theoretical Computer Science* 2010. p. 121-141.

[15] Nicolescu, R. and H. Wu, *BFS Solution for Disjoint Paths in P Systems*, in *Unconventional Computation*, C. Calude, et al., Editors. 2011, Springer Berlin Heidelberg. p. 164-176.

[16] Nicolescu, R. and H. Wu, *New solutions for disjoint paths in P systems.* Natural Computing, 2012. 11(4): p. 637-651.

[17] Salehi, E., S. M. Shamsuddin, and K. Nemati, *A Linear Time Complexity of Breadth-First Search Using P System with Membrane Division.* Mathematical Problems in Engineering, 2013. 2013: p. 11.

[18] Russell, S. J. and P. Norvig, *Artificial Intelligence: A Modern Approach(2nd Edition)* 2ed. 2002: Prentice Hall.

[19] Even, S. and E. Guy, *Graph algorithms*. 2nd ed. 2012, Cambridge, NY: Cambridge University Press. xii, 189 p.

[20] Díaz-Pernil, D., et al., *A P-Lingua Programming Environment for Membrane Computing*, in *Membrane Computing*, D. Corne, et al., Editors. 2009, Springer Berlin Heidelberg. p. 187-203.