

A Tissue P Systems Based Uniform Solution To Tripartite Matching Problem

Yunyun Niu, Linqiang Pan*

*Key Laboratory of Image Processing and Intelligent Control
Department of Control Science and Engineering
Huazhong University of Science and Technology
Wuhan 430074, China
niuyunyun1003@163.com, lqpan@mail.hust.edu.cn*

Mario J. Pérez-Jiménez

*Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Sevilla, 41012 Sevilla, Spain
marper@us.es*

Miquel Rius Font

*Department of Applied Mathematics IV
Universitat Politècnica de Catalunya
Edifici C3, Despatx 016, Av. del Canal Olímpic s/n, 08860 Castelfels, Spain
mrius@ma4.upc.edu*

Abstract. A tissue P system with cell division is a computing model which has two basic features: intercellular communication and the ability of cell division. The ability of cell division allows us to obtain an exponential amount of cells in linear time and to design cellular solutions to computation-ally hard problems in polynomial time. In this work we present an efficient solution to the tripartite matching problem by a family of such devices. This solution leads to an interesting open problem

*Address for correspondence: Huazhong University of Science and Technology, Wuhan 430074, China

whether tissue P systems with cell division and communication rules of length 2 can solve NP-complete problems. An answer to this open problem will provide a borderline between efficiency and non-efficiency in terms of the lengths of communication rules.

Keywords: Membrane Computing, Tissue P System, Cell Division, Tripartite Matching Problem.

1. Introduction

Membrane computing is an emergent branch of natural computing, which is inspired by the structure and the functioning of living cells [10], as well as the organization of cells in tissues, organs, and other higher order structures. It provides distributed parallel and non-deterministic computing models called *P systems*. Since being introduced by Gh. Păun in 1998, membrane computing has received important attention from the scientific community. As computer scientists, biologists, formal linguists and complexity theoreticians plug into this area, it has definitely become a rich and exciting realm of cross-disciplinary research. Please refer to [11, 13] for an introduction of membrane computing, and refer to [14] for further bibliography.

In last years, many different classes of P systems have been investigated. The most studied variants are the *cell-like P systems*, where membranes are hierarchically arranged in a tree-like structure. Most of them are *computationally universal*, i.e., able to compute whatever a Turing machine can do, as well as *computationally efficient*, i.e., able to trade space for time and solve in this way presumably intractable problems in a feasible time (e.g., [1, 2, 8]).

Another interesting class of P systems are *tissue P systems*, where instead of considering a hierarchical arrangement, membranes are placed in the nodes of a graph. Tissue P systems are an abstraction of communicating and cooperating cells in tissues [7], where the membrane structure did not change along the computation. However, alive tissues are not static network of cells, since new cells are generated by membrane fission in a natural way. With this biological inspiration, *tissue P systems with cell division* were introduced [12].

One of the main features of tissue P systems with cell division is related to their computational efficiency. Specifically, tissue P systems with cell division can solve **NP**-complete problems in polynomial time (even linear time), e.g., the subset sum problem [4], the partition problem [5], and the 3-coloring problem [3]. The 3-coloring problem is **NP**-complete, and *tripartite matching problem* (TMP) also belongs to the class of **NP**-complete problems, so tripartite matching problem can be reduced to 3-coloring problem in polynomial time. But it still remains open how one can compute the reduction of an **NP** problem to another **NP**-complete problem by P systems. In this work, we give a direct solution to tripartite matching problem in polynomial time by tissue P systems with cell division.

The TMP can be described as following [9]: *Given three sets B, G and H , each containing n elements, and a set T of triples $\subseteq B \times G \times H$. Decide whether there exists a subset T' of T such that $|T'| = n$ and no two of triples belong to T' have a component in common.* In this work, for each size n , a P system with cell division is constructed for solving all instances of TMP with size n , where the computation time is polynomial with respect to n .

For an instance of TMP with sets B, G, H and $T \subseteq B \times G \times H$, $|B| = |G| = |H|$, we can associate a hypergraph \mathcal{G} with the vertex set $V(\mathcal{G}) = B \cup G \cup H$, the hyperedge set $E(\mathcal{G}) = T$ (that is, a triple in T is a hyperedge in \mathcal{G}). In this way, TMP can be restated to decide whether there exists a sub-hypergraph

\mathcal{G}' such that $V(\mathcal{G}) = V(\mathcal{G}')$ and no two hyperedges of $E(\mathcal{G}')$ have a common vertex. In the hypergraph version of TMP, it is easy to see that the family of tissue P systems with cell division is independent of the number of hyperedges. However, the family of P systems with cell division constructed in [3] for the 3-coloring problem depends on the number of vertices and the number of edges. For instances of the 3-coloring problem with the same number of vertices, but with different number of edges, a family of different P systems have to be constructed. The technique of constructing P systems given in this work is an improvement of the construction in [3] in the sense that the family of P systems only depends on the number of vertices.

The solution to TMP given in this work shows that the lengths of communication rules have essential role for the efficiency of tissue P systems with cell division. Specifically, it leads to an interesting open problem whether tissue P systems with cell division and communication rules of length 2 can solve NP-complete problems. An answer to this open problem will provide a borderline between efficiency and non-efficiency in terms of the lengths of communication rules.

The paper is organized as follows: some preliminaries are recalled in section 2, including the definition of recognizer tissue P systems with cell division; a polynomial-time solution to the tripartite matching problem is presented in section 3, with a short overview of the computation, and the informal verification of the solution; some discussion is presented in section 4.

2. Preliminaries

An *alphabet* Σ is a non-empty set, whose elements are called *symbols*. A *string* is an ordered sequence of symbols. The number of symbols in a string u is the *length* of the string, and it is denoted by $|u|$. For a set T , the notation $|T|$ denotes the cardinality of T . As usual, the empty string (with length 0) will be denoted by λ . The set of strings of length n built with symbols from the alphabet Σ is denoted by Σ^n and $\Sigma^* = \cup_{n \geq 0} \Sigma^n$. A *language* over Σ is a subset from Σ^* .

A *multiset* m over a set A is a pair (A, f) where f is a map from A to the set of natural numbers \mathbb{N} . If $m = (A, f)$ is a multiset then its *support* is defined as $supp(m) = \{x \in A \mid f(x) > 0\}$ and its *size* is defined as $\sum_{x \in A} f(x)$. A multiset is empty (resp. finite) if its support is the empty set (resp. finite).

If $m = (A, f)$ is a finite multiset over A , and $supp(m) = \{a_1, \dots, a_k\}$, then it will be denoted as $m = \{\{a_1^{f(a_1)}, \dots, a_k^{f(a_k)}\}\}$. That is, superscripts indicate the multiplicity of each element, and if $f(x) = 0$ for any $x \in A$, then this element is omitted. If $m_1 = (A, f)$ and $m_2 = (A, g)$ are multisets over A , then the union of m_1 and m_2 is defined as $m_1 m_2 = (A, h)$, where $h = f + g$.

A recognizer tissue P system with cell division of degree $q \geq 1$ is a tuple of the form

$$\Pi = (\Gamma, \Sigma, \Omega, w_1, \dots, w_q, \mathcal{R}, i_{in}, i_{out}), \text{ where:}$$

- $q \geq 1$ (the initial degree of the system; the system contains q cells, labeled with $1, 2, \dots, q$; all these q cells are placed in the environment; the environment is labeled with 0);
- Γ is the working alphabet, which contains two distinguished objects *yes* and *no*, at least one copy of them occurring in some initial multisets w_1, \dots, w_q , but not occurring in Ω ;
- Σ is an input alphabet strictly contained in Γ ;
- $\Omega \subseteq \Gamma$ is the set of objects occurring in the environment, each one in arbitrarily many copies;

- w_1, \dots, w_q are strings over Γ , describing the multisets of objects located in the cells of the system at the beginning of the computation;
- \mathcal{R} is a finite set of rules of the following forms:
 - (a) *Communication rules*: $(i, u/v, j)$, for $i, j \in \{0, 1, 2, \dots, q\}, i \neq j, u, v \in \Gamma^*$ ($|u| + |v|$ is called the length of the communication rule $(i, u/v, j)$).
 - (b) *Division rules*: $[a]_i \rightarrow [b]_i[c]_i$, where $i \in \{1, 2, \dots, q\}, a \in \Gamma$ and $b, c \in \Gamma \cup \{\lambda\}$.
- $i_{in} \in \{1, \dots, q\}$ is the input cell;
- $i_{out} \in \{0, 1, \dots, q\}$ indicates the output region, where $i_{out} = 0$ denotes that the output region is the environment;
- All computations halt (that is, they always reach a configuration where no further rule can be applied);
- During a computation of Π , either the object yes or the object no (but not both) must be released into the environment, and only in the last step of the computation.

