International Journal of Networking and Computing – www.ijnc.org
ISSN 2185-2839 (print) ISSN 2185-2847 (online)
Volume 4, Number 1, pages 2–22, January 2014

Asynchronous P systems for hard graph problems

Kohei Tanaka and Akihiro Fujiwara
Graduate School of Computer Science and Systems Engineering,
Kyushu Institute of Technology,
Iizuka, Fukuoka 820-8502, Japan

Received: July, 23, 2013
Revised: October 18, 2013
Accepted: November 29, 2013
Communicated by Susumu Matsumae

Abstract

In the present paper, we consider fully asynchronous parallelism in membrane computing and propose asynchronous P systems for the following four graph problems: minimum coloring, maximum independent set, minimum vertex cover, and maximum clique. We first propose an asynchronous P system that solves the minimum graph coloring for a graph with n nodes and show that the proposed P system works in $O(n^{n+2})$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ kinds of objects. Second, we propose an asynchronous P system that solves the maximum independent set for a graph with n nodes and show that the proposed P system works in $O(n^2 \cdot 2^n)$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ kinds of objects. We next propose two asynchronous P systems that solve the minimum vertex cover and the maximum clique for the same input graph by reduction to the maximum independent set and show that the proposed P system works in $O(n^2 \cdot 2^n)$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ kinds of objects.

Keywords: membrane computing, asynchronous P system, graph coloring, independent set, vertex cover, clique

1 Introduction

Membrane computing, which is an example of natural computing, is a computational model inspired by the structures and behaviors of living cells. In the initial study on membrane computing, a basic feature of membrane computing was introduced by Păun [5] as a P system. In the P system, activities in living cells are considered as parallel computing. Each P system consists of hierarchically embedded cell membranes, and each membrane may contain objects. Each object evolves into another object according to applicable evolution rules.

The P system and most variants have proved to be universal [7]. In addition, the models deliver superior performance on problems that need exponential computation time, such as NP-complete or NP-hard problems, and various P systems [2, 4, 6, 9, 10] have been proposed for solving NP problems. However, synchronous application of evolution rules is assumed on the above P systems with maximal parallelism, which is a main feature of P systems. Maximal parallelism means that all applicable rules in all membranes are applied synchronously.

On the other hand, cell biochemistry has obvious asynchronous parallelism. Asynchronous parallelism means that all objects may react on rules with different speeds, and evolution rules are applied to objects independently. Since all objects in a living cell basically work in an asynchronous manner, asynchronous parallelism must be considered to make the P system a more realistic model.

For considering asynchronous parallelism, a number of P systems have been proposed. As an example, some sequential P systems [1], which assume the sequential application of appropriate evolution rules, have been proposed. As another example, two asynchronous P systems [3] have been proposed for solving the SAT and Hamiltonian cycle problem. The P systems solve NP problems in a polynomial number of parallel steps. In addition, another asynchronous P system [8] has been proposed for computing arithmetic operations and factorization.

As a complexity of the asynchronous P system, two kinds of numbers are considered: the number of sequential steps and the number of parallel steps. The number of sequential steps is the number of executed steps in the case that rules are applied sequentially, and the number of parallel steps is the number of executed steps with maximal parallelism.

In the present paper, we propose asynchronous P systems for four graph problems: graph coloring, maximum independent set, minimum vertex cover, and maximum clique. The four problems are well-known NP hard graph problems, and the minimum vertex cover and the maximum clique are reducible to the maximum independent set in a polynomial number of steps.

We first propose an asynchronous P system that solves the minimum graph coloring for a graph with n nodes.

The proposed P system solves the problem in $O(n^{n+2})$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ kinds of objects.

We next propose an asynchronous P system for the maximum independent set for a graph with n nodes. The proposed P system solves the problem in $O(n^2 \cdot 2^n)$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ kinds of objects.

We finally propose asynchronous P systems that solve the minimum vertex cover and the maximum clique for the same input graph by reduction to the maximum independent set. The proposed P systems solve the problems in $O(n^2 \cdot 2^n)$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ kinds of objects.

The remainder of the present paper is organized as follows. In Section 2, we give a preliminary description of the model for asynchronous membrane computing. In Section 3, we propose the asynchronous P system for minimum graph coloring. We propose the asynchronous P system for the maximum independent set in Section 4. We also propose two asynchronous P systems for the other two problems in Section 5. Section 6 concludes the paper.

2 Preliminaries

2.1 Computational model for membrane computing

The P system [2] mainly consists of membranes and objects. A membrane is a computing cell, in which independent computations are executed in parallel. Each membrane may contain objects and other membranes. In other words, the membranes form nested structures. In the present paper, each membrane is denoted by a pair of square brackets, and the number on the right-hand side of each right bracket denotes the label of the corresponding membrane. An object in the P system is a memory cell that stores each data and can divide, dissolve, and pass through membranes. In the present paper, each object is denoted by finite strings over a given alphabet and is contained in one of the membranes.

We formally define a P system Π and the sets used in the system as follows.

$$\Pi = (O, \mu, \omega_1, \omega_2, \cdots, \omega_m, R_1, R_2, \cdots, R_m, i_{in}, i_{out})$$

O: O is the set of all objects used in the system.

μ: μ is membrane structure that consists of m membranes. Each membrane in the structure is labeled with an integer. In addition, the outermost membrane is called the skin membrane, and the skin membrane contains all of the other membranes.

ω_j: ω_j is a set of objects initially contained in the membrane labeled j.

R_j: R_j is a set of evolution rules that are applicable to objects in the membrane labeled j.

i_{in}: i_{in} is a label of the input membrane.

i_{out}: i_{out} is a label of the output membrane.

In the present paper, we assume that input objects are given from the outside region into the skin membrane, and computation starts by applying evolution rules. We also assume that output objects are sent out from the skin membrane to the outside region.

In membrane computing, several types of rules are proposed. We consider five basic rules of the following forms in [2].

(1) Object evolution rule: $[\alpha]_h \rightarrow [\beta]_h$
where h is the label of the membrane, and $\alpha, \beta \in O$. With this rule, object α evolves into another object β. (We omit the brackets in each evolution rule for cases that the corresponding membrane is obvious.)

(2) Send-in communication rule: $\alpha[\ ]_h \rightarrow [\beta]_h$
where h is the label of the membrane, and $\alpha, \beta \in O$. With this rule, object α is sent into the membrane and can evolve into another object β.

(3) Send-out communication rule: $[\alpha]_h \rightarrow [\ ]_h\beta$
where h is the label of the membrane, and $\alpha, \beta \in O$. With this rule, object α is sent out of the membrane and can evolve into another object β.

(4) Dissolution rule: $[\alpha]_h \rightarrow \beta$
where h is the label of the membrane, and $\alpha, \beta \in O$. With this rule, the membrane, which contains object α, is dissolved, and the object can evolve into another object β. (Note that the skin membrane cannot be dissolved.)

(5) Division rule: $[\alpha]_h \rightarrow [\beta]_h[\gamma]_h$
where h is the label of the membrane, and $\alpha, \beta, \gamma \in O$. With this rule, the membrane, which contains object α, is divided into two membranes that contain objects β and γ.

We assume that each of the above rules is applied in a constant number of biological steps. In the following sections, we consider the number of steps executed in a P system as the complexity of the P system.

In addition, the membrane computing has two features, which are maximal parallelism and non-determinism. Maximal parallelism means that all applicable rules are applied in parallel. (Membrane computing is considered as a kind of parallel computing from this feature.) On the other hand, non-determinism means that applicable rules are non-deterministically chosen in the case that there are several possibilities of the applicable rules.

For an example of the above two features, we show a simple P system. We define the P system Π and the sets used in the system as follows.

$$\Pi = (O, \mu, \omega_1, R_1, i_{in}, i_{out})$$

- $O = \{a, b, c, d, e, f, g\}$
- $\mu = [\ ]_1$
- $\omega_1 = \phi$

- $R_1 = \{a \rightarrow b, bc \rightarrow d, c \rightarrow e, be \rightarrow f, bbe \rightarrow g\}$
- $i_{in} = 1$
- $i_{out} = 1$

We now show an example of the computation of the P system Π. Let us assume that an input object *aac* is given to the skin membrane from the outside region into the skin membrane. The initial state of the P system is given as follows.

$$[\ aac\ ]_1$$

In the initial state, the applicable rules are $a \rightarrow b$ and $c \rightarrow e$, and the two rules are applied in parallel with maximal parallelism. Then, the state of the P system is changed as follows after the first computation step in the P system.

$$[\ bbe\ ]_1$$

In the second state, the applicable rule is $be \rightarrow f$ or $bbe \rightarrow g$, and one of the rules is applied with non-determinism. Then, the state of the P system is changed to one of the following states after the second computation step in the P system.

$$[\ bf\ ]_1 \text{ or } [\ g\ ]_1$$

2.2 Asynchronous P systems

In this subsection, we describe the differences between an asynchronous P system, which is considered in the paper, and a conventional P system. In a conventional P system, evolution rules are applied in the maximal parallel manner described in the above subsection. In an asynchronous P system, evolution rules are applied in an asynchronous parallel manner, i.e., at least one of the applicable evolution rules is applied in each step of the computation. The reason why we assume asynchronous parallelism in this paper is based on the fact that every living cell acts independently and asynchronously. Since the conventional P system ignores the asynchronous feature of living cells, the asynchronous P system is a more realistic computation model for cell activities.

We now show an example of the differences between the conventional P system with maximal parallelism and the asynchronous P system. We assume the asynchronous P system Π_a, whose sets used in the system are same as Π.

We consider the computations in the conventional P system and the asynchronous P system, respectively. In the conventional P system, all applicable evolution rules are applied in parallel, and objects are evolved as follows.

$$aac \rightarrow bbe \rightarrow bf$$
$$aac \rightarrow bbe \rightarrow g$$

On the other hand, the asynchronous P system uses four kinds of computations, which are given below according to the order of application of the evolution rules.

$$aac \rightarrow bbe \rightarrow bf$$
$$aac \rightarrow bbe \rightarrow g$$
$$aac \rightarrow abc \rightarrow ad$$
$$aac \rightarrow abc \rightarrow bd$$

Therefore, a number of executions are possible in the asynchronous P system, and the evolution rules in the conventional P system, which assumes a maximal parallel manner, may not work in an asynchronous parallel manner.

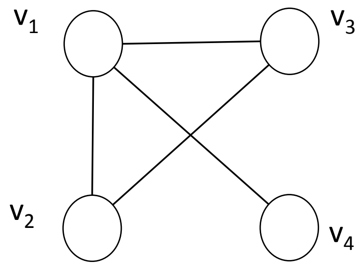

Figure 1: An example of an input graph

In the asynchronous P system, all evolution rules can be applied completely in parallel, which is the same as the conventional P system, or all evolution rules can be applied sequentially. We define the number of steps executed in the asynchronous P system in the maximal parallel manner as the number of parallel steps. We also define the number of steps in the case that the applicable evolution rules are applied sequentially as the number of sequential steps. The numbers of parallel and sequential steps indicate the best and worst case complexities, respectively, for the asynchronous P system. In addition, the proposed asynchronous P system must be guaranteed to output a correct solution in any asynchronous execution.

3 Minimum graph coloring

3.1 Input and output

In this section, we consider a P system for the minimum graph coloring problem. Given a graph $G = (V, E)$ such that $V = \{v_1, v_2, \cdots, v_n\}$, coloring is the assignment of colors to all vertices such that no pair of the same color vertices are bridged by an arbitrary edge $e \in E$. For example, given the graph in Figure 1, a color assignment, such that $c_1 = \{v_1\}$, $c_2 = \{v_2\}$ and $c_3 = \{v_3, v_4\}$, is a coloring of the graph. The minimum graph coloring problem needs to find a color assignment for the input graph with the minimum number of colors.

The input of the minimum graph coloring is the set of objects O_E, given below.

$$O_E \quad = \quad \{\langle e_{i,j}, W\rangle \mid 1 \le i \le n, 1 \le j \le n, W \in \{T, F\}\}$$

Each object $\langle e_{i,j}, W\rangle$ denotes an edge (v_i, v_j). If an edge (v_i, v_j) exists in the input graph, W is set to T; otherwise, W is set to F.

For example, the following objects denote the input of the minimum graph coloring for the graph in Figure 1.

$$\begin{array}{cccc}
\langle e_{1,1}, F\rangle & \langle e_{1,2}, T\rangle & \langle e_{1,3}, T\rangle & \langle e_{1,4}, T\rangle \\
\langle e_{2,1}, T\rangle & \langle e_{2,2}, F\rangle & \langle e_{2,3}, T\rangle & \langle e_{2,4}, F\rangle \\
\langle e_{3,1}, T\rangle & \langle e_{3,2}, T\rangle & \langle e_{3,3}, F\rangle & \langle e_{3,4}, F\rangle \\
\langle e_{4,1}, T\rangle & \langle e_{4,2}, F\rangle & \langle e_{4,3}, F\rangle & \langle e_{4,4}, F\rangle
\end{array}$$

The output of the P system is denoted by the set of n objects given below.

$$O_C = \{\langle V_i, h\rangle | 1 \le i \le n, 1 \le h \le m (m \le n)\}$$

Each object $\langle V_i, h\rangle$ means that vertex v_i is labeled with color h, and O_C denotes the minimum color assignment for the input graph. For example, the following set of objects is the output of the minimum graph coloring for the graph in Figure 1.

$$O_C = \{\langle V_1, 1\rangle, \langle V_2, 2\rangle, \langle V_3, 3\rangle, \langle V_4, 3\rangle\}$$

We assume that the set of input objects is given from the outside region into the skin membrane, and the output object is sent out from the skin membrane to the outside region.

3.2 Overview of the asynchronous P system

We now describe an overview of the asynchronous P system for the minimum graph coloring. The P system for the minimum graph coloring problem consists of $n + 1$ membranes such that $[\ [\ ]_1\ [\ ]_2 \cdots [\ ]_n\ ]_0$. For each inner membrane, labeled h, the input graph is checked to determine whether the graph can be colored with h colors using the division of the membrane. (In the following, a graph is called h-colorable if the graph can be colored with h colors.)

The computation in the asynchronous P system generally consists of the following five steps.

Step 1 Move modified copies of input objects into all inner membranes.

Step 2 In each inner membrane labeled h, create all possible color assignments with h colors by dividing the inner membrane.

Step 3 Check whether each color assignment is h-coloring, and send out the result of the check to the outer membrane.

Step 4 In the outer membrane, check the results of the inner membranes.

Step 5 Dissolve one of the inner membranes, which includes the minimum graph coloring, and send out the result from the membrane.

3.3 Details of the asynchronous P system

We show each step of the asynchronous P system for the minimum graph coloring in the following. In Step 1, modified copies of input objects are moved to all inner membranes. Step 1 is executed by applying the following set of evolution rules. (In the following description, a set of evolution rules $R_{i,j}$ represents that the rules are used for membrane i in Step j.)
(Evolution rules for the outer membrane)

$$\begin{aligned}
R_{0,1} \quad = \quad & \{\langle e_{i,j}, W\rangle \rightarrow \langle e_{i,j}, W, 1\rangle\langle f_{i,j}, W, 1\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, W \in \{T, F\}\} \\
& \cup\{\langle f_{i,j}, W, h\rangle \rightarrow \langle e_{i,j}, W, h+1\rangle\langle f_{i,j}, W, h+1\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, \\
& \quad 1 \leq h \leq n-2, W \in \{T, F\}\} \\
& \cup\{\langle f_{i,j}, W, n-1\rangle \rightarrow \langle e_{i,j}, W, n\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, W \in \{T, F\}\} \\
& \cup\{\langle e_{1,1}, F, h\rangle[\ ]_h \rightarrow [\langle M_{2,1}, h\rangle\langle e_{1,1}, F, h\rangle]_h \mid 1 \leq h \leq n\} \\
& \cup\{\langle M_{i,j}, h\rangle\langle e_{i,j}, W, h\rangle[\ ]_h \rightarrow [\langle M_{i+1,j}, h\rangle\langle e_{i,j}, W, h\rangle]_h \mid 1 \leq i \leq n, 1 \leq j \leq n, W \in \{T, F\}\} \\
& \cup\{\langle M_{n+1,j}, h\rangle \rightarrow \langle M_{1,j+1}, h\rangle \mid 1 \leq h \leq n, 1 \leq j \leq n\}
\end{aligned}$$

(Evolution rules for the inner membrane)

$$\begin{aligned}
R_{h,1} \quad = \quad & \{[\langle M_{i,j}, h\rangle]_h \rightarrow [\ ]_h\langle M_{i,j}, h\rangle \mid 1 \leq h \leq n, 1 \leq i \leq n+1, 1 \leq j \leq n+1\} \\
& \cup\{\langle M_{1,n+1}, h\rangle \rightarrow \langle S_1, h\rangle\langle h\rangle | 1 \leq h \leq n\}
\end{aligned}$$

In the above evolution rules, two kinds of objects $\langle e_{i,j}, W, h\rangle$ and $\langle f_{i,j}, W, h\rangle$ are created from input object $\langle e_{i,j}, W\rangle$, and the created objects of the form $\langle e_{i,j}, W, h\rangle$ are moved into a membrane labeled h by object $\langle M_{i,j}, h\rangle$. At the end of Step 1, object $\langle S_1, h\rangle$ and object $\langle h\rangle$ are created in the membrane labeled h, and the object triggers the computation of Step 2.

In Step 2, all possible color assignments are created with h colors. Step 2 is executed by applying the following set of evolution rules.
(Evolution rules for the inner membrane)

$$\begin{aligned}
R_{h,2} \quad = \quad & \{\langle S_i, 1\rangle \rightarrow \langle D_i, 1\rangle | 1 \leq i \leq n\} \\
& \cup\{[\langle S_i, k\rangle]_h \rightarrow [\langle S_{i+1}\rangle\langle V_i, k\rangle]_h[\langle D_i, k-1\rangle]_k \mid 2 \leq k \leq n, 1 \leq i \leq n, 2 \leq h \leq n\}
\end{aligned}$$

$$\cup\{[\langle D_i,k\rangle]_h \rightarrow [\langle S_{i+1}\rangle\langle V_i,k\rangle]_h[\langle D_i,k-1\rangle]_k \mid 2\le k\le n, 1\le i\le n, 2\le h\le n\}$$
$$\cup\{\langle D_i,1\rangle \rightarrow \langle S_{i+1}\rangle\langle V_i,1\rangle \mid 1\le i\le n\}$$
$$\cup\{\langle S_i\rangle\langle k\rangle \rightarrow \langle S_i,k\rangle\langle k\rangle \mid 1\le i\le n, 1\le k\le n\}$$

In Step 2, subsets of vertices are created by dividing membrane with object $\langle S_i, k\rangle$ repeatedly. First, the color assignment for each vertex is executed by labeling an integer for the vertex. For example, object $\langle V_1, k\rangle$ denotes a vertex v_1 colored with k. Then, the subsets of vertices are created by dividing membrane with object $\langle S_i, k\rangle$ and object $\langle D_i, k-1\rangle$. Object $\langle S_i, k\rangle$, which is created by object $\langle S_i\rangle$ and object $\langle k\rangle$, starts labeling the next vertex v_{i+1}. Object $\langle D_i, k\rangle$ labels vertices as well as object $\langle S_i, k\rangle$ and vertex can be labeled different colors due to reducing color number k of object $\langle D_i, k\rangle$. At the end of Step 2, object $\langle S_{n+1}\rangle$ is created, and the object triggers the computation of Step 3.

In Step 3, the assignment is checked to determine whether each color assignment is h-coloring, and a result of the check is sent out to the outer membrane. Step 3 is executed by applying the following set of evolution rules.
(Evolution rules for the inner membrane)

$$\begin{aligned}
R_{h,3} \quad = \quad & \{\langle S_{n+1}\rangle \rightarrow \langle I_{1,1}\rangle\} \\
& \cup\{\langle I_{i,j}\rangle\langle e_{i,j},F,h\rangle \rightarrow \langle I_{i,j+1}\rangle\langle e_{i,j},F,h\rangle \mid 1\le i\le n, 1\le j\le n, 1\le h\le n\} \\
& \cup\{\langle I_{i,j}\rangle\langle e_{i,j},T,h\rangle\langle V_i,v\rangle\langle V_j,v'\rangle \rightarrow \langle I_{i+1,j}\rangle\langle V_i,v\rangle\langle V_j,v'\rangle\langle e_{i,j},T,h\rangle \\
& \qquad \mid 1\le i\le n, 1\le j\le n, 1\le v\le n, 1\le h\le n, 1\le v'\le n, v\ne v', i\ne j\} \\
& \cup\{\langle I_{i,j}\rangle\langle e_{i,j},T,h\rangle\langle V_i,v\rangle\langle V_j,v\rangle\langle h\rangle \rightarrow \langle T_{1,1}\rangle\langle e_{i,j},T,h\rangle\langle V_i,v\rangle\langle V_j,v\rangle \\
& \qquad \mid 2\le h\le n, 1\le i\le n, 1\le j\le n, 1\le v\le n, i\ne j\} \\
& \cup\{\langle I_{i,n+1}\rangle \rightarrow \langle I_{i+1,j}\rangle | 1\le i\le n\} \\
& \cup\{[\langle I_{n+1,1}\rangle\langle h\rangle]_h \rightarrow \langle TRUE,h\rangle \mid 1\le h\le n\} \\
& \cup\{\langle T_{i,j}\rangle\langle e_{i,j},W,h\rangle \rightarrow \langle T_{i+1,j}\rangle \mid 1\le i\le n, 1\le j\le n, 1\le h\le n, W\in\{T,F\}\} \\
& \cup\{\langle T_{i,n+1}\rangle\langle V_i,h\rangle \rightarrow \langle T_{i+1,n+1}\rangle \mid 1\le i\le n, 1\le h\le n\} \\
& \cup\{\langle T_{n+1,j}\rangle \rightarrow \langle T_{1,j+1}\rangle \mid 1\le j\le n+1\} \\
& \cup\{[\langle T_{1,n+2}\rangle]_h \rightarrow \langle FALSE,1,h\rangle \mid 1\le h\le n\}
\end{aligned}$$

In the above evolution rules, each color assignment is checked to determine whether the assignment is h-coloring. Since the condition for h-coloring is that no two vertices with the same color are connected, the check is executed in the following three cases.

- **In the case that object $\langle e_{i,j}, F\rangle$ exists in the membrane:** No edge exists between v_i and v_j. Then, the check is omitted and next check is executed by $\langle I_{i,j+1}\rangle$.
- **In the case that objects $\langle e_{i,j}, T\rangle$ exist in the membrane, and $h_1 \ne h_2$ for objects $\langle V_i, h_1\rangle$ and $\langle V_j, h_2\rangle$:**

  An edge exists between v_i and v_j, and the two vertices are colored with different colors. Then, the check is passed and the next check is executed by $\langle I_{i+1,j}\rangle$.
- **In the other case:**

  An edge exists between v_i and v_j, and the two vertices are colored with the same color. Then, the color assignment for the membrane is not h-coloring. Therefore, object $\langle T_{1,1}\rangle$ is created, and object $\langle T_{1,1}\rangle$ starts the deletion of all objects in the membrane. Finally, object $\langle T_{1,n+2}\rangle$ dissolves the inner membrane, and object $\langle FALSE, 1, h\rangle$ which denotes the failure of h-coloring is created.

If object $\langle I_{i,j}\rangle$ checks all pairs of vertices, and object $\langle FALSE, 1, h\rangle$ is not created, object $\langle TRUE, h\rangle$, which denotes the success of h-coloring in the membrane, is sent out from the inner membrane.

In Step 4, the results of the inner membranes are checked in the outer membrane. Step 4 is executed by applying the following set of evolution rules. (In the following description, $\langle\alpha\rangle^h$ denotes h copies of object $\langle\alpha\rangle$.)
(Evolution rules for the outer membrane)

$$\begin{aligned} R_{0,4} &= \{\langle FALSE, h^i, h\rangle^h \rightarrow \langle FALSE, h^{i+1}, h\rangle \mid 0 \le i \le n-1, 1 \le h \le n\} \\ &\cup \{\langle FALSE, h^n, h\rangle \rightarrow \langle FALSE, h\rangle \mid 1 \le h \le n\} \\ &\cup \{\langle FALSE, 1\rangle \rightarrow \langle C_2\rangle \mid 0 \le i \le n-1\} \\ &\cup \{\langle FALSE, h\rangle\langle C_h\rangle \rightarrow \langle C_{h+1}\rangle \mid 2 \le h \le n\} \\ &\cup \{\langle TRUE, 1\rangle \rightarrow \langle B_1\rangle\} \\ &\cup \{\langle TRUE, h\rangle\langle C_h\rangle \rightarrow \langle B_h\rangle \mid 2 \le h \le n\} \end{aligned}$$

In the above evolution rules, object $\langle C_h\rangle$ is used to check whether the input graph is h-colorable. In the case that the input graph is not h-colorable, h^n objects, $\langle FALSE, 1, h\rangle$, are in the outer membrane. The number of objects is checked by creating objects $\langle FALSE, h^{i+1}, h\rangle$ from h copies of $\langle FALSE, h, h\rangle$. If object $\langle FALSE, h^n, h\rangle$ is created, the object is evolved into $\langle FALSE, h\rangle$, which is an output indicating that the input graph is not h-colorable. Since $\langle C_h\rangle$ is evolved into $\langle C_{h+1}\rangle$, if object $\langle FALSE, h\rangle$ exists in the membrane, the object continues the evolution until the input graph is h-colorable.

On the other hand, object $\langle TRUE, h\rangle$ is in the outer membrane in the case that the input graph is h-colorable. In this case, object $\langle B_h\rangle$ is created by two objects[1], $\langle TRUE, h\rangle$ and $\langle C_h\rangle$, and the object triggers the computation of Step 5.

In Step 5, one of the inner membranes, which includes the minimum graph coloring, is dissolved, and the result is sent out from the outer membrane. Step 5 is executed by applying the following set of evolution rules.
(Evolution rules for the outer membrane)

$$\begin{aligned} R_{0,5} &= \{\langle B_h\rangle[\ ]_h \rightarrow [\langle B_h\rangle]_h | 1 \le h \le n\} \\ &\cup \{\langle O_i\rangle \rightarrow \langle O_{i+1}\rangle\langle A_i\rangle | 1 \le i \le n\} \\ &\cup \{[\langle V_i, h\rangle\langle A_i\rangle]_0 \rightarrow [\ ]_0\langle V_i, h\rangle \mid 1 \le i \le n, 1 \le h \le n\} \end{aligned}$$

(Evolution rules for the inner membrane)

$$R_{h,5} = \{[\langle B_h\rangle\langle e_{i,1}, F, h\rangle]_h \rightarrow \langle O_1\rangle \mid 2 \le h \le n\}$$

At the beginning of Step 5, object $\langle B_h\rangle$ is moved into the inner membrane labeled h. Then, the membrane is dissolved by the object, and object $\langle O_1\rangle$ is created by other objects in the inner membrane. Next, a set of objects $\{\langle V_i, h\rangle | 1 \le i \le n\}$ is sent out from the outer membrane to the outside region by auxiliary objects $\langle O_{i+1}\rangle$ and $\langle A_i\rangle$.

We now summarize asynchronous P system Π_{mcg} for the minimum graph coloring as follows. (In the description of Π_{mcg}, $R_1, R_2 \cdots R_h$ are sets of evolution rules described in the above.)

$$\Pi_{mcg} = (O, \mu, \omega_1, \omega_2, \cdots, \omega_n, R_1, R_2, \cdots, R_n, i_{in}, i_{out})$$

$$\begin{aligned} O &= \{\langle e_{i,j}, W\rangle | 1 \le i \le n, 1 \le j \le n, W \in \{T, F\}\} \\ &\cup \{\langle V_{i,j}, h\rangle | 1 \le i \le n, 1 \le h \le n\} \\ &\cup \{\langle h\rangle \mid 0 \le h \le n\} \\ &\cup \{\langle M_{i,j}, h\rangle \mid 1 \le i \le n+1, 1 \le j \le n+1, 1 \le h \le n\} \\ &\cup \{\langle T_{i,j}\rangle, \langle I_{i,j}\rangle \mid 1 \le i \le n+1, 1 \le j \le n+2\} \\ &\cup \{\langle B_i\rangle, \langle O_i\rangle, \langle A_i\rangle \mid 1 \le i \le n+1\} \end{aligned}$$

[1] In the case that the input graph is 1-colorable, object $\langle C_h\rangle$ is not used.

$$\cup\{\langle FALSE, h^k, h\rangle, \langle FALSE, h\rangle, \langle TRUE, h\rangle, \mid 1 \leq h \leq n, 1 \leq k \leq n\}$$
$$\cup\{\langle D_i, h\rangle, \langle S_i, h\rangle \mid 1 \leq i \leq n, 1 \leq h \leq n\}$$
$$\cup\{\langle e_{i,j}, W, h\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, 1 \leq h \leq n, W \in \{T, F\}\}$$

$$\mu = [\ [\ ]_1\ [\ ]_2 \cdots [\ ]_n\ ]_0, \qquad \omega_1 = \omega_2 = \cdots = \omega_3 = \phi$$

3.4 Example execution of the P system

Figure 2 illustrates an example execution of the proposed P system Π_{mcg} for the graph in Figure 1. (The number at the upper left corner of each membrane represents a label of the membrane.)

At the beginning of the example, a set of objects, O_E, is given from the outside region into the outer membrane. Then, sets of evolution rules $R_{0,1} \cup R_{1,1} \cup R_{2,1} \cup R_{3,1} \cup R_{4,1}$ are applied, and four sets of objects $O_{E\_1}, O_{E\_2}, O_{E\_3}, O_{E\_4}$ are created. Each set of objects is moved into each corresponding inner membrane.

In the next step, sets of evolution rules $R_{1,2} \cup R_{2,2} \cup R_{3,2} \cup R_{4,2} \cup R_{1,3} \cup R_{2,3} \cup R_{3,3} \cup R_{4,3}$ are applied, and objects $\langle V_i, W\rangle (1 \leq i \leq n)$, which denote a color assignment with h colors, are created in each inner membrane labeled h. Then, each color assignment is checked to determine whether the assignment is h-coloring. In the case that the assignment is not h-coloring, all objects are removed in the membrane, and object $\langle FALSE, 1, h\rangle$ is sent out to the outer membrane. In this example, objects $\langle FALSE, 1, 1, \rangle$, $\langle FALSE, 1, 2\rangle$, $\langle FALSE, 1, 3\rangle$ and $\langle FALSE, 1, 4\rangle$ are sent out to the outer membrane.

In the other case, object $\langle TRUE, h\rangle$ is sent out to the outer membrane. In this example, $\langle TRUE, 3\rangle$ are$\langle TRUE, 4\rangle$ are sent out to the outer membrane.

Finally, the sets of evolution rules $R_{0,5} \cup R_{1,4} \cup R_{2,4} \cup R_{3,4}$ and $\cup R_{4,4}$ are applied, and one of the membranes containing the minimum graph coloring is dissolved, and the output objects are sent out from the outer membrane. In this example, $\langle V_1, 1\rangle$, $\langle V_2, 2\rangle$, $\langle V_3, 3\rangle$ and $\langle V_4, 2\rangle$ are sent out from the outer membrane.

3.5 Complexity

The complexity of asynchronous P system Π_{mcg} is as follows. Since $O(n^3)$ objects are moved into n membranes sequentially in Step 1, the numbers of sequential and parallel steps in Step 1 are $O(n^3)$ and $O(n^2)$, respectively. In Step 2, $O(n^n)$ membranes are created, and evolution rules are applied sequentially in each membrane. Therefore, the numbers of sequential and parallel steps of Step 2 are $O(n^n)$ and $O(n)$, respectively. In Step 3, each subset is checked sequentially, and both the numbers of sequential and parallel steps in Step 3 are $O(n^{n+2})$ and $O(n^2)$ by using evolution rules of size $O(n^5)$. In Step 4, the numbers of sequential and parallel steps are $O(n^n)$ and $O(n)$, respectively, since $O(n^n)$ objects are merged in the case that no assignment is n-coloring. In Step 5, both the numbers of sequential and parallel steps are $O(n)$ because n objects are outputted from the outer membrane sequentially.

Since the number of types of objects in P system Π_{mcg} is $O(n^2)$, and $O(n^5)$ kinds of evolution rules are used in Step 3, we obtain the following theorem for Π_{mcg}.

Theorem 1 *The asynchronous P system Π_{mcg}, which computes the minimum graph coloring with n vertices, works in $O(n^{n+2})$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ types of objects and evolution rules of size $O(n^5)$.* □

4 Maximum independent set

4.1 Input and output

Given a graph $G = (V, E)$ such that $V = \{v_1, v_2, \cdots v_n\}$, an independent set is defined as a subset $V' \subseteq V$ such that no two vertices in V' are bridged by an arbitrary edge $e \in E$. In addition, the maximum independent set for a graph is a problem that finds the largest independent set for

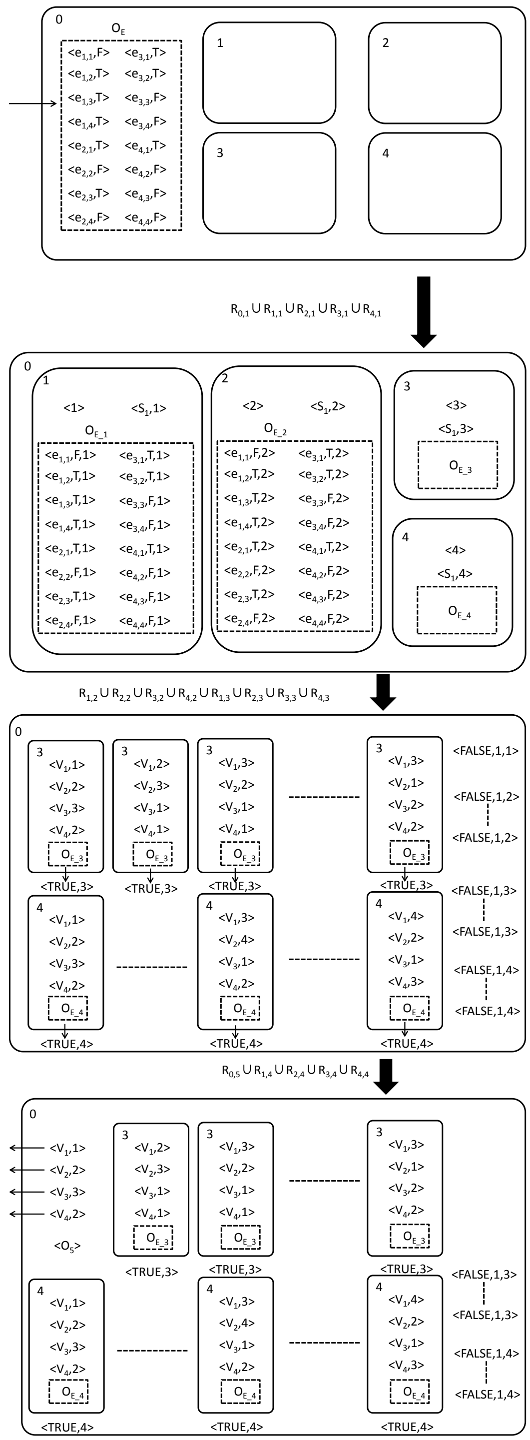

Figure 2: An example execution of Π_{mcg}

the graph. For example, given the graph in Figure 1, the set of vertices $\{v_3, v_4\}$ is the maximum independent set for the graph.

In the present paper, the input of the maximum independent set is the following set of objects

O_E, which is the same input set used for the minimum graph coloring in Section 3.

$$O_E = \{\langle e_{i,j}, W\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, W \in \{T, F\}\}$$

The output of the P system is denoted by the set of n objects given below.

$$O_S = \{\langle V_i, A\rangle \mid 1 \leq i \leq n, A \in \{0, 1\}\}$$

Each object $\langle V_i, A\rangle$ in O_S denotes the output for vertex v_i, and A is set to 1 if v_i is in the maximum independent set; otherwise, A is set to 0.

We also assume that a set of input objects is given from the outside region into the skin membrane, and the output object is sent out from the skin membrane to the outside region.

4.2 Overview of the asynchronous P system

We now describe an overview of the asynchronous P system for the maximum independent set. The P system consists of inner and outer membranes, i.e., the membrane structure of the P system is $[\,[\,]_2\,]_1$. The computation in the asynchronous P system generally consists of the following six steps.

Step 1 Move all input objects in the outer membrane into the inner membrane.

Step 2 Create all possible subsets of vertices by dividing the inner membrane repeatedly.

Step 3 Check each subset to determine if it is an independent set and sent out all independent subset from all inner membranes and compute the size of the maximum independent set in the outer membrane.

Step 4 By using the computed size of the maximum independent set, create all possible subsets of vertices again by dividing the inner membrane repeatedly.

Step 5 In each divided membrane, check whether each subset of vertices is the maximum independent set and dissolve all inner membranes if the membrane contains a subset that is not the maximum independent set.

Step 6 Dissolve one of the inner membranes that includes the maximum independent set and send out the result from the outer membrane.

4.3 Details of the asynchronous P system

We now show details of each step of the asynchronous P system for the maximum independent set.

In Step 1, all input objects in the outer membrane are moved into the inner membrane. Step 1 is executed by applying the following set of evolution rules.
(Evolution rules for the outer membrane)

$$\begin{aligned} R_{1,1} = \; & \{\langle e_{1,1}, F\rangle[\,]_2 \rightarrow [\langle M_{2,1}\rangle\langle e_{1,1}, F\rangle]_2\} \\ & \cup\{\langle M_{i,j}\rangle\langle e_{i,j}, W\rangle[\,]_2 \rightarrow [\langle M_{i+1,j}\rangle\langle e_{i,j}, W\rangle]_2 \mid 1 \leq i \leq n, 1 \leq j \leq n, W \in \{T, F\}\} \\ & \cup\{\langle M_{n+1,j}\rangle \rightarrow \langle M_{1,j+1}\rangle \mid 1 \leq j \leq n\} \\ & \cup\{\langle M_{1,n+1}\rangle[\,]_2 \rightarrow [\langle M_{1,n+2}\rangle]_2\} \end{aligned}$$

(Evolution rules for the inner membrane)

$$\begin{aligned} R_{2,1} = \; & \{[\langle M_{i,j}\rangle]_2 \rightarrow [\,]_2\langle M_{i,j}\rangle \mid 1 \leq i \leq n+1, 1 \leq j \leq n+1\} \\ & \cup\{[\langle M_{1,n+2}\rangle]_2 \rightarrow [\langle S_1\rangle]_2[\langle D\rangle]_2\} \end{aligned}$$

In the above evolution rules, object $\langle e_{1,1}, F\rangle$ starts the computation, and input objects, $\langle e_{i,j}, W\rangle$, are moved into the inner membrane by object $\langle M_{i,j}\rangle$. After all input objects, $\langle e_{i,j}, W\rangle$, are moved into the inner membrane, object $\langle M_{i,j}\rangle$ is changed into object $\langle M_{1,n+1}\rangle$. At the end of Step 1, object $\langle D\rangle$ and object $\langle S_1\rangle$ are created by division rules. The object $\langle D\rangle$ is used for the computation of Step 5, and object $\langle S_1\rangle$ triggers the computation of Step 2.

In Step 2, all possible subsets of vertices are created by dividing the inner membrane repeatedly. Step 2 is executed by applying the following set of evolution rules. In the evolution rules, $\langle V_i, 1\rangle$ denotes that v_i is contained in the subset of vertices.
(Evolution rules for inner membrane)

$$R_{2,2} = \{[S_i]_2 \rightarrow [\langle S_{i+1}\rangle\langle V_i, 0\rangle]_2[\langle S_{i+1}\rangle\langle V_i, 1\rangle]_2 \mid 1 \leq i \leq n\}$$

At the end of Step 2, 2^n membranes are created. In addition, object $\langle S_{n+1}\rangle$ is created in each divided membrane, and the object triggers the computation of Step 3.

In Step 3, in each divided membrane, each subset of vertices is checked to determine whether the subset is an independent set, and the number of vertices is sent out to the outer membrane in the case that the subset is an independent set. Then, the size of the maximum independent set is computed in the outer membrane.

Step 3 is executed by applying the following set of evolution rules.
(Evolution rules for the outer membrane)

$$\begin{aligned} R_{1,3} = \; & \{\langle l, 2^k\rangle\langle m, 2^k\rangle \rightarrow \langle m, 2^{k+1}\rangle \mid 0 \leq l \leq m \leq n, 0 \leq k \leq n-1\} \\ & \cup\{\langle k, 2^n\rangle \rightarrow \langle k\rangle\langle T\rangle \mid 0 \leq k \leq n\} \end{aligned}$$

(Evolution rules for the inner membrane)

$$\begin{aligned} R_{2,3} = \; & \{\langle S_{n+1}\rangle \rightarrow \langle I_{1,1}\rangle\} \\ & \cup\{\langle I_{i,j}\rangle\langle e_{i,j}, F\rangle \rightarrow \langle I_{i,j+1}\rangle\langle e_{i,j}, F\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n\} \\ & \cup\{\langle I_{i,j}\rangle\langle e_{i,j}, T\rangle\langle V_i, 0\rangle \rightarrow \langle I_{i+1,1}\rangle\langle e_{i,j}, T\rangle\langle V_i, 0\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n\} \\ & \cup\{\langle I_{i,j}\rangle\langle e_{i,j}, T\rangle\langle V_j, 0\rangle \rightarrow \langle I_{i,j+1}\rangle\langle e_{i,j}, T\rangle\langle V_j, 0\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n\} \\ & \cup\{\langle I_{i,j}\rangle\langle e_{i,j}, T\rangle\langle V_i, 1\rangle\langle V_j, 1\rangle \rightarrow \langle e_{i,j}, T\rangle\langle V_i, 1\rangle\langle V_j, 1\rangle\langle T_{1,1}\rangle\langle 0, 1\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n\} \\ & \cup\{\langle I_{i,n+1}\rangle \rightarrow \langle I_{i+1,1}\rangle \mid 1 \leq i \leq n\} \\ & \cup\{\langle I_{n+1,1}\rangle \rightarrow \langle 0\rangle\langle V_1\rangle\} \\ & \cup\{\langle V_i, 0\rangle\langle V_i\rangle \rightarrow \langle V_i, 0\rangle\langle V_{i+1}\rangle \mid 1 \leq i \leq n\} \\ & \cup\{\langle V_i, 1\rangle\langle V_i\rangle\langle k\rangle \rightarrow \langle V_i, 1\rangle\langle V_{i+1}\rangle\langle k+1\rangle \mid 1 \leq i \leq n, 0 \leq k \leq n\} \\ & \cup\{\langle V_{n+1}\rangle\langle k\rangle \rightarrow \langle T_{1,1}\rangle\langle k, 1\rangle \mid 1 \leq k \leq n\} \\ & \cup\{\langle T_{i,j}\rangle\langle e_{i,j}, V\rangle \rightarrow \langle T_{i+1,j}\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, V \in \{T, F\}\} \\ & \cup\{\langle T_{i,n+1}\rangle\langle V_i, A\rangle \rightarrow \langle T_{i+1,n+1}\rangle \mid 1 \leq i \leq n, A \in \{0, 1\}\} \\ & \cup\{\langle T_{n+1,j}\rangle \rightarrow \langle T_{1,j+1}\rangle \mid 1 \leq j \leq n+1\} \\ & \cup\{[\langle T_{1,n+2}\rangle\langle k, 1\rangle]_2 \rightarrow \langle k, 1\rangle \mid 0 \leq k \leq n\} \end{aligned}$$

In the above evolution rules for the inner membrane, each subset of vertices is checked to determine whether the subset is an independent set. Since the subset is an independent set if no two vertices are connected in the subset, the check is executed in the following four cases for each pair of vertices v_i and v_j.

- **In the case that object $\langle e_{i,j}, F\rangle$ is in the membrane:**

  No edge exists between v_i and v_j, and the next check is executed for a pair of vertices v_i and v_{j+1} by object $\langle I_{i,j+1}\rangle$.

- **In the case that objects $\langle e_{i,j}, T\rangle$ and $\langle V_i, 0\rangle$ are in the membrane:**

  An edge exists between v_i and v_j, but vertex v_i is not in the subset. The check is passed in this case, and the next check is executed for v_{i+1} and v_1 by object $\langle I_{i+1,1}\rangle$.

- **In the case that object $\langle e_{i,j}, T\rangle$ and object $\langle V_j, 0\rangle$ are in the membrane:**

  An edge exists between v_i and v_j, but vertex v_j is not in the subset. The check is passed, and the next check is executed for a pair of vertices v_i and v_{j+1} by $\langle I_{i,j+1}\rangle$.

- **In the other case:**

  An edge exists between v_i and v_j, and the two vertices are in the subset. Then, the subset of vertices is not an independent set. Therefore, object $\langle T_{1,1}\rangle$ and object $\langle 0, 1\rangle$ are created. (The object $\langle T_{1,1}\rangle$ denotes that the check for the membrane is finished, and the object $\langle 0, 1\rangle$ denotes the size of the subset is 0 because the subset is not an independent set.) Then, object $\langle T_{1,1}\rangle$ starts the deletion of all objects in the membrane except for object $\langle 0, 1\rangle$. Finally, object $\langle T_{1,n+2}\rangle$ dissolves the inner membrane.

If all pairs of vertices are checked, unless object $\langle T_{i,j}\rangle$ is created, objects $\langle V_i\rangle$ and $\langle 0\rangle$ are created. Then, the number of vertices in the subset is counted by object $\langle k\rangle$ in the membrane. When counting the number of vertices is finished, object $\langle T_{1,1}\rangle$ and object $\langle k, 1\rangle$ are created. The $\langle T_{1,1}\rangle$ denotes that the check for the membrane is finished, and $\langle k, 1\rangle$ denotes the size of the subset in the membrane. Then, object $\langle T_{1,1}\rangle$ starts the deletion of all objects in the membrane except for object $\langle k, 1\rangle$, and object $\langle T_{n+2,1}\rangle$ dissolves the inner membrane.

In the above evolution rules for the outer membrane, the maximum size of independent sets is decided by 2^n objects $\langle k, 1\rangle$ to denote the size of the independent set. In the evolution rules, pairs of objects, $\langle l, 2^k\rangle$ and $\langle m, 2^k\rangle$, are compared, and object $\langle m, 2^{k+1}\rangle$, which denotes $m \geq l$ and the number of objects is 2^{k+1}, is created. At the end of Step 3, objects $\langle k, 1\rangle$ and $\langle T\rangle$ are created. The object $\langle k, 1\rangle$ denotes that the maximum size of independent sets is k, and object $\langle T\rangle$ triggers the computation of Step 4.

In Step 4, all possible subsets of vertices subsets are created again by dividing the inner membrane repeatedly. Step 4 is executed by applying the following set of evolution rules.
(Evolution rules for the outer membrane)

$$R_{1,4} = \{\langle T\rangle\langle k\rangle[\ ]_2 \rightarrow [\langle T\rangle\langle k\rangle]_2 \mid 0 \leq k \leq n\}$$

(Evolution rules for the inner membrane)

$$\begin{aligned} R_{2,4} &= \{\langle T\rangle\langle D\rangle \rightarrow \langle S'_1\rangle\} \\ &\cup\{[S'_i]_2 \rightarrow [\langle S'_{i+1}\rangle\langle V_i, 0\rangle]_2[\langle S'_{i+1}\rangle\langle V_i, 1\rangle]_2 \mid 1 \leq i \leq n\} \end{aligned}$$

In the above evolution rules for the outer membrane, objects $\langle T\rangle$ and $\langle k\rangle$, which are created in Step 3, are moved into the inner membrane. On the other hand, in the above evolution rules for the inner membrane, object $\langle S'_{n+1}\rangle$ is created, and the object triggers the division rules. Then, 2^n membranes are created as in Step 2, and object $\langle S'_{n+1}\rangle$, which triggers the computation of Step 5, is created in each divided membrane.

In Step 5, each divided membrane is checked to determine whether a subset of vertices in the membrane is the maximum independent set, and the membrane is dissolved if the membrane contains a subset that is not the maximum independent set. Step 5 is executed by applying the following set of evolution rules.
(Evolution rules for the outer membrane)

$$\begin{aligned} R_{1,5} &= \{\langle D, 2^k\rangle\langle D, 2^k\rangle \rightarrow \langle D, 2^{k+1}\rangle \mid 0 \leq k \leq n-1\} \\ &\cup\{\langle D, 2^n\rangle \rightarrow \langle B\rangle\} \end{aligned}$$

(Evolution rules for the inner membrane)

$$\begin{aligned} R_{2,5} &= \{\langle S'_{n+1}\rangle \rightarrow \langle C_1\rangle\} \\ &\cup\{\langle C_i\rangle\langle V_i, 0\rangle \rightarrow \langle C_{i+1}\rangle\langle V_i, 0\rangle\} \end{aligned}$$

$$
\begin{aligned}
&\cup\{\langle C_i\rangle\langle V_i,1\rangle\langle k\rangle \rightarrow \langle C_{i+1}\rangle\langle V_i,1\rangle\langle k-1\rangle \mid 1\le k\le n\}\\
&\cup\{\langle C_{n+1}\rangle\langle 0\rangle \rightarrow \langle I'_{1,1}\rangle\}\\
&\cup\{\langle C_{n+1}\rangle\langle k\rangle \rightarrow \langle E_{1,1}\rangle \mid 1\le k\le n\}\\
&\cup\{\langle C_i\rangle\langle V_i,1\rangle\langle 0\rangle \rightarrow \langle V_i,1\rangle\langle E_{1,1}\rangle\}\\
&\cup\{\langle E_{i,j}\rangle\langle e_{i,j},V\rangle[\ ]_2 \rightarrow [\langle E_{i+1,j}\rangle\langle e_{i,j},V\rangle]_2 \mid 1\le i\le n, 1\le j\le n, V\in\{T,F\}\}\\
&\cup\{\langle E_{i,n+1}\rangle\langle V_i,A\rangle \rightarrow \langle E_{i+1,n+1}\rangle \mid 1\le i\le n, A\in\{0,1\}\}\\
&\cup\{\langle E_{n+1,j}\rangle \rightarrow \langle E_{1,j+1}\rangle \mid 1\le j\le n+1\}\\
&\cup\{[\langle E_{1,n+2}\rangle]_2 \rightarrow \langle D,1\rangle\}\\
&\cup\{\langle I'_{i,j}\rangle\langle e_{i,j},F\rangle \rightarrow \langle I'_{i,j+1}\rangle \mid 1\le i\le n, 1\le j\le n\}\\
&\cup\{\langle I'_{i,j}\rangle\langle e_{i,j},T\rangle\langle V_i,0\rangle \rightarrow \langle I'_{i+1,1}\rangle\langle V_i,0\rangle \mid 1\le i\le n, 1\le j\le n\}\\
&\cup\{\langle I'_{i,j}\rangle\langle e_{i,j},T\rangle\langle V_j,0\rangle \rightarrow \langle I'_{i,j}+1\rangle\langle V_j,0\rangle \mid 1\le i\le n, 1\le j\le n\}\\
&\cup\{\langle I'_{i,j}\rangle\langle e_{i,j},T\rangle\langle V_i,1\rangle\langle V_j,1\rangle \rightarrow \langle V_i,1\rangle\langle V_j,1\rangle\langle E_{1,1}\rangle \mid 1\le i\le n, 1\le j\le n\}\\
&\cup\{\langle I'_{i,n+1}\rangle \rightarrow \langle I'_{i+1,1}\rangle\}\\
&\cup\{[\langle I'_{n+1,1}\rangle]_2 \rightarrow [\ ]_2\langle D,1\rangle\}
\end{aligned}
$$

In the above evolution rules for the inner membrane, object $\langle C_i\rangle$ is created from object $\langle S'_{n+1}\rangle$ and used for counting down the size of the subset in the membrane. According to the three conditions for the two objects, $\langle C_i\rangle$ and $\langle k\rangle$ created in Step 3, one of the following three procedures is executed when the counting is finished.

- **In the case that objects $\langle C_{n+1}\rangle$ and $\langle 0\rangle$ are in the membrane:**

  The size of the subset in the membrane is the same as that of the maximum independent set. Therefore, object $\langle I'_{1,1}\rangle$ is created to check whether the subset of vertices is an independent set, as in Step 3. In the case that the object $\langle I'_{i,j}\rangle$ becomes object $\langle I'_{n+1,1}\rangle$, the subset in the membrane is the maximum independent set. Then, object $\langle D,1\rangle$ is sent out to the outer membrane. In the other case, object $\langle E_{1,1}\rangle$ is created to delete all objects in the membrane.

- **In the case that objects $\langle C_{n+1}\rangle$ and $\langle k\rangle$ $(k\ge 0)$ are in the membrane:**

  The size of the subset in the membrane is more than that of the maximum independent set, and the subset is not the maximum independent set. Then, object $\langle E_{1,1}\rangle$ is created to delete all objects in the membrane.

- **In the other case:**

  The size of the subset in the membrane is less than that of the maximum independent set, so the subset is not the maximum independent set. Therefore, object $\langle E_{1,1}\rangle$ is created to delete all objects in the membrane.

Object $\langle E_{1,1}\rangle$ is created in the membrane when one of the above three procedures is finished. Object $\langle E_{1,1}\rangle$ deletes all objects in the membrane as well as the object $\langle T_{1,1}\rangle$ in Step 3. When the deletion is finished, object $\langle D,1\rangle$ is created for each deleted membrane, and the inner membrane is dissolved in the case that the subset in the membrane is not the maximum independent set. Therefore, 2^n objects, $\langle D,1\rangle$, are created in the outer membrane after checking all inner membranes. Since object $\langle D,k^{i+1}\rangle$ is created with two $\langle D,k^i\rangle$s according to the above evolution rules, $\langle D,k^n\rangle$ is evolved into object $\langle B\rangle$, which triggers the computation of Step 6, at the end of Step 5.

In Step 6, one of the inner membranes, which includes the maximum independent set, is dissolved, and the result is sent out from the outer membrane. Step 6 is executed by applying the following set of evolution rules.

(Evolution rules for the outer membrane)

$$
\begin{aligned}
R_{1,6} \;=\; &\{\langle B\rangle[\ ]_2 \rightarrow [\langle B\rangle]_2\}\\
&\cup\{\langle O_i\rangle\langle V_i,W\rangle \rightarrow \langle O_{i+1}\rangle\langle A_i\rangle\langle V_i,W\rangle \mid 1\le i\le n, W\in\{0,1\}\}\\
&\cup\{[\langle V_i,W\rangle\langle A_i\rangle]_1 \rightarrow [\ ]_1\langle V_i,W\rangle \mid 1\le i\le n, W\in\{0,1\}\}
\end{aligned}
$$

(Evolution rules for the inner membrane)

$$R_{2,6} \quad = \quad \{[\langle B\rangle\langle e_{1,1},F\rangle]_2 \rightarrow \langle O_1\rangle\}$$

At the beginning of Step 6, object $\langle B\rangle$ is moved into one of the inner membranes, which is selected non-deterministically. Then, the membrane is dissolved by the object, and object $\langle O_1\rangle$ is created with other objects in the inner membrane. Next, a set of output objects $\{\langle V_i,W\rangle \mid W \in \{0,1\}\}$ is sent out from the outer membrane to the outside region by auxiliary objects $\langle O_{i+1}\rangle$ and $\langle A_i\rangle$.

We now summarize the asynchronous P system Π_{MIS} for the maximum independent set as follows. (In the description of the P system, R_1 and R_2 are set of evolution rules in the above.)

$$\Pi_{MIS} = (O,\mu,\omega_1,\omega_2,,R_1,R_2,i_{in},i_{out})$$

- $$\begin{aligned} O \quad = \quad & \{\langle e_{i,j},W\rangle \mid 1 \le i \le n, 1 \le j \le n, W \in \{T,F\}\} \\ & \cup\{\langle k\rangle \mid 0 \le k \le n\} \\ & \cup O_S = \{\langle V_i,A\rangle \mid 1 \le i \le n, A \in \{0,1\}\} \\ & \cup\{\langle M_{i,j}\rangle,\langle T_{i,j}\rangle,\langle E_{i,j}\rangle \mid 1 \le i \le n+1, 1 \le j \le n+2\} \\ & \cup\{\langle S_i\rangle,\langle S_i^{'}\rangle,\langle C_i\rangle,\langle O_i\rangle,\langle A_i\rangle \mid 1 \le i \le n+1\} \\ & \cup\{\langle D,2^k\rangle \mid 0 \le k \le n\} \\ & \cup\{\langle I_{i,j}\rangle,\langle I_{i,j}^{'}\rangle \mid 1 \le i \le n+1, 1 \le j \le n+1\} \\ & \cup\{\langle T\rangle,\langle B\rangle\} \\ & \cup\{\langle V_i,A\rangle \mid 1 \le i \le n, A \in \{0,1\}\} \\ & \cup\{\langle l,2^k\rangle \mid 0 \le l \le n, 0 \le k \le n\} \end{aligned}$$

$$\mu = [\ [\ ]_2\ ]_1, \qquad \omega_1 = \omega_2 = \phi$$

4.4 Example execution of the P system

Figures 3 and 4 illustrate an example execution of the proposed P system Π_{MIS} for the graph in Figure 1.

Figure 3 illustrates Steps 1, 2 and 3. In the first step, a set of objects O_E is given from the outside region into the outer membrane. Then, two sets of evolution rules, $R_{1,1} \cup R_{2,1}$, are applied, and all objects are moved into the inner membrane.

In the next step, each inner membrane is repeatedly divided to create objects denoting all possible subsets of vertices by applying $R_{2,2}$.

In Step 3, each inner membrane is checked to determine whether the subset of vertices is an independent set. In the case that the subset is not an independent set, all objects are deleted in the membrane, and object $\langle 0,1\rangle$ is sent out from the membrane. Otherwise, object $\langle k,1\rangle$ is sent out from the membrane. In this example, two kinds of objects, $\langle 2,1\rangle$ and $\langle 1,1\rangle$, are sent out from the inner membrane, and the objects are merged into object $\langle 2\rangle$ by applying $R_{1,3}$.

Figure 4 illustrates Steps 4, 5 and 6. In Step 4, object $\langle 2\rangle$ is sent into the remaining membrane by applying $R_{1,4}$. Then, each inner membrane with $\langle 2\rangle$ is repeatedly divided again to create objects denoting all possible subsets of vertices by applying $R_{2,4}$.

In Step 5, each inner membrane is checked to determine whether the subset of vertices is the maximum independent set. In the case that the subset is not the maximum independent set whose size is two, all objects are deleted in the membrane.

In the final step, one of the inner membranes is dissolved by applying $R_{1,6}$. A set of output objects $\{\langle V_1,0\rangle,\langle V_2,1\rangle,\langle V_3,0\rangle,\langle V_4,1\rangle\}$ is sent out to the outside region by applying $R_{1,6}$.

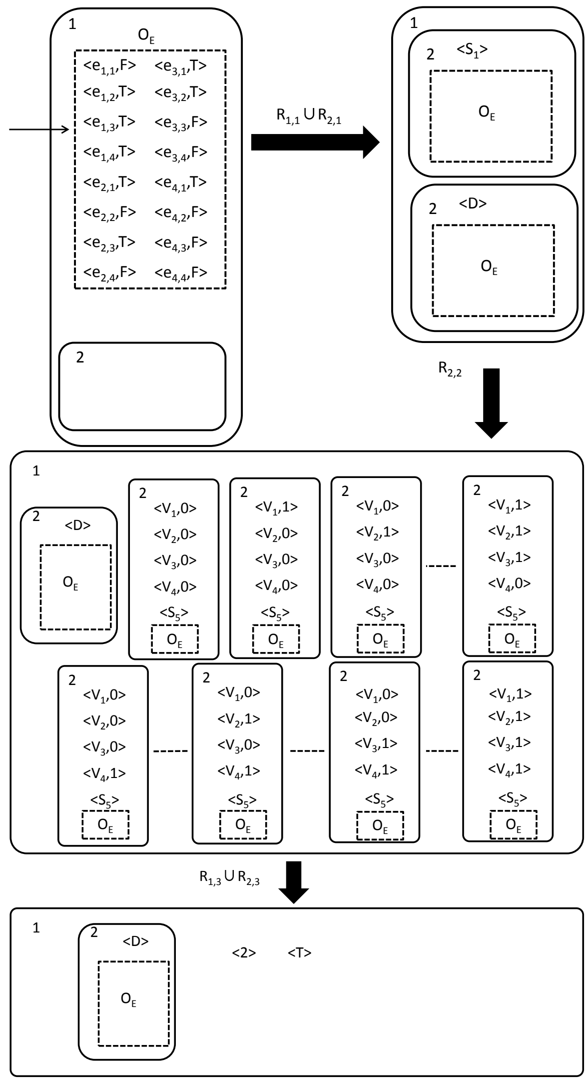

Figure 3: An example execution of Π_{MIS} (Steps 1 – 3)

4.5 Complexity

We now consider the complexity of asynchronous P system Π_{MIS}. Since $O(n^2)$ objects are moved sequentially in Step 1, both the numbers of sequential and parallel steps in Step 1 are $O(n^2)$. In Step 2, $O(2^n)$ membranes are created, and the numbers of sequential and parallel steps of Step 2 are $O(2^n)$ and $O(n)$, respectively. In Step 3, the numbers of sequential and parallel steps are $O(n^2 \cdot 2^n)$ and $O(n^2)$, respectively, since each subset is checked sequentially. In Step 4 and Step 5, the procedure is almost the same as that of Step 2 and Step 3. Therefore, the numbers of sequential and parallel steps of Step 4 are $O(2^n)$ and $O(n)$, respectively, and the numbers of sequential and parallel steps of Step 5 are $O(n^2 \cdot 2^n)$ and $O(n^2)$, respectively. Since $O(n)$ objects are sent out as the output sequentially in Step 6, both the numbers of sequential and parallel steps in Step 6 are $O(n)$.

Since the number of types of objects in P system Π_{MIS} is $O(n^2)$, and $O(n^3)$ kinds of evolution rules are used in Step 3, we obtain the following theorem for Π_{MIS}.

Theorem 2 *The asynchronous P system Π_{MIS}, which computes the maximum independent set for a graph with n vertices, works in $O(n^2 \cdot 2^n)$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$*

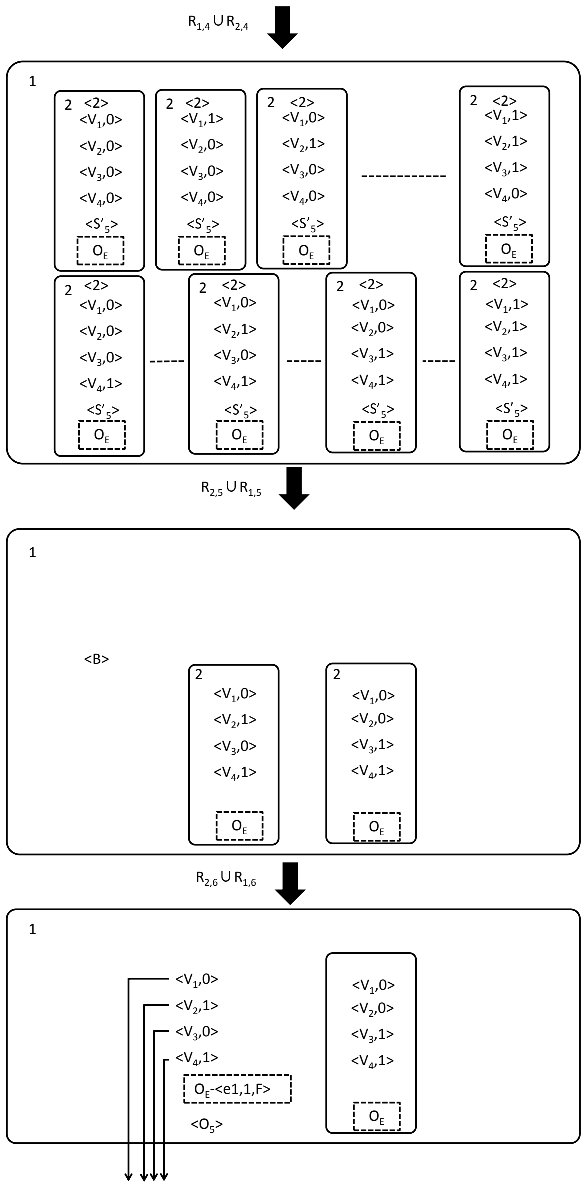

Figure 4: An example execution of Π_{MIS} (Steps 4 – 6)

types of objects and evolution rules of size $O(n^3)$. □

5 Minimum vertex cover and maximum clique

5.1 Input and output

Given a graph $G = (V, E)$ such that $V = \{v_1, v_2, \cdots v_n\}$, a vertex cover is defined as the subset $V' \subseteq V$ such that each edge of the graph is incident to at least one vertex in V'. In addition, the minimum vertex cover is a problem that finds the smallest vertex cover for the graph. For example, given the graph in Figure 1, the set of vertices $\{v_1, v_2\}$ is the minimum vertex cover for the graph.

On the other hand, a clique for a graph is defined as the subset $V' \subseteq V$ such that any pair of two vertices in V' has an edge in E. In addition, the maximum clique is a problem that finds the largest clique for the graph. For example, given the graph in Figure 1, the set of vertices $\{v_1, v_2, v_3\}$ is the maximum clique for the graph.

The minimum vertex cover and the maximum clique are well known to be reducible to the maximum independent set in a polynomial number of steps from the following two properties.

- A subset $V' \subseteq V$ is the minimum vertex cover if and only if a complement of vertices $V - V'$ is the maximum independent set.
- A subset $V' \subseteq V$ of vertices is the maximum clique if and only if V' is the maximum independent set for a complement graph $G' = (V, E')$ such that $e \in E'$ if and only if $e \notin E$.

Using the above properties, we consider two asynchronous P systems Π_{MVC} and Π_{MC} for the minimum vertex cover and the maximum clique by using reduction to the asynchronous P system Π_{MIS}, which is proposed in the previous section.

The input O_E is the same set of objects used for the maximum independent set.

$$O_E = \{\langle e_{i,j}, W\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, W \in \{T, F\}\}$$

The output of the P system is denoted by using the set of n objects given below as well as the output of the maximum independent set.

$$O_S = \{\langle V_i, A\rangle \mid 1 \leq i \leq n, A \in \{0, 1\}\}$$

We also assume that a set of input objects is given from the outside region into the skin membrane, and the output object is sent out from the skin membrane to the outside region.

5.2 Overview of the asynchronous P system

We now describe an overview of the asynchronous P systems Π_{MVC} and Π_{MC}, which are for the minimum vertex cover and the maximum clique, respectively. Both of P systems forms membrane structure $[\ [\ ]_{MIS}\ ]_1$. The membrane $[\ ]_{MIS}$ is the P system Π_{MIS}, which is described in the previous section.

The reductions in the asynchronous P systems Π_{MVC} and Π_{MC} are executed as follows.

Π_{MVC}:

Step 1 Compute the maximum independent set for the instance and send out the result to the outer membrane.

Step 2 Reduce the output and send out the reduced output from the outer membrane.

Π_{MC}:

Step 1 Reduce input objects of the maximum clique to an instance of the maximum independent set.

Step 2 Compute the maximum independent set for the instance and send out the result from the outer membrane.

We next show details of the reduction in the asynchronous P systems Π_{MVC} and Π_{MC}.

Details of Π_{MVC}:

In Step 1, the maximum independent set is computed for the input in the inner P system, and the result is sent out to the outer membrane. The set of evolution rules are almost [2] the same as those for P system Π_{MIS}.

[2] The different part is the only label of membrane whose evolution rules are applied.

In Step 2, the output is reduced and the reduced output is sent out from the out-most membrane. Step 2 is executed by applying the following set of evolution rules.
(Evolution rules for the outer membrane)

$$\begin{aligned} R_{1,2} &= \{\langle O_1\rangle\langle V_1, W\rangle \rightarrow \langle RO_2\rangle\langle RA_1\rangle\langle V_1, W\rangle | W \in \{0,1\}\} \\ &\cup\{\langle RO_i\rangle\langle V_i, W\rangle \rightarrow \langle RO_{i+1}\rangle\langle RA_i\rangle\langle V_i, W\rangle \mid 2 \leq i \leq n, W \in \{0,1\}\} \\ &\cup\{[\langle V_i, W\rangle\langle RA_i\rangle]_1 \rightarrow [\ ]_1\langle V_i, W'\rangle \mid 0 \leq i \leq n, W \in \{0,1\}, W' \in \{0,1\}, W \neq W'\} \end{aligned}$$

In the above evolution rules, the output objects are reduced for the minimum vertex cover. In the case, a set of output objects $\{\langle V_i, W\rangle | W \in \{0,1\}\}$ must be changed into a complement of the vertices. Therefore, auxiliary objects, $\langle RO_{i+1}\rangle$ and $\langle RA_i\rangle$, are used for creating a complement of $\langle V_i, W\rangle$, and the object is sent out from the outer membrane to the outside region.

Details of Π_{MC}:
In Step 1, input objects are reduced to an instance of the maximum independent set. Step 1 is executed by applying the following set of evolution rules.
(Evolution rules for the outer membrane)

$$\begin{aligned} R_{1,1} &= \cup\{\langle RE_{i,j}\rangle\langle e_{i,j}, T\rangle \rightarrow \langle RE_{i,j+1}\rangle\langle e_{i,j}, F\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, i \neq j\} \\ &\cup\{RE_{i,j}\rangle\langle e_{i,j}, F\rangle \rightarrow \langle RE_{i,j+1}\rangle\langle e_{i,j}, T\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n, i \neq j\} \\ &\cup\{\langle RE_{i,i}\rangle \rightarrow \langle RE_{i,i+1}\rangle \mid 1 \leq i \leq n\} \\ &\cup\{\langle RE_{i,n+1}\rangle \rightarrow \langle RE_{i+1,1}\rangle \mid 1 \leq i \leq n\} \\ &\cup\{\langle RE_{n+1,1}\rangle \rightarrow \langle M'_{1,1}\rangle\} \\ &\cup\{\langle M'_{1,1}\rangle[\ ]_{MIS} \rightarrow [\langle M'_{2,1}\rangle]_{MIS}\} \\ &\cup\{\langle e_{1,1}, F\rangle[\ ]_{MIS} \rightarrow [\langle e_{1,1}, F\rangle\langle RE_{1,1}\rangle]_{MIS}\} \\ &\cup\{\langle M'_{i,j}\rangle\langle e_{i,j}, W\rangle[\ ]_{MIS} \rightarrow [\langle M'_{i+1,j}\rangle\langle e_{i,j}, W\rangle]_{MIS} \mid 2 \leq i \leq n, 1 \leq j \leq n, W \in \{T, F\}\} \\ &\cup\{\langle M'_{n+1,j}\rangle \rightarrow \langle M'_{1,j+1}\rangle \mid 1 \leq j \leq n\} \end{aligned}$$

(Evolution rules for the membrane $[\ ]_{MIS}$)

$$\begin{aligned} R_{MIS,1} &= \{[\langle M'_{i,j}\rangle]_{MIS} \rightarrow [\ ]_{MIS}\langle M'_{i,j}\rangle \mid 2 \leq i \leq n+1, 1 \leq j \leq n+1\} \\ &\cup\{[\langle RE_{1,1}\rangle]_{MIS} \rightarrow [\ ]_{MIS}\langle RE_{1,2}\rangle\} \end{aligned}$$

In the above evolution rules, input objects for the maximum clique are reduced into input objects for the maximum independent set. The input graph must be changed into a complement graph in the case of the maximum clique. First, object $\langle RE_{1,1}\rangle$ is created for starting the procedure, and then each $\langle RE_{i,j}\rangle$ is used for creating a complement of the edge $e_{i,j}$.

After creating the reduced input objects for the maximum independent set, object $\langle M'_{1,1}\rangle$ is created, and the object starts movement of the reduced object from the outer membrane into the membrane $[\ ]_{MIS}$. Object $\langle M'_{i,j}\rangle$ brings the reduced input objects $\langle e_{i,j}, W\rangle$ into the membrane $[\ ]_{MIS}$.

In Step 2, the maximum independent set is computed for the reduced input, and the result is sent out from the outer membrane. The set of evolution rules are almost [3] the same as those for P system Π_{MIS}.

5.3 Complexity

Since the numbers of sequential and parallel steps of the reductions in are $O(n^2)$, we obtain the following theorems for Π_{MVC} and Π_{MC} from Theorem 2.

[3] The different parts are the label of membrane whose evolution rules are applied and the set of evolution rules $R_{MIS,1}$.

Theorem 3 *The asynchronous P system Π_{MVC}, which computes the minimum vertex cover for a graph with n vertices, works in $O(n^2 \cdot 2^n)$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ types of objects and evolution rules of size $O(n^3)$.* □

Theorem 4 *The asynchronous P system Π_{MC}, which computes the maximum clique for a graph with n vertices, works in $O(n^2 \cdot 2^n)$ sequential steps or $O(n^2)$ parallel steps by using $O(n^2)$ types of objects and evolution rules of size $O(n^3)$.* □

6 Conclusions

In the present paper, we proposed P systems that solve four graph problems. The proposed P systems are fully asynchronous, i.e., any number of applicable rules may be applied in one step of the P system. The first and second P systems solve the minimum graph coloring and the maximum independent set. The third and fourth P systems solve the minimum vertex cover and the maximum clique by reduction to the maximum independent set.

We showed that the proposed P systems work in a polynomial number of steps in the maximal parallel manner and also showed that the P systems work sequentially. Although the number of sequential steps is exponential, the result means that the proposed P systems work for any combinations of sequential and asynchronous applications of evolution rules, and guarantees that the P systems can output a correct solution in the case that any number of evolution rules are synchronized.

As future work, we are considering an asynchronous P system using a fewer number of membranes and evolution rules. In addition, we are considering a general reduction from a conventional P system to an asynchronous P systems.

Acknowledgments

This research was partially supported by JSPS KAKENHI, Grand-in-Aid for Scientific Research (C), 24500019.

References

[1] R. Freund. Asynchronous P systems and P systems working in the sequential mode. *WMC'04 Proceedings of the 5th International Conference on Membrane Computing*, 3365:36–62, 2005.

[2] A. Leporati and C. Zandron. P systems with input in binary form. *International Journal of Foundations of Computer Science*, 17:127–146, 2006.

[3] T. Murakawa and A. Fujiwara. Asynchronous P system for arithmetic operations and factorization. *Proceedings of 3rd International Workshop on Parallel and Distributed Algorithms and Applications*, 2011.

[4] L. Pan and A. Alhazov. Solving HPP and SAT by P systems with active membranes and separation rules. *Acta Informatica*, 43(2):131–145, 2006.

[5] G. Păun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1), 2000.

[6] G. Păun. P system with active membranes: Attacking NP-complete problems. *Journal of Automata, Languages and Combinatorics*, 6(1):75–95, 2001.

[7] G. Păun. Introduction to membrane computing. *Applications of Membrane Computing, Springer*, 2006.

[8] H. Tagawa and A. Fujