SCIENCE CHINA
Information Sciences

Tissue P systems with cell separation: attacking the partition problem

XingYi ZHANG[1], Shuo WANG[2], YunYun NIU[2] and LinQiang PAN[2,*]

Citation: SCIENCE CHINA Information Sciences **54**, 293 (2011); doi: 10.1007/s11432-010-4162-y

View online: https://engine.scichina.com/doi/10.1007/s11432-010-4162-y

View Table of Contents: https://engine.scichina.com/publisher/scp/journal/SCIS/54/2

Published by the Science China Press

SCIENCE CHINA
Information Sciences

• **RESEARCH PAPERS** •

February 2011 Vol. 54 No. 2: 293–304
doi: 10.1007/s11432-010-4162-y

Tissue P systems with cell separation: attacking the partition problem

ZHANG XingYi[1], WANG Shuo[2], NIU YunYun[2] & PAN LinQiang[2]*

[1]*Key Lab of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei* 230039*, China;*
[2]*Image Processing and Intelligent Control Key Laboratory of Education Ministry of China, Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan* 430074*, China*

Received March 9, 2009; accepted August 23, 2010

Abstract Tissue P systems are distributed parallel and non-deterministic computing models in the framework of membrane computing, which are inspired by intercellular communication and cooperation between neurons. Recently, cell separation is introduced into tissue P systems, which enables systems to generate an exponential workspace in a polynomial time. In this work, the computational power of tissue P systems with cell separation is investigated. Specifically, a uniform family of tissue P systems with cell separation is constructed for efficiently solving a well-known NP-complete problem, the partition problem.

Keywords membrane computing, tissue P system, cell separation, partition problem

Citation Zhang X Y, Wang S, Niu Y Y, et al. Tissue P systems with cell separation: attacking the partition problem. Sci China Inf Sci, 2011, 54: 293–304, doi: 10.1007/s11432-010-4162-y

1 Introduction

Membrane computing is one of the recent branches of natural computing, which is inspired by the structure and the function of a single living cell, as well as from the organization of cells in tissues, organs, and other higher order structures. The devices in membrane computing, called P systems, provide distributed parallel and non-deterministic computing models. The field of membrane computing developed very rapidly (already in 2003, ISI considered membrane computing as a "fast emerging research area in computer science", see http://esi-topics.com). Please refer to the handbook of membrane computing [1] for general information in this area, and to the membrane computing web site [2] for the up-to-date information.

Since the first P system was introduced in [3], many different variants of P systems have been proposed. Tissue P systems are a class of variants of P systems, which are inspired by intercellular communication and cooperation between neurons [4]. In a tissue P system, membranes are placed in the nodes of a graph. The net of membranes (abstracted from cells) deals with symbols and communicates symbols along channels specified in advance. The communication among cells is based on symport/antiport

*Corresponding author (email: lqpan@mail.hust.edu.cn, lqpan@us.es)

rules [5]. Symport rules move objects across a membrane together in one direction, whereas antiport rules move objects across a membrane in opposite directions.

There exist several tissue-like P systems (see, e.g., [6–11]). An interesting variant of tissue P systems, called tissue P systems with cell division, was presented in [12] (and studied in depth in [13]). This computing model is endowed with the ability of getting new cells based on the mitosis or cellular division, thus having the ability of generating an exponential amount of workspace in a polynomial time. In the last years, the computational efficiency of tissue P systems with cell division has been widely investigated for solving computationally hard problems. The first attempt in this topic was done in [12], where it was shown that SAT problem could be efficiently solved by tissue P systems with cell division in a polynomial time. Since then, tissue P systems with cell division were also considered to solve other NP-complete problems: 3-coloring [14] (further studied in [15]), subset sum [16], vertex cover [17], partition problem [18], and so on. In all these works, the tissue P systems with cell division constructed for solving NP-complete problems are uniform, by which efficient solutions can be obtained in a polynomial time or even in a linear time.

Recently, a new variant of tissue P systems, called tissue P systems with cell separation, was proposed in [19]. Although this class of tissue P systems also has the ability of getting new cells, it is quite different from the class of tissue P systems with cell division. In a tissue P system with cell division, when cell division is used to generate an exponential workspace in a polynomial time, there is an advantage: all the other objects in a cell are duplicated except for the object that activates the cell division operation. But cell separation does not have such duplication function. This "disadvantage" of cell separation leads to a significant difference in specific techniques for designing P systems to solve NP-complete problems. It was shown that a polynomial-time solution for the SAT problem can be obtained by a uniform family of tissue P systems with cell separation [19]. In this work, a uniform family of tissue P systems with cell separation is constructed for efficiently solving a well-known NP-complete problem: the partition problem. In the literature, some uniform solutions to this problem were reported in the framework of P systems with active membranes or tissue-like P systems with cell division. However, this is the first solution to the partition problem in the framework of tissue-like P systems with cell separation.

The paper is organized as follows. In section 2, some preliminaries are recalled. The formal definition of tissue P systems with cell separation is given in section 3. In section 4, recognizer tissue P systems with cell separation are briefly described. A polynomial-time solution to the partition problem is presented in section 5, including a short overview of the computation and of the necessary resources. In section 6, the main results and some discussion are presented.

2 Preliminaries

An alphabet Σ is a non-empty set, whose elements are called symbols. An ordered sequence of symbols is a string. The number of symbols in a string u is the length of the string, and it is denoted by $|u|$. 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 $\Sigma^* = \cup_{n\geqslant 0}\Sigma^n$. A language over Σ is a subset of Σ^*.

A multiset m over a set A is a pair (A, f) where $f: A \to \mathbb{N}$ is a mapping. 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 (respectively, finite) if its support is the empty set (respectively, finite).

If $m = (A, f)$ is a finite multiset over A, and $supp(m) = \{a_1, \ldots, a_k\}$, then it will be denoted by $m = \{\{a_1^{f(a_1)}, \ldots, a_k^{f(a_k)}\}\}$; that is, superscripts indicate the multiplicity of each element. 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 function $m_1m_2 = (A, h)$, where $h = f + g$.

3 Tissue P systems with cell separation

Readers are assumed to be familiar with the basic notions and notations of P systems (see, e.g., [20]). Let us directly discuss tissue P systems.

In the tissue P systems given in [4, 21], the membrane structure does not change along the computation. Based on the cell-like P systems with membrane separation [22], a new class of P systems, called tissue P systems with cell separation, was recently presented in [19]. Tissue P systems with cell separation are inspired by the biological fact: alive tissues are not static network of cells, since new cells are generated by membrane fission in a natural way.

The main features of tissue P systems with cell separation, from the computational point of view, are that cells are not polarized (the opposite holds in the cell-like P systems with active membranes, see [20]); the cells obtained by separation have the same labels as the original cell. If a cell is separated, its interaction with other cells or with the environment is blocked during the separation process. This means that while a cell is separating it closes its communication channels.