When a division rule $[a]_i \rightarrow [b]_i[c]_i$ is applied, all the objects in the original cells are replicated and copies of them are placed in each of the new cells, with the exception of the object a , which is replaced by $b \in \Gamma \cup \{\lambda\}$ in the first new cell and by $c \in \Gamma \cup \{\lambda\}$ in the second one.

When a rule $(i, u/v, j)$ is applied, the objects of the multiset represented by u are sent from region i to region j and simultaneously the objects of the multiset v are sent from region j to region i . For a cell in the system Π , it is possible to have more than one applicable communication rules in a step. These applicable communication rules are used in non-deterministic maximally parallel manner (the system non-deterministically chooses and applies a multiset of communication rules that is maximal, no further rule can be added).

In general, the rules of a system as above are used in the non-deterministic maximally parallel manner. In each step, all cells which can evolve must evolve in a maximally parallel way. This way of applying rules has only one restriction: when a cell is divided, the division rule is the only one which is applied for that cell in that step; the objects inside that cell do not evolve by means of communication rules. Their labels precisely identify the rules which can be applied to them.

A configuration of Π at an instant t is described by the multisets of objects over Γ associated with all the cells present in the system at that moment, and the multiset over $\Gamma - \Omega$ associated with the environment at the instant t . All computations start from the initial configuration and proceed as defined above. A computation \mathcal{C} is called an accepting computation (respectively, rejecting computation) if the object yes (respectively, no) appears in the environment associated to the corresponding halting configuration of \mathcal{C} , and only in the last step of the computation.

Definition 2.1. Let $X = (I_X, \theta_X)$ be a decision problem, where I_X is a language over a finite alphabet (whose elements are called instances) and θ_X is a total boolean function over I_X (that is, a predicate). The decision problem X is solvable in polynomial time by a family $\Pi = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer tissue P systems with cell division if the following holds:

- The family Π is *polynomially uniform* by Turing machines, that is, there exists a deterministic Turing machine working in polynomial time which constructs the system $\Pi(n)$ from $n \in \mathbb{N}$.
- There exists a pair (cod, s) of polynomial-time computable functions over I_X such that:
 - for each instance $u \in I_X$, $s(u)$ is a natural number and $cod(u)$ is an input multiset of the system $\Pi(s(u))$;
 - the family Π is *polynomially bounded* with regard to (X, cod, s) , that is, there exists a polynomial function p , such that for each $u \in I_X$ every computation of $\Pi(s(u))$ with input $cod(u)$ halts and, moreover, it performs at most $p(s(u))$ steps;
 - the family Π is *sound* with regard to (X, cod, s) , that is, for each $u \in I_X$, if there exists an accepting computation of $\Pi(s(u))$ with input $cod(u)$, then $\theta_X(u) = 1$;
 - the family Π is *complete* with regard to (X, cod, s) , that is, for each $u \in I_X$, if $\theta_X(u) = 1$, then every computation of $\Pi(s(u))$ with input $cod(u)$ is an accepting one.

We denote by \mathbf{PMC}_{TDC} (respectively, $\mathbf{PMC}_{TDC(k)}$) the set of all decision problems which can be solved by means of recognizer tissue P systems with cell division in polynomial time (respectively, by using communication rules whose length is at most k).

3. A Uniform Solution to TMP

3.1. A Uniform Family of Tissue P Systems with Cell Division for Solving TMP

Let us consider an instance of TMP with three sets $B = \{b_1, \dots, b_n\}$, $G = \{g_1, \dots, g_n\}$, $H = \{h_1, \dots, h_n\}$, and a set T of triples $\subseteq B \times G \times H = \{(b_i, g_j, h_k) \mid 1 \leq i, j, k \leq n\}$. We address the solution of TMP via a brute force algorithm in the framework of recognizer tissue P systems with cell division. Our strategy consists of the following phases:

- *Generation Stage*: Cell division is applied to generate an exponential number of cells such that each possible subset of the set $B \times G \times H$ is encoded by exactly one cell.
- *Checking Stage*: The system checks whether or not there exists a subset T' of T such that $|T'| = n$ and no two triples belonging to T' have a component in common.
- *Output Stage*: The system sends to the environment the right answer according to the results of the previous stage.

We shall construct a family $\Pi = \{\Pi(n) \mid n \in \mathbb{N}\}$ such that each system $\Pi(n)$ will solve all instances of TMP with the size n of each of the finite sets B , G and H , provided that the appropriate input – the ternary relation $T \subseteq B \times G \times H$ – is given.

For each $n \in \mathbb{N}$, the system

$$\Pi(n) = (\Gamma(n), \Sigma(n), \Omega(n), w_1, w_2, \mathcal{R}(n), i_{in}, i_{out})$$

is constructed with the following components:

- $\Gamma(n) = \Sigma(n) \cup \{a_{i,j,k}, R_{i,j,k}, \bar{R}_{i,j,k}, \tilde{R}_{i,j,k}, R'_{i,j,k} \mid 1 \leq i, j, k \leq n\}$
 $\cup \{b_i \mid 1 \leq i \leq n^3 + 4n + 1\} \cup \{c_i \mid 1 \leq i \leq n^3 + 4n + 3\}$
 $\cup \{f, g, \text{yes}, \text{no}\}.$
- $\Sigma(n) = \{A_{i,j,k} \mid 1 \leq i, j, k \leq n\}.$
- $w_1 = \{\{b_1, c_1, g, \text{yes}, \text{no}\}\}.$
- $w_2 = \{\{a_{i,j,k} \mid 1 \leq i, j, k \leq n\}\} \cup \{\{f\}\}.$
- $\mathcal{R}(n)$ is the set of rules:

(1) **Division rule:**

$$r_{1,i,j,k} \equiv [a_{i,j,k}]_2 \rightarrow [R_{i,j,k}]_2[\lambda]_2, \text{ for } 1 \leq i, j, k \leq n.$$

(2) **Communication rules:**

$$r_{2,i} \equiv (1, b_i/b_{i+1}^2, 0), \text{ for } 1 \leq i \leq n^3;$$

$$r_{3,i} \equiv (1, c_i/c_{i+1}, 0), \text{ for } 1 \leq i \leq n^3 + 4n + 2;$$

$$r_4 \equiv (1, b_{n^3+1}/f, 2);$$

$$r_{5,i,j,k} \equiv (2, b_{n^3+i}R_{i,j,k}/b_{n^3+i+1}\bar{R}_{i,j,k}, 0), \text{ for } 1 \leq i, j, k \leq n;$$

$$r_{6,i,j,k} \equiv (2, b_{n^3+n+j}\tilde{R}_{i,j,k}/b_{n^3+n+j+1}\tilde{R}_{i,j,k}, 0), \text{ for } 1 \leq i, j, k \leq n;$$

$$r_{7,i,j,k} \equiv (2, b_{n^3+2n+k}\bar{R}_{i,j,k}/b_{n^3+2n+k+1}R'_{i,j,k}, 0), \text{ for } 1 \leq i, j, k \leq n;$$

$$r_{8,i,j,k} \equiv (2, b_{n^3+3n+i}R'_{i,j,k}A_{i,j,k}/b_{n^3+3n+i+1}, 0), \text{ for } 1 \leq i, j, k \leq n;$$

$$r_9 \equiv (2, b_{n^3+4n+1}/g \text{ yes}, 1);$$

$$r_{10} \equiv (2, \text{yes}/\lambda, 0);$$

$$r_{11} \equiv (1, c_{n^3+4n+3}g \text{ no}/\lambda, 2);$$

$$r_{12} \equiv (2, \text{no}/\lambda, 0).$$

- $\Omega(n) = \Gamma(n) - \{\text{yes}, \text{no}\}.$
- $i_{in} = 2$ is the input cell.
- $i_{out} = 0$ is the output region (i.e., the environment).

3.2. An Overview of a Computation

First of all we define a polynomial encoding for the tripartite matching problem in Π . Let $u = (B, G, H, T)$ an instance of the TMP. Let the size mapping be $s(u) = n$ and the encoding of instance be $cod(u) = \{A_{i,j,k} \mid (b_i, g_j, h_k) \in T\}$, for a given instance of tripartite matching problem $u = (B, G, H, T)$, $B = \{b_1, \dots, b_n\}$, $G = \{g_1, \dots, g_n\}$, $H = \{h_1, \dots, h_n\}$ and $T \subseteq B \times G \times H$. Next we informally describe how the system $\Pi(s(u))$ with input $cod(u)$ works.

Let us start with the generation stage. In cells with label 2, the division rules are applied. Cells with label 2 is repeatedly divided, each time expanding one object $a_{i,j,k}$, $1 \leq i, j, k \leq n$ into $R_{i,j,k}$ and λ , corresponding to the existence or absence of (b_i, g_j, h_k) in certain subset. In this way, after n^3 steps, 2^{n^3} cells with label 2 are generated, which represent all subsets of $B \times G \times H$. The object f is duplicated, hence a copy of it will appear in each cell. In parallel with the above operation of dividing cells with

label 2, the counters b_i, c_i from cell with label 1 grow their subscripts. Also, in each step, the number of copies of object b_i is doubled, hence after n^3 steps we get 2^{n^3} copies of b_{n^3+1} in cell 1. Objects b_i are used to check whether there exists a *perfect matching* $T' \subseteq T$ such that $|T'| = n$, and every element of B, G and H occurs exactly once in any triple of T' . The object c_i will be used to produce the object no, if this will not be the case, in the end of the computation.

The *checking stage* starts when the generation stage is finished after n^3 steps. Note that cells with label 2 cannot divide any more, because the objects $a_{i,j,k}$ were exhausted. At this moment, the content of the cell with label 1 is $\{\{b_{n^3+1}^{2^{n^3}}, c_{n^3+1}, g, \text{yes}, \text{no}\}\}$. In step $n^3 + 1$, a copy of the counter b_{n^3+1} is brought into each cell with label 2, in exchange of f by applying rule r_4 . Since we have 2^{n^3} copies of b_{n^3+1} and 2^{n^3} cells with label 2, each one containing exactly one copy of f , and due to the maximality of the parallelism of using the rules, each cell with label 2 gets precisely one copy of b_{n^3+1} .

In the presence of the object b_{n^3+1} , the rule $r_{5,i,j,k}$ is used to check if there exist n objects $R_{i,j,k}$ whose first subscripts i run from 1 to n in each cell with label 2. If and only if it is positive, the subscript of b_i grows to $n^3 + n + 1$, and the n objects $R_{i,j,k}$ are replaced by n objects $\bar{R}_{i,j,k}$. Then the rule $r_{6,i,j,k}$ is used to check if the second subscripts j of the n objects $\bar{R}_{i,j,k}$ run from 1 to n in each cell with label 2. If and only if it is positive, the subscript of b_i grows to $n^3 + 2n + 1$, and the n objects $\bar{R}_{i,j,k}$ are replaced by objects $\tilde{R}_{i,j,k}$. When the object b_{n^3+2n+1} is present, rule $r_{7,i,j,k}$ is used to check if the third subscripts k of n objects $\tilde{R}_{i,j,k}$ run from 1 to n in each cell with label 2. If and only if it is positive, the subscript of b_i grows to $n^3 + 3n + 1$, and the n objects $\tilde{R}_{i,j,k}$ are replaced by objects $R'_{i,j,k}$. After that, n objects $R'_{i,j,k}$ in each cell with label 2 constitute a possible subset T' of triples $\subseteq B \times G \times H$ such that $|T'| = n$, and every element of B, G and H occurs exactly once in any triple of T' . At last, rule $r_{8,i,j,k}$ is used to check if the given input T include any above generated subset T' . If it is positive, the subscript of b_i in the corresponding cell with label 2 grows to $n^3 + 4n + 1$.

When the checking stage is done, the subscript of object c_i in cell with label 1 grows to $n^3 + 4n + 2$. The output stage starts from step $n^3 + 4n + 2$.

- *Affirmative answer:* If there exists at least one subset T' of set T such that $|T'| = n$ and every element of B, G and H occurs exactly once in a triple of T' , there is an object b_{n^3+4n+1} in the corresponding cell with label 2 as described above. One of cells with label 2 containing object b_{n^3+4n+1} gets the objects yes and g in exchange of b_{n^3+4n+1} at step $n^3 + 4n + 2$. In the next step, the object yes in cell 2 leaves the system by the rule r_{10} , signaling the fact that there exists at least one subset T' of T such that $|T'| = n$, and no two triples belonging to T' have a component in common. At that step, the cell with label 1 contains the object c_{n^3+4n+3} but no the object g . That cell cannot evolve and the system halts at step $n^3 + 4n + 3$
- *Negative answer:* In this case, the subscript of the counter c_i reaches $n^3 + 4n + 3$ and the object g is still in the cell with label 1. The object no can be moved to the environment by the rules r_{11} and r_{12} , signaling that there does not exist any subset T' of T such that $|T'| = n$, and every element of B, G and H occurs exactly once in a triple of T' . The computation finishes at step $n^3 + 4n + 4$.

3.3. Informal Verification

In this subsection, we show that the family built above solves the tripartite matching problem in polynomial time, according to Definition 2.1. First of all, this definition requires that the defined family is

consistent, in the sense that all systems of the family must be recognizer tissue P systems with cell division. By construction (type of rules and working alphabet) it is clear that it is a family of tissue P systems with cell division.

It is easy to check that the rules of a system $\Pi(n)$ of the family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ are defined from the value n . Furthermore, the necessary resources to build an element of the family $\mathbf{\Pi}$ are of a polynomial order, as shown below:

- Size of the alphabet: $8n^3 + 8n + 8 \in \Theta(n^3)$.
- Initial number of cells: $2 \in \Theta(1)$.
- Initial number of objects: $n^3 + 6 \in \Theta(n^3)$.
- Number of rules: $7n^3 + 4n + 7 \in \Theta(n^3)$.
- Maximal length of a rule: $4 \in \Theta(1)$.

Therefore, a deterministic Turing machine can build $\Pi(n)$ in a polynomial time with respect to n ; that is, the family $\mathbf{\Pi}$ is polynomially uniform by Turing machines.

From the overview of a computation in section 3.2, we can find that all computations halt in a polynomial time with respect to n , and that either an object yes or an object no is sent out exactly in the last step of the computation; that is, the family $\mathbf{\Pi}$ is polynomially bounded, sound and complete.

3.4. Main Results

From the discussion in the previous sections (noting that the maximum length of communication rules is 4) and according to the definition of solvability given in Section 2, we have the following result:

Theorem 3.1. $\text{TMP} \in \text{PMC}_{TDC(4)}$.

Corollary 3.1. $\text{NP} \cup \text{co-NP} \subseteq \text{PMC}_{TDC(4)}$, where co-NP is the class of complements of NP problems.

Proof:

It suffices to make the following observations: the Tripartite Matching Problem is NP-complete, $\text{TMP} \in \text{PMC}_{TDC(4)}$ and this complexity class is closed under polynomial-time reduction and under complement [9]. □

We can check that the length of rules of the system given in Section 3 is not more than 3 except for the rules $r_{5,i,j,k}$, $r_{6,i,j,k}$, $r_{7,i,j,k}$, $r_{8,i,j,k}$. Let us replace $r_{5,i,j,k}$ by the rules $(2, b_{n^3+i}R_{i,j,k}/b_{n^3+i,j,k}^*, 0)$ and $(2, b_{n^3+i,j,k}^*/b_{n^3+i+1}\bar{R}_{i,j,k}, 0)$; replace $r_{6,i,j,k}$ by the rules $(2, b_{n^3+n+j}\bar{R}_{i,j,k}/b_{n^3+n+j,j,k}^*, 0)$ and $(2, b_{n^3+n+j,j,k}^*/b_{n^3+n+j+1}\tilde{R}_{i,j,k}, 0)$; replace $r_{7,i,j,k}$ by the rules $(2, b_{n^3+2n+k}\tilde{R}_{i,j,k}/b_{n^3+2n+k,j,k}^*, 0)$ and $(2, b_{n^3+2n+k,j,k}^*/b_{n^3+2n+k+1}R'_{i,j,k}, 0)$; replace $r_{8,i,j,k}$ by the rules $(2, R'_{i,j,k}A_{i,j,k}/A_{i,j,k}^*, 0)$ and the rules $(2, b_{n^3+3n+i,j,k}A_{i,j,k}^*/b_{n^3+3n+i+1}, 0)$. Of course, the subscripts of related objects such as counters b_i and c_i should be updated. In this way, a new system is obtained, where the length of all rules is not more than 3. It is not difficult to check that the new system can also solve the TMP in polynomial time. Therefore, Theorem 3.1 and Corollary 3.1 can be improved as follows.

Theorem 3.2. $\text{TMP} \in \text{PMC}_{\text{TDC}(3)}$.

Corollary 3.2. $\text{NP} \cup \text{co-NP} \subseteq \text{PMC}_{\text{TDC}(3)}$, where **co-NP** is the class of complements of **NP** problems.

4. Conclusions and Future Work

In this work, we have proposed a solution to the tripartite matching problem by a family of recognizer tissue P systems with cell division. The construction of tissue P systems with cell division is simple and elegant in terms of the number of the groups of rules. The results show that tissue-like P systems are suitable as a framework to address the efficient solution to intractable problems.

In the solution given in this work, the length of communication rules is at most 3. Let us consider a tissue P system with cell division and communication rules of length 1. In this case, each rule of the system can be activated by a single object. Hence, there exists, in some sense, a *dependency* between the object triggering the rule and the object or objects produced by its application. This dependency allows to adopt the ideas developed in [6] for cell-like P systems with active membranes to tissue P systems with cell division. In this way, it is not difficult to prove that tissue P systems with cell division and communication rules of length 1 can only solve problems solvable in polynomial time by deterministic Turing machines. It remains open whether tissue P systems with cell division and communication rules of length at most 2 can solve NP-complete problems. This open problem is worth further investigation. An answer to this open problem will provide a borderline between efficiency and non-efficiency in terms of the lengths of the communication rules.

Acknowledgements

Suggestions and comments made by three anonymous referees are gratefully acknowledged.

The authors wish to acknowledge the support of the project TIN2009-13192 of the Ministerio de Educación y Ciencia of Spain, cofinanced by FEDER funds, and the support of the Project of Excellence with Investigador de Reconocida Valía of the Junta de Andalucía, grant P08-TIC-04200. The first two authors are also supported by National Natural Science Foundation of China (61033003 and 30870826), the Fundamental Research Funds for the Central Universities (2010ZD001), Ph.D. Programs Foundation of Ministry of Education of China (20100142110072), and Natural Science Foundation of Hubei Province (2008CDB113 and 2008CDB180).

References

- [1] Alhazov, A., Martín Vide, C., Pan, L. Solving a PSPACE-complete problem by P systems with restricted active membranes. *Fundamenta Informaticae*, 58, 2(2003), 67–77.
- [2] Alhazov, A., Pérez-Jiménez, M.J. Uniform solution of QSAT using polarizationless active membranes. *Lecture Notes in Computer Science*, 4664(2007), 122–133.
- [3] Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A. A linear-time tissue P system based solution for the 3-coloring problem. *Electronic Notes in Theoretical Computer Science*, 171(2007), 81–93.

- [4] Díaz-Pernil, D., Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A. Solving subset sum in linear time by using tissue P systems with cell division. *Proc. of IWINAC 2007, LNCS 4527*, Springer, 170-179, 2007.
- [5] Díaz Pernil, D., Gutiérrez Naranjo, M.A., Pérez Jiménez, M.J., Riscos Núñez, A. Solving the partition problem by using tissue-like P systems with cell division. *Proc. of 6th Brainstorming Week on Membrane Computing, Report RGNC 01/08*, Fenix Editoria, 124-134, 2008.
- [6] Gutiérrez-Naranjo, M.A., Pérez-Jiménez, M.J., Riscos-Núñez, A., Romero-Campero, F.J. On the power of dissolution in P systems with active membranes, *Lecture Notes in Computer Science*, 3850(2006), 224–240.
- [7] Martín Vide, C., Pazos, J., Păun, Gh., Rodríguez Patón, A. Tissue P systems. *Theoretical Computer Science*, 296(2003), 295–326.
- [8] Pan, L., Martín Vide, C. Solving multidimensional 0-1 knapsack problem by P systems with input and active membranes. *Journal of Parallel and Distributed Computing*, 65(2005), 1578–1584.
- [9] Papadimitriou C.H. *Computational Complexity*. Addison-Wesley, Reading, Massachusetts (1994).
- [10] Păun, Gh. Computing with membranes. *Journal of Computer and System Sciences*, 61, 1(2000), 108–143.
- [11] Păun, Gh. *Membrane Computing. An Introduction*. Springer-Verlag, Berlin (2002).
- [12] Păun, Gh., Pérez-Jiménez, M.J. Riscos-Núñez, A. Tissue P systems with cell division. *Int. J. of Computers, Communications and Control*, 3, 3(2008), 295–303.
- [13] Păun, Gh., Rozenberg, G., Salomaa, A., eds. *Handbook of Membrane Computing*. Oxford University Press, 2009.
- [14] P systems web page <http://ppage.psystems.eu/>