Formally, a tissue P system with cell separation of degree $g \geqslant 1$ is a construct

$$\Pi = (\Gamma, O_1, O_2, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_0),$$

where

1. Γ is an alphabet whose elements are called objects, $\Gamma = O_1 \cup O_2$, $O_1, O_2 \neq \emptyset$, $O_1 \cap O_2 = \emptyset$;

2. $w_1, \ldots, w_q$ are strings over Γ, representing the multisets of objects placed in the q cells of the system at the beginning of the computation;

3. $\mathcal{E} \subseteq \Gamma$ is an alphabet representing the set of objects in the environment with arbitrary copies of each;

4. $\mathcal{R}$ is a finite set of rules of the following forms:

(a) $(i, u/v, j)$, for $i, j \in \{0, 1, 2, \ldots, q\}, i \neq j$, $u, v \in \Gamma^*$; communication rules: $1, 2, \cdots, q$ identify the cells of the system, 0 is the environment; when applying a rule $(i, u/v, j)$, 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 ($|u| + |v|$ is the length of the communication rule $(i, u/v, j)$);

(b) $[a]_i \rightarrow [O_1]_i[O_2]_i$, where $i \in \{1, 2, \ldots, q\}$, $a \in \Gamma$ and $i \neq i_0$; separation rules: with the presence of an object a, the cell is separated into two cells with the same label; at the same time, the object a is consumed; the objects from O_1 are placed in the first cell, those from O_2 are placed in the second cell; the output cell i_0 cannot be divided;

5. the output region i_0 is the environment.

The rules of a system like the ones above are used in the non-deterministic maximally parallel manner, as is customary in membrane computing. During each step, all cells which can evolve must evolve in a maximally parallel way (in each step the system applies a multiset of rules which is maximal: no further rule can be added). This way of applying rules has only one restriction: when a cell is separated, the separation rule is the only one which is applied to that cell during that step; the objects inside that cell do not evolve by means of communication rules. The new cells can participate in the interaction with other cells or the environment by means of communication rules in the next step—providing that they are not separated once again. Their labels precisely identify the rules which can be applied to them.

The configuration of a tissue P system with cell separation is described by the multisets of objects over Γ associated with all the cells present in the system and the multiset of objects over $\Gamma - \mathcal{E}$ associated with environment. Thus, the initial configuration of the system Π is the tuple $(w_1, w_2, \ldots, w_q; \emptyset)$. Using the rules as described above, one can define transitions among configurations. Any sequence of transitions starting from the initial configuration is called a computation. A computation halts if it reaches a configuration where no rule can be used. Only halting computations give a result, and the result is encoded by the multiset of objects over $\Gamma - \mathcal{E}$ appearing in the environment i_0 in the halting configuration.

4 Recognizer tissue P systems with cell separation

NP-completeness has usually been studied in the framework of decision problems, that is, pairs (I_X, θ_X) where I_X is a language over an alphabet (whose elements are called instances) and θ_X is a total Boolean function over I_X.

In order to study the computational efficiency, the notions from classical computational complexity theory are adapted for membrane computing. A class of cell-like P systems was introduced in [23]: recognizer P systems. For tissue P systems, with the same idea as for recognizer cell-like P systems, recognizer tissue P systems are introduced in [12].

A recognizer tissue P system with cell separation of degree $q \geqslant 1$ is a tuple

$$\Pi = (\Gamma, O_1, O_2, \Sigma, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_{in}, i_0),$$

where

• $(\Gamma, O_1, O_2, w_1, \ldots, w_q, \mathcal{E}, \mathcal{R}, i_o)$ is a tissue P system with cell separation of degree $q \geqslant 1$ (as defined in the previous section).

• The working alphabet Γ has two distinguished objects "yes" and "no", at least one copy of them present in some initial multisets $w_1, \ldots, w_q$, but none of them present in $\mathcal{E}$.

• Σ is an (input) alphabet strictly contained in Γ.

• $i_{in} \in \{1, \ldots, q\}$ is the input cell.

• The output region i_0 is the environment.

• All computations halt.

• If $\mathcal{C}$ is a computation of Π, then either object "yes" or object "no" (but not both) must have been released into the environment, and only at the last step of the computation.

The computations of the system Π with input $w \in \Sigma^*$ start from a configuration of the form $(w_1, w_2, \ldots, w_{i_{in}}w, \ldots, w_q; \emptyset)$. That is, the computations start after adding the multiset w to the content of the input cell i_{in}. A computation $\mathcal{C}$ is said to be an accepting one (respectively, rejecting one) if the object "yes" (respectively, "no") appears in the environment associated with the corresponding halting configuration of $\mathcal{C}$.

Definition 4.1. A decision problem $X = (I_X, \theta_X)$ is solvable in a polynomial time by a family $\mathbf{\Pi} = \{\Pi(n) \mid n \in \mathbb{N}\}$ of recognizer tissue P systems with cell separation if the following conditions hold:

• The family $\mathbf{\Pi}$ is polynomially uniform via Turing machines. That is, there exists a deterministic Turing machine working in a 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 $\mathbf{\Pi}$ 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 $cod(u)$ is halting and it performs at most $p(|u|)$ steps;

– the family $\mathbf{\Pi}$ 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 $\mathbf{\Pi}$ 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 PMC_{TS} the set of all decision problems which can be solved by means of recognizer tissue P systems with cell separation in polynomial time.

5 A solution for the partition problem

In this section, a solution to the partition problem is presented. Let us recall that a partition of a set V is a family of non-empty pairwise disjoint subsets of V such that the union of the subsets of the family is equal to V. The partition problem (PART, for short) can be stated as follows: let V be a finite set and w be a weight function on V, $w : V \to \mathbb{N}$ (that is, an additive function), determine whether or not there exists a partition $\{V_1, V_2\}$ of V such that $w(V_1) = w(V_2)$. This is a well-known NP-complete problem [24].

The solution proposed follows a brute force approach in the framework of recognizer tissue P systems with cell separation. This process consists of the following stages:

• Generation stage: All the possible subsets of V are generated, each subset is represented by a cell.

• Pre-checking stage: The weight of each subset of V is calculated.

• Checking stage: Each subset of V is checked to determine whether there exists one subset such that its weight equals to the weight of its complementary set.

• Output stage: The system sends to the environment the right answer according to the results of the previous stage.

In what follows the instance of PART $u = (V, w)$ will be considered, where $V = \{v_1, v_2, \ldots, v_n\}$ and w is defined as $w_i = w(v_i)$ for each $i \in \{1, 2, \ldots, n\}$. Such instance will also be represented by $u = (n, (w_1, w_2, \ldots, w_n))$. We denote that $m = w_1 + w_2 + \ldots + w_n$.

The following polynomial encoding (cod, s) will be considered: for each instance $u = (n, (w_1, \ldots, w_n))$, $s(u)$ is defined as $s(u) = \langle n, w_1 + \cdots + w_n\rangle$ and $cod(u)$ is defined as $cod(u) = k_1^{w_1+1} v_1^{w_1} \cdots k_n^{w_n+1} v_n^{w_n}$, where $\langle x, y\rangle = ((x+y)(x+y+1)/2) + x$ is a polynomial-time computable pair function between $\mathbb{N}^2$ and $\mathbb{N}$.

For given $n, m \in \mathbb{N}$, a family of recognizer tissue P systems with cell separation $\mathbf{\Pi} = \{\Pi(i) \mid i \in \mathbb{N}\}$ will be constructed such that each system $\Pi(i)$ will solve all instances of PART with the size parameters n and m, where $i = \langle n, m\rangle$, provided that the appropriate input multisets are given.

For each $n, m \in \mathbb{N}$,

$$\Pi(\langle n,m\rangle) = (\varGamma(\langle n,m\rangle), \varSigma(\langle n,m\rangle), w_1, w_2, \mathcal{R}(\langle n,m\rangle), \mathcal{E}(\langle n,m\rangle), i_{in}, i_0),$$

with the following components:

• $\varGamma(\langle n,m\rangle) = O_1 \cup O_2$,

$$\begin{aligned}
O_1 &= \{k_i, \bar{k}_i, v_i, \bar{A}_i, B_i, z_i, B_i'', D_{2,i}, c_{1,i}, c_{2,i} \mid 1 \leqslant i \leqslant n\}\\
&\cup\{A_i \mid -2 \leqslant i \leqslant n\}\\
&\cup\{a_{1,i}, b_{1,i}, h_i, \bar{B}_i \mid 1 \leqslant i \leqslant n-1\}\\
&\cup\{d_{2,i} \mid 2 \leqslant i \leqslant n-2\}\\
&\cup\{B_{i,j}, A_{i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant \lceil lgm\rceil + 1\}\\
&\cup\{a_{i,j,k}, b_{i,j,k} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m+n, 1 \leqslant k \leqslant n\}\\
&\cup\{a_{2,i}, b_{2,i} \mid 2 \leqslant i \leqslant n-1\}\\
&\cup\{d_{i,j,k} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m+n, 1 \leqslant k \leqslant n-1\}\\
&\cup\{D_{1,i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m+n\}\\
&\cup\{a_i \mid 1 \leqslant i \leqslant 3n + \lceil lgn\rceil + \lceil lgm\rceil + 16\}\\
&\cup\{H_i \mid 1 \leqslant i \leqslant \lceil lgn\rceil + \lceil lgm\rceil + 4\}\\
&\cup\{d_{1,i}, e_i, l_i \mid 1 \leqslant i \leqslant n-2\}\\
&\cup\{d_i \mid 1 \leqslant i \leqslant \lceil lgn\rceil + 1\}\\
&\cup\{g_i \mid -2 \leqslant i \leqslant n-1\}\\
&\cup\{G_i \mid 1 \leqslant i \leqslant \lceil lgn\rceil + 1\}\\
&\cup\{c_{3,i} \mid 2 \leqslant i \leqslant n+1\}\\
&\cup\{a_{-2,1}, a_{-1,1}, a_{0,1}, c_{-2,1}, c_{-1,1}, c_{0,1}, z, y, s, c, c_{3n}, b, p, q\}\\
&\cup\{D_1, E_1, F_1, F_2, T, Y, N, \text{yes}, \text{no}\},\\
O_2 &= \{\bar{A}'_i, z_i', v_i', \bar{B}_i' \mid 1 \leqslant i \leqslant n\} \cup \{y', z'\} \cup \{B_i' \mid 1 \leqslant i \leqslant n-1\}.
\end{aligned}$$

• $\varSigma(\langle n,m\rangle) = \{k_i, v_i \mid 1 \leqslant i \leqslant n\}$.

• $w_1 = \{\{a_1, a_{-2,1}, g_{-2}, c_{-2,1}, b, \text{yes}, \text{no}\}\} \cup \{\{a_{i,j,1}, D_{1,i,j} \mid 1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m+n\}\}$.

• $w_2 = \{\{A_{-2}, \bar{A}_1, \ldots, \bar{A}_n\}\}$.

• $\mathcal{E}(\langle n,m\rangle) = \varGamma(\langle n,m\rangle) - \{\text{yes}, \text{no}\}$.

• $i_{\text{in}} = 2$ is the input cell.

• $i_0 = 0$ is the output region.

• $\mathcal{R}(\langle n,k\rangle)$ is the set of rules:

(1) Separation rule:

$r_1 \equiv [s]_2 \rightarrow [O_1]_2[O_2]_2$.

(2) Communication rules:

$r_2 \equiv (1, a_{-2,1}/a_{-1,1}, 0)$;
$r_3 \equiv (1, a_{-1,1}/a_{0,1}, 0)$;
$r_4 \equiv (1, a_{0,1}/a_{1,1}, 0)$;
$r_{5,i} \equiv (1, a_{1,i}/b_{1,i}, 0)$ for $1 \leqslant i \leqslant n-1$;
$r_{6,i} \equiv (1, b_{1,i}/c^2 d_{1,i}^2 e_i^2, 0)$ for $1 \leqslant i \leqslant n-2$;
$r_7 \equiv (1, b_{1,n-1}/c^2, 0)$;
$r_{8,i} \equiv (1, d_{1,i}/a_{1,i+1}, 0)$ for $1 \leqslant i \leqslant n-2$;
$r_{9,i} \equiv (1, e_i/a_{2,i+1}, 0)$ for $1 \leqslant i \leqslant n-2$;
$r_{10,i} \equiv (1, a_{2,i}/b_{2,i}, 0)$ for $2 \leqslant i \leqslant n-1$;
$r_{11,i} \equiv (1, b_{2,i}/c^2 d_{2,i}^2, 0)$ for $2 \leqslant i \leqslant n-2$;
$r_{12} \equiv (1, b_{2,n-1}/c^2, 0)$;
$r_{13,i} \equiv (1, d_{2,i}/a_{2,i+1}, 0)$ for $2 \leqslant i \leqslant n-2$;
$r_{14} \equiv (1, g_{-2}/g_{-1}, 0)$;
$r_{15} \equiv (1, g_{-1}/g_0, 0)$;
$r_{16} \equiv (1, g_0/g_1, 0)$;
$r_{17,i} \equiv (1, g_i/h_i, 0)$ for $1 \leqslant i \leqslant n-1$;
$r_{18,i} \equiv (1, h_i/l_i^2 A_{i+1}^2, 0)$ for $1 \leqslant i \leqslant n-2$;
$r_{19} \equiv (1, h_{n-1}/A_n^2, 0)$;
$r_{20,i} \equiv (1, l_i/g_{i+1}, 0)$ for $1 \leqslant i \leqslant n-2$;
$r_{21} \equiv (2, A_{-2}/A_{-1}, 0)$;
$r_{22} \equiv (2, A_{-1}/A_0, 0)$;
$r_{23} \equiv (2, A_0/A_1, 0)$;
$r_{24,i} \equiv (2, A_i/B_i\bar{B}_i' zz' yy' s, 0)$ for $1 \leqslant i \leqslant n-1$;
$r_{25} \equiv (2, A_n/B_n\bar{B}_n' yy' s, 0)$;
$r_{26,i} \equiv (2, cB_i/zz' B_i B_i', 0)$ for $1 \leqslant i \leqslant n-1$;
$r_{27,i} \equiv (2, cB_i'/zz' B_i B_i', 0)$ for $1 \leqslant i \leqslant n-1$;
$r_{28,i} \equiv (2, c\bar{B}_i/zz' \bar{B}_i \bar{B}_i', 0)$ for $1 \leqslant i \leqslant n-1$;
$r_{29,i} \equiv (2, c\bar{B}_i'/zz' \bar{B}_i \bar{B}_i', 0)$ for $1 \leqslant i \leqslant n-1$;
$r_{30} \equiv (1, c/z, 2)$;
$r_{31} \equiv (1, c/z', 2)$;
$r_{32,i} \equiv (1, A_i/y, 2)$ for $2 \leqslant i \leqslant n$;
$r_{33,i} \equiv (1, A_i/y', 2)$ for $2 \leqslant i \leqslant n$;
$r_{34} \equiv (1, y/\lambda, 0)$;
$r_{35} \equiv (1, y'/\lambda, 0)$;
$r_{36} \equiv (1, z/\lambda, 0)$;
$r_{37} \equiv (1, z'/\lambda, 0)$;
$r_{38,i,j,k} \equiv (1, a_{i,j,k}/b_{i,j,k}, 0)$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m+n$, $1 \leqslant k \leqslant n$;
$r_{39,i,j,k} \equiv (1, b_{i,j,k}k_i/d_{i,j,k}^2 k_i^2 \bar{k}_i, 0)$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m+n$, $1 \leqslant k \leqslant n-1$;
$r_{40,i,j} \equiv (1, b_{i,j,n}k_i/\bar{k}_i, 0)$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m+n$;
$r_{41,i,j,k} \equiv (1, d_{i,j,k}/a_{i,j,k+1}, 0)$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m+n$, $1 \leqslant k \leqslant n-1$;
$r_{42,i,j} \equiv (1, D_{1,i,j}/k_i, 2)$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m+n$;
$r_{43,i,j} \equiv (2, D_{1,i,j}/D_{2,i}, 0)$ for $1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m+n$;
$r_{44,i} \equiv (2, D_{2,i}/\bar{k}_i, 1)$ for $1 \leqslant i \leqslant n$;
$r_{45,i} \equiv (1, D_{2,i}/\lambda, 0)$ for $1 \leqslant i \leqslant n$;
$r_{46,i} \equiv (2, \bar{k}_i\bar{A}_i/z_i z_i' \bar{A}_i \bar{A}_i', 0)$ for $1 \leqslant i \leqslant n$;
$r_{47,i} \equiv (2, \bar{k}_i\bar{A}_i'/z_i z_i' \bar{A}_i \bar{A}_i', 0)$ for $1 \leqslant i \leqslant n$;
$r_{48,i} \equiv (2, \bar{k}_i v_i/z_i z_i' v_i v_i', 0)$ for $1 \leqslant i \leqslant n$;

$r_{49,i} \equiv (2, \bar{k}_i v_i'/z_i z_i' v_i v_i', 0)$ for $1 \leqslant i \leqslant n$;
$r_{50,i} \equiv (1, \bar{k}_i/z_i, 2)$ for $1 \leqslant i \leqslant n$;
$r_{51,i} \equiv (1, \bar{k}_i/z_i', 2)$ for $1 \leqslant i \leqslant n$;
$r_{52,i} \equiv (1, z_i/\lambda, 0)$ for $1 \leqslant i \leqslant n$;
$r_{53,i} \equiv (1, z_i'/\lambda, 0)$ for $1 \leqslant i \leqslant n$;
$r_{54,i} \equiv (1, a_i/a_{i+1}, 0)$ for $i = 1, \ldots, 3n+3+\lceil lgn \rceil + \lceil lgm \rceil + 12$;
$r_{55} \equiv (1, c_{-2,1}/c_{-1,1}, 0)$;
$r_{56} \equiv (1, c_{-1,1}/c_{0,1}, 0)$;
$r_{57} \equiv (1, c_{0,1}/c_{1,1}, 0)$;
$r_{58,i} \equiv (1, c_{1,i}/c_{2,i}, 0)$ for $1 \leqslant i \leqslant n$;
$r_{59,i} \equiv (1, c_{2,i}/c_{3,i+1}^2, 0)$ for $1 \leqslant i \leqslant n$;
$r_{60,i} \equiv (1, c_{3,i}/c_{1,i}, 0)$ for $2 \leqslant i \leqslant n$;
$r_{61} \equiv (1, c_{3,n+1}/c_{n+1}, 0)$;
$r_{62} \equiv (1, c_{n+1}/y, 2)$;
$r_{63} \equiv (1, c_{n+1}/y', 2)$;
$r_{64} \equiv (2, c_{n+1}/D_1 G_1, 0)$;
$r_{65,i} \equiv (2, G_i/G_{i+1}^2, 0)$ for $i = 1, 2, \ldots, \lceil lgn \rceil$;
$r_{66} \equiv (2, D_1/d_1 H_1, 0)$;
$r_{67,i} \equiv (2, d_i/d_{i+1}^2, 0)$ for $i = 1, \ldots, \lceil lgn \rceil$;
$r_{68,i} \equiv (2, H_i/H_{i+1}, 0)$ for $i = 1, 2, \ldots, \lceil lgn \rceil + \lceil lgm \rceil + 3$;
$r_{69,i} \equiv (2, G_{\lceil lgn \rceil+1} B_i/B_i'', 0)$ for $1 \leqslant i \leqslant n$;
$r_{70,i} \equiv (2, G_{\lceil lgn \rceil+1} B_i'/B_i'', 0)$ for $1 \leqslant i \leqslant n$;
$r_{71,i} \equiv (2, B_i'' \bar{A}_i/B_{i,1}, 0)$ for $1 \leqslant i \leqslant n$;
$r_{72,i} \equiv (2, B_i'' \bar{A}_i'/B_{i,1}, 0)$ for $1 \leqslant i \leqslant n$;
$r_{73,i} \equiv (2, d_{\lceil lgn \rceil+2} \bar{A}_i/A_{i,1}, 0)$ for $1 \leqslant i \leqslant n$;
$r_{74,i} \equiv (2, d_{\lceil lgn \rceil+2} \bar{A}_i'/A_{i,1}, 0)$ for $1 \leqslant i \leqslant n$;
$r_{75,i,j} \equiv (2, B_{i,j}/B_{i,j+1}^2, 0)$ for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, \lceil lgm \rceil$;
$r_{76,i,j} \equiv (2, A_{i,j}/A_{i,j+1}^2, 0)$ for $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, \lceil lgm \rceil$;
$r_{77,i} \equiv (2, B_{i,\lceil lgm \rceil+1} v_i/p, 0)$ for $1 \leqslant i \leqslant n$;
$r_{78,i} \equiv (2, B_{i,\lceil lgm \rceil+1} v_i'/p, 0)$ for $1 \leqslant i \leqslant n$;
$r_{79,i} \equiv (2, A_{i,\lceil lgm \rceil+1} v_i/q, 0)$ for $1 \leqslant i \leqslant n$;
$r_{80,i} \equiv (2, A_{i,\lceil lgm \rceil+1} v_i'/q, 0)$ for $1 \leqslant i \leqslant n$;
$r_{81} \equiv (2, pq/\lambda, 0)$;
$r_{82} \equiv (2, H_{\lceil lgn \rceil+\lceil lgm \rceil+4}/E_1 F_1, 0)$;
$r_{83} \equiv (2, E_1 p/\lambda, 0)$;
$r_{84} \equiv (2, E_1 q/\lambda, 0)$;
$r_{85} \equiv (2, F_1/F_2, 0)$;
$r_{86} \equiv (2, E_1 F_2/T, 0)$;
$r_{87} \equiv (2, T/\lambda, 1)$;
$r_{88} \equiv (1, bT/Y, 0)$;
$r_{89} \equiv (1, Y\,\texttt{yes}/\lambda, 0)$;
$r_{90} \equiv (1, a_{3n+3+\lceil\lceil lgn \rceil+\lceil lgm \rceil+13} b/N, 0)$;
$r_{91} \equiv (1, \texttt{no}\ N/\lambda, 0)$.

5.1 An overview of the computation

A family of recognizer tissue P systems with cell separation is constructed as above. Let $u = (V, w)$ be an instance of PART, where $V = \{v_1, v_2, \ldots, v_n\}$ and w is defined as $w_i = w(v_i)$, for each $i \in \{1, 2, \ldots, n\}$. A size mapping on the set of instances is defined as $s(u) = \langle n, m \rangle$, where $m = w_1 + w_2 + \cdots + w_n$. The encoding of the instance is the multiset $cod(u) = k_1^{w_1+1} v_1^{w_1} \ldots k_n^{w_n+1} v_n^{w_n}$.

Now, we informally describe how the recognizer tissue P system with cell separation $\Pi(s(u))$ with input $cod(u)$ works.

Let us start with the generation stage. This stage has several parallel processes, which are described in several items.

1) In the cells with label 2, with the presence of $\bar{k}_i$, by the rules $r_{46,i} - r_{49,i}$, the objects $\bar{k}_i\bar{A}_i$, $\bar{k}_i\bar{A}'_i$, $\bar{k}_iv_i$, $\bar{k}_iv'_i$ introduce the objects $z_iz'_i\bar{A}_i\bar{A}'_i$, $z_iz'_i\bar{A}_i\bar{A}'_i$, $z_iz'_iv_iv'_i$, $z_iz'_iv_iv'_i$, respectively. In the next step, the objects with prime and the objects without prime are separated into the new daughter cells with label 2. The idea is that $\bar{k}_i$ is used to duplicate $\bar{A}_i$ and v_i (in the sense ignoring the prime), so that one copy of each of them will appear in each cell with label 2. The objects z_i and z'_i in the cells with label 2 are exchanged with the object $\bar{k}_i$ in the cell with label 1 by the rules $r_{50,i}$ and $r_{51,i}$. In this way, the cycle of duplication-separation can be iterated.

2) In parallel with the above duplication-separation process, the object c is used to duplicate the objects B_i, B'_i, $\bar{B}_i$, $\bar{B}'_i$ by the rules $r_{26,i}$–$r_{29,i}$. The rules r_{30} and r_{31} take care of introducing the object c from the cell with label 1 to the cells with label 2.

3) In the initial configuration of the system, the cell with label 2 contains an object A_{-2}. After three steps, the object A_{-2} evolves to object A_1 ($A_i, 1 \leqslant i \leqslant n$, are used to generate all possible 2^n subsets of V and each possible subset corresponds to a cell with label 2, thus generating 2^n cells with label 2 in total). The objects B_1, $\bar{B}'_1$, z, z', y, y' and s are brought into the cell with label 2, in exchange with A_1, by the rule $r_{24,1}$. If the object B_1 or B'_1 occurs in one cell with label 2, then its corresponding subset contains the element v_1; if the object $\bar{B}_1$ or $\bar{B}'_1$ occurs in one cell with label 2, then its corresponding subset does not contain the element v_1 (in general, B_i, B'_i, $\bar{B}_i$ and $\bar{B}'_i$ are used to indicate the element v_i). In the next step, they are separated into the new cells with label 2 by the separation rule, because $B_1, \bar{B}_1 \in O_1$ and $B'_1, \bar{B}'_1 \in O_2$. The object s is used to activate the separation rule r_1, and is consumed during the application of separation rule. The objects y and y' are used to introduce A_2 from the cell with label 1, thus the process for the case of the element v_2 can continue. In this way, after $3n+2$ steps 2^n cells with label 2 can be generated, and each cell contains one of the 2^n possible subsets of V.

4) In parallel with the operations in the cells with label 2, each object $a_{1,i+1}$ from the cell with label 1 is exchanged with one copy of object $b_{1,i+1}$ from the environment at the step $3i+4$ ($0 \leqslant i \leqslant n-2$) by the rule $r_{5,i}$ (after the first three steps, object $a_{-2,1}$ evolves to $a_{1,1}$). In the next step, each object $b_{1,i+1}$ is exchanged with two copies of objects c, $d_{1,i+1}$ and e_{i+1} by the rule $r_{6,i+1}$. At step $3i+6$ ($0 \leqslant i \leqslant n-2$), each object $d_{1,i+1}$ is exchanged with one copy of object $a_{1,i+2}$ by the rule $r_{8,i+1}$, and each object e_i is exchanged with one copy of object $a_{2,i+2}$ by the rule $r_{9,i+1}$. In the next step, by using the rules $r_{5,i+2}$ and $r_{10,i+2}$, each object $a_{1,i+2}$ is exchanged with one copy of object $b_{1,i+2}$ and each object $a_{2,i+2}$ is exchanged with one copy of object $b_{2,i+2}$. At step $3(i+1)+5$, each object $b_{1,i+2}$ is exchanged with two copies of objects c, $d_{1,i+2}$ and e_{i+2} by using the rule $r_{6,i+2}$. At the same time, each object $b_{2,i+2}$ is exchanged with two copies of objects c and $d_{2,i+2}$ by using the rule $r_{11,i+2}$. In particular, at step $3n-1$, each object $b_{1,n-1}$ is exchanged with only two copies of object c by the rule r_7 and each object $b_{2,n-1}$ is exchanged with only two copies of object c by the rule r_{12}. After step $3n-1$, there is no object $a_{1,i}$ appearing in the cell with label 1, and the group of rules $r_{5,i} - r_{9,i}$ will not be used again. This is also the case for object $a_{2,i}$ and the group of rules $r_{10,i} - r_{13,i}$ will not be used again. Note that the subscript i of object $a_{1,i}$ grows by one in every three steps until it becomes the value $n-1$, and the number of copies of $a_{1,i}$ is doubled in every three steps. This is also the case for object $a_{2,i}$. At step $3i+5$ ($0 \leqslant i \leqslant n-2$), the cell with label 1 has $(i+1)\cdot 2^{i+1}$ copies of object c (the cell with label 1 contains $(n-1)\cdot 2^{n-1}$ copies of object c at step $3n-1$). At the same time, there are 2^{i+1} cells with label 2, and each cell with label 2 totally contains $i+1$ copies of object z or z'. Due to the maximality of the parallelism of using the rules, each cell with label 2 gets exactly $i+1$ copies of c from the cell with label 1 by the rules r_{30} and r_{31}. The object c in the cells with label 2 is used for duplication as described above.

5) The object $a_{i,j,k}$ in the cell with label 1 has a similar role to object $a_{1,i}$ in the cell with label 1, which introduces appropriate copies of object $\bar{k}_i$ for the duplication of objects $\bar{A}_i$, $\bar{A}'_i$, v_i, and v'_i by the rules $r_{46,i} - r_{49,i}$. To this aim, by using the rule $r_{42,i,j}$, all the objects k_i ($1 \leqslant i \leqslant n$) which are used in the encoding of the instance are introduced in the cell with label 1 from the cell with label 2. After this operation, it is possible that there still exist objects $D_{1,i,j}$ ($1 \leqslant i \leqslant n$, $1 \leqslant j \leqslant m+n$) in the cell with label 1, but they will not be used again, because there is no object k_i appearing in the cells with

label 2 (object $\bar{k}_i$ is used to duplicate $\bar{A}_i$, $\bar{A}'_i$, v_i and v'_i). Similarly, in every three steps, each object k_i is exchanged with two copies of k_i and one copy of $\bar{k}_i$ by using the rules $r_{38,i,j,k} - r_{41,i,j,k}$. All copies of object k_i stay in the cell with label 1 and they are used to continue to generate $\bar{k}_i$, each object $\bar{k}_i$ is introduced into the cells with label 2 to duplicate $\bar{A}_i$, $\bar{A}'_i$, v_i, and v'_i by the rules $r_{50,i}$ and $r_{51,i}$. Note that at step $3j + 5$ $(0 \leqslant j \leqslant n-2)$, there are $(w_i + 1) \cdot 2^{j+1}$ copies of object $\bar{k}_i$, which ensures that each cell with label 2 gets exactly $w_i + 1$ copies of object $\bar{k}_i$ by the maximality of the parallelism of using the rules. This process can continue until all 2^n cells with label 2 are generated and each of them gets one copy of $\bar{A}_i$, as well as w_i copies of v_i, for all $i \in \{1, 2, \ldots, n\}$ (ignoring the prime). After the last duplication finishes (at step $3n + 1$), object z_i or z_i will not be removed from each cell with label 2, by exchanging with $\bar{k}_i$. Because these objects will not used anymore, in what follows they are ignored when the computation is discussed.

6) After the first three steps, the object g_{-2} evolves to g_1 in the cell with label 1. Starting from the subscript $i = 1$, each object g_{i+1} in the cell with label 1 is exchanged with one copy of h_{i+1} from the environment at step $3i + 4$ by the rule $r_{17,i+1}$, $0 \leqslant i \leqslant n-2$. In the next step, each object h_{i+1} is exchanged with two copies of objects l_{i+1} and A_{i+2} by the rule $r_{18,i+1}$. At step $3i + 6$ $(0 \leqslant i \leqslant n-2)$, each object l_{i+1} is exchanged with two copies of g_{i+2}, thus the process can be iterated until the subscript i of g_i becomes $n - 1$. Specially, at step $3n - 2$, each object g_{n-1} is exchanged with one copy of h_{n-1} by the rule $r_{17,i}$. At step $3n - 1$, each object h_{n-1} is exchanged with only two copies of A_n. After step $3n - 1$, there is no object g_i in the cell with label 1, and the group of rules $r_{17,i} - r_{20,i}$ will not be used again. At step $3i + 5$ $(0 \leqslant i \leqslant n-2)$, the cell with label 1 contains 2^{i+1} copies of A_{i+2}, and there are 2^{i+1} cells with label 2, each of them contains one copy of object y or y'. Due to the maximality of the parallelism of using the rules, each cell with label 2 gets exactly one copy of A_{i+2} from the cell with label 1 by the rule $r_{32,i}$ or $r_{33,i}$. In this way, the system can continue to generate the possible cases of element v_{i+2}.

7) The counter a_i in the cell with label 1 grows its subscript by the rule $r_{54,i}$. Hence, when the process of generating all the possible subsets of V finishes at step $3n + 2$, the counter a_i evolves to a_{3n+3}. It will continue to grow its subscript until it evolves to $a_{3n+3+\lceil lgn \rceil + \lceil lgm \rceil + 13}$, which together with b will be used to introduce the object N in the cell with label 1, then bring object "no" to the environment, if this will be the case, in the end of the computation. Object c_{n+1} is used to generate the objects which are useful in the following stages. After the first three steps, the object $c_{-2,1}$ evolves to $c_{1,1}$ by the rules $r_{55} - r_{57}$. Starting from the subscript $i = 1$, in every three steps each object $c_{1,i}$ is exchanged with two copies of object $c_{3,i+1}$ by using the rules $r_{58,i} - r_{60,i}$. Hence, after step $3n + 2$, the cell with label 1 contains 2^n copies of $c_{3,n+1}$. In the next step, each object $c_{3,n+1}$ is exchanged with one copy of c_{n+1}. This means that at step $3n + 4$ each cell with label 2 gets one copy of c_{n+1} (at this step the system has 2^n cells with label 2).

8) The objects z_i, z'_i, y, y', z and z' in the cell with label 1 are removed by the rules r_{34}–r_{37} and $r_{52,i}$–$r_{53,i}$. (Actually, if the objects z_i, z'_i, y, y', z and z' stay in the cell with label 1, they do not influence the work of the system.)

In this way, after the $(3n + 4)$-step the generation stage finishes and the pre-checking stage starts. At this moment, the cell with label 1 contains one copy of objects a_{3n+5}, b, "yes" and "no" (there are other objects in the cell with label 1, but they are not used again, so they can be ignored), and there are 2^n cells with label 2, each of them corresponding to a possible subset of V which is represented by objects B_i or B'_i $(1 \leqslant i \leqslant n)$, and also containing one copy of object c_{n+1}, as well as the encoding of the instance. Actually, the cells with label 2 do not contain all the objects of the encoding of the instance. It contains w_i copies of object v_i or v'_i, and one copy of object $\bar{A}_i$ or $\bar{A}'_i$ for each $i \in \{1, 2, \ldots, n\}$. If the prime and the object k_i are ignored, then they are the encoding of the instance. Note that at step $(3n + 4)$ each cell with label 2 contains w_i copies of object z_i or z'_i instead of $\bar{k}_i$, $1 \leqslant i \leqslant n$.

At step $3n + 5$, in each cell with label 2 the object c_{n+1} is exchanged with one copy of D_1 and G_1 by the rule r_{64}, thus the counter G_i is introduced in the cell. Object D_1 is exchanged with the counters d_i and H_i at step $3n + 6$ by the rule r_{66}. From step $3n + 6$ to step $3n + 5 + \lceil lgn \rceil$, the counter G_i is doubled in each step by the rule $r_{64,i}$ until at least n copies of object $G_{\lceil lgn \rceil + 1}$ are produced. In the next step, the object B_i or B_i' in each cell with label 2 is exchanged with object B_i'' from the environment (by using the rules $r_{69,i}$ and $r_{70,i}$). At step $3n + 7 + \lceil lgn \rceil$, each pair of objects B_i'' and $\bar{A}_i$ (or $\bar{A}_i'$) that appear in the cells with label 2 is exchanged with one copy of object B_{i1} by applying the rule $r_{71,i}$ or $r_{72,i}$.

In parallel with the previous operations, the counter H_i grows its subscript until the subscript becomes the value $\lceil lgn \rceil + \lceil lgm \rceil + 4$ by the rule $r_{68,i}$. This operation is done from step $3n+7$ to step $3n + \lceil lgn \rceil + \lceil lgm \rceil + 9$ in all cells with label 2. Moreover, from step $3n + 7$ to step $3n + 6 + \lceil lgn \rceil$ the counter d_i is doubled in each step by the rule $r_{67,i}$ until at least n copies of object $d_{\lceil lgn \rceil+1}$ are produced. Thus, when the above process finishes, each cell with label 2 contains at least n copies of object $d_{\lceil lgn \rceil+1}$. The appearance of object $d_{\lceil lgn \rceil+1}$ triggers the rules $r_{73,i}$ and $r_{74,i}$, hence all the remaining objects $\bar{A}_i$ or $\bar{A}_i'$ in the cells with label 2 are exchanged with one copy of objects $A_{i,1}$, $1 \leqslant i \leqslant n$.

In this way, each cell with label 2 contains a pair of complementary subsets, encoded by objects $B_{i,1}$ and $A_{i,1}$, respectively. Because there are 2^n cells with label 2 at this moment, all possible partitions of V are generated.

From step $3n + 8 + \lceil lgn \rceil$ to step $3n + 7 + \lceil lgn \rceil + \lceil lgm \rceil$, by using the rules $r_{75,i,j}$ and $r_{76,i,j}$, in every step the subscripts of objects $B_{i,1}$ and $A_{i,1}$ increase by one, at the same time their numbers are doubled. Thus, at step $3n + 7 + \lceil lgn \rceil + \lceil lgm \rceil$, there are at least m copies of objects $B_{i,\lceil lgm \rceil+1}$ and $A_{i,\lceil lgm \rceil+1}$ ($1 \leqslant i \leqslant n$), where $m = w_1 + \ldots + w_n$ represents the total weight of the set V.

In order to obtain the weight of the subsets, each pair of objects $B_{i,\lceil lgn \rceil+1}$ and v_i (or v_i') in the cells with label 2 is exchanged with one copy of object p by using the rule $r_{77,i}$ (or $r_{78,i}$). Thus, in a cell with label 2 the weight of its corresponding subset is the total number of object p. Simultaneously, each pair of objects $A_{i,\lceil lgn \rceil+1}$ and v_i (or v_i') in the cells with label 2 is exchanged with one copy of object q by using the rule $r_{79,i}$ (or $r_{80,i}$), and the total number of object q represents the weight of its complementary subset.

The checking stage starts from the step $3n + 9 + \lceil lgn \rceil + \lceil lgm \rceil$, where the rule r_{81} is used to remove as many pairs of objects p and q as possible from each cell with label 2. Therefore, if a cell with label 2 contains a pair of subsets which have the same weight (that is, $m/2$, where m is the weight of the set V), all the objects p and q will be removed in this cell. Otherwise, at least one object p or q will remain in this cell.

The output stage starts at the step $3n + 10 + \lceil lgn \rceil + \lceil lgm \rceil$. At this step, the object $H_{\lceil lgn \rceil+\lceil lgm \rceil+4}$ is exchanged with one copy of objects E_1 and F_1 according to the rule r_{82}. From this step, there are two possible situations:

1) Affirmative answer: If there exists a pair of complementary subsets of V in a cell with label 2 such that they have the same weight $m/2$, then the corresponding cell does not contain any object p or q at step $3n + 10 + \lceil lgn \rceil + \lceil lgm \rceil$. Therefore, in the next step, the rules r_{83} and r_{84} are not applicable in that cell; however, the rule r_{85} can be applied, by which the object F_1 evolves to F_2. The object F_2 together with object E_1 introduces object T at step $3n + 12 + \lceil lgn \rceil + \lceil lgm \rceil$, and the object T will enter into the cell with label 1 at step $3n + 13 + \lceil lgn \rceil + \lceil lgm \rceil$. In the next step, object T together with object b (this object is in the cell with label 1 from the beginning of the computation) introduces object Y. Because the cell with label 1 also contains the object "yes" initially, object "yes" together with object Y will be moved to the environment by the rule r_{89}. This is done at the step $3n + 15 + \lceil lgn \rceil + \lceil lgm \rceil$ and the computation halts at this step. In this case, the computation gives an affirmative answer.

2) Negative answer: If there does not exist a pair of complementary subsets of V such that their weights are both equal to $m/2$, then all the 2^n cells with label 2 contain at least one object p or q, but not both of them. Thus, the object E_1 is removed from all the cells with label 2 together with one copy of p or q at step $3n + 11 + \lceil lgn \rceil + \lceil lgm \rceil$ by using the rule r_{83} or r_{84}. At the same time, the object F_1 evolves to F_2 by using the rule r_{85}. In this way, after the step $3n + 13 + \lceil lgn \rceil + \lceil lgm \rceil$ the object b remains in the cell with label 1. In the next two steps, only the rule $r_{54,i}$ can be applied; the subscript of object a_i increases by one in every step, thus it eventually evolves to $a_{3n+16+\lceil lgn \rceil+\lceil lgm \rceil}$. At step $3n + 16 + \lceil lgn \rceil + \lceil lgm \rceil$, the object $a_{3n+16+\lceil lgn \rceil+\lceil lgm \rceil}$ together with object b introduces object N into the cell with label 1 by the rule r_{90}, which is sent to the environment together with the object `no` at step $3n + 17 + \lceil lgn \rceil + \lceil lgm \rceil$ by the rule r_{91}. The computation ends with a negative answer.

5.2 Necessary resources

We will show that the family $\mathbf{\Pi} = \{\Pi(\langle n, m\rangle) \mid n, m \in \mathbb{N}\}$ defined above is polynomially uniform by Turing machines. To this aim, it will be proved that $\Pi(\langle n, m\rangle)$ is built in polynomial time with respect to the size parameters n and m of instances of PART.

It is easy to check that the rules of a system $\Pi(\langle n, m\rangle)$ of the family are defined recursively from the values n and m, and the necessary resources for building an element of the family are of a polynomial order, as shown below:

- Size of the alphabet: $3n^3 + 3n^2m + 4\lceil lgn\rceil + (n+2)\lceil lgm\rceil + 38n + 28 \in O(n^2 \cdot \max\{n, m\})$.
- Initial number of cells: $2 \in O(1)$.
- Initial number of objects: $2n^2 + 2nm + n + 8 \in O(n \cdot \max\{n, m\})$.
- Number of rules: $3n^3+3n^2m+n^2+mn+2n\lceil lgm\rceil+4\lceil lgn\rceil+2\lceil lgm\rceil+42n+28 \in O(n^2\cdot\max\{n, m\})$.
- Maximal length of a rule: $8 \in O(1)$.

Therefore, a deterministic Turing machine can build $\Pi(\langle n, m\rangle)$ in polynomial time with respect to n and m.

5.3 Main results

From the discussion in the previous sections and according to the definition of solvability given in section 4, the following result holds.

Theorem 5.1. PART$\in$ PMC$_{TS}$.

Corollary 5.2. $\text{NP} \cup \text{co} - \text{NP} \subseteq \text{PMC}_{TS}$.

Proof. It suffices to make the following observations: the PART problem is NP-complete, PART$\in$ PMC$_{TS}$ and this complexity class is closed under polynomial-time reduction and under complement.

6 Discussion

P systems are introduced as parallel computing models, which are inspired by biological systems. The efficiency of P systems for solving computationally hard problems has been widely investigated. As we know, all NP problems can be reduced to an NP-complete problem in a polynomial time. In principle, if a family of P systems can efficiently solve an NP-complete problem, then this family of P systems can also efficiently solve all NP problems. But, until now, it remains open how a family of P systems can be designed to efficiently compute the reduction from an NP problem to an NP-complete problem. In this work, a uniform family of P systems with cell separation is directly constructed for efficiently solving the NP-complete problem PART.

Before the above open problem for computing the reduction process by P systems is solved, it is still interesting to give efficient solutions to computationally hard problems in the framework of tissue P systems with cell separation, such as the vertex cover problem, the clique problem, the Hamiltonian path problem.

In the definition of tissue P systems with cell separation given in this work, it is supposed that there are arbitrary copies of objects in the environment. From the practical point of view, it is reasonable to have an upper bound on the number of objects in the environment. It remains open whether tissue P systems with finite number of objects in the environment can efficiently solve NP-complete problems.

Membrane creation is another way to generate exponential workspace in a polynomial time and it is already shown that cell-like P systems with membrane creation can efficiently solve computationally hard problems [25]. It is interesting to introduce membrane creation into tissue P systems and investigate the computational power of tissue P systems with cell creation rules.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant Nos. 61033003, 30870826, 60703047, 60772122 and 61003038), Ph.D Programs Foundation of Ministry of Education of China (Grant No. 20100142110072), Fundamental Research Funds for the Central Universities (Grant No. 2010ZD001), the Natural Science Foundation of Hubei Province (Grant Nos. 2008CDB113, 2008CDB180), and the Scientific Research Foundation for Doctor of Anhui University (Grant No. 02203104).

References

1 Păun Gh, Rozenberg G, Salomaa A. Handbook of Membrane Computing. Oxford: Oxford University Press, 2010
2 The P System Web Page: http://ppage.psystems.eu
3 Păun Gh. Computing with membranes. J Comput Syst Sci, 2000, 61: 108–143
4 Martín-Vide C, Pazos J, Păun Gh, et al. Tissue P systems. Theor Comput Sci, 2003, 296: 295–326
5 Păun A, Păun Gh. The power of communication: P systems with symport/antiport. New Gen Comput, 2002, 20: 295–305
6 Alhazov A, Freund R, Oswald M. Tissue P systems with antiport rules and small numbers of symbols and cells. Lect Notes Comput Sci, 2005, 3572: 100–111
7 Bernardini F, Gheorghe M. Cell communication in tissue P systems and cell division in population P systems. Soft Comput, 2005, 9: 640–649
8 Freund R, Păun Gh, Pérez-Jiménez M J. Tissue P systems with channel states. Theor Comput Sci, 2005, 330: 101–116
9 Krishna S N, Lakshmanan K, Rama R. Tissue P systems with contextual and rewriting rules. Lect Notes Comput Sci, 2003, 2597: 339–351
10 Lakshmanan K, Rama R. On the power of tissue P systems with insertion and deletion rules. In: Preproceedings of the Fourth International Workshop on Membrane Computing (report RGML 28/03). Tarragona, Spain, 2003. 304–318
11 Prakash V J. On the power of tissue P systems working in the maximal-one mode. In: Preproceedings of the Fourth International Workshop on Membrane Computing (report RGML 28/03). Tarragona, Spain, 2003. 356–364
12 Păun Gh, Pérez-Jiménez M J, Riscos-Núñez A. Tissue P systems with cell division. Int J Comput Commun Control, 2008, III: 295–302
13 Díaz-Pernil D. Sistemas P de tejido: formalización y efficiencia computacional, Ph.D. Thesis, University of Seville, Seville, Spain, 2008
14 Díaz-Pernil D, Gutiérrez-Naranjo M A, Pérez-Jiménez M A, et al. A linear-time tissue P system based solution for the 3-coloring problem. Electr Notes Theor Comput Sci, 2007, 171: 81–93
15 Díaz-Pernil D, Gutiérrez-Naranjo M A, Pérez-Jiménez M A, et al. A uniform family of tissue P system with cell division solving 3-COL in a linear time. Theor Comput Sci, 2008, 404: 76–87
16 Díaz-Pernil D, Gutiérrez-Naranjo M A, Pérez-Jiménez M A, et al. Solving subset sum in linear time by using tissue P system with cell division. Lect Notes Comput Sci, 2007, 4527: 170–179
17 Díaz-Pernil D, Gutiérrez-Naranjo M A, Pérez-Jiménez M A, et al. Computational efficiency of cellular division in tissue-like membrane systems. Rom J Inf Sci Tech, 2008, 11: 229–241
18 Díaz-Pernil D, Gutiérrez-Naranjo M A, Pérez-Jiménez M A, et al. A fast solution to the partition problem by using tissue-like P systems. In: Proceedings of the Third International Conference on Bio-Inspired Computing: Theories and Applications, BIC-TA 2008. Adelaide, SA, Australia, 2008. 43–48
19 Pan L, Pérez-Jiménez M A. Computational complexity of tissue-like P systems. J Complex, 2010, 26: 296–315
20 Păun Gh. Membrane Computing—An Introduction. Berlin: Springer-Verlag, 2002
21 Martín Vide C, Pazos J, Păun Gh, et al. A new class of symbolic abstract neural nets: tissue P systems. Lect Notes Comput Sci, 2002, 2387: 573–679
22 Pan L, Ishdorj T-O. P systems with active membranes and separation rules. J Univer Comput Sci, 2004, 10: 630–649
23 Pérez-Jiménez M J, Romero-Jiménez A, Sancho-Caparrini F. A polynomial complexity class in P systems using membrane division. In: Proceedings of the Fifth Workshop on Descriptional Complexity of Formal Systems, DCFS 2003. Budapest, Maryland, Hungary, 2003. 284–294
24 Garey M R, Johnson D S. Computers and Intractability. A Guide to the Theory of NP-Completeness. New York: W.H. Freeman and Company, 1979
25 Gutiérrez-Naranjo M A, Pérez-Jiménez M A, Romero-Campero F J. A uniform solution to SAT using membrane creation. Theor Comput Sci, 2007, 371: 54–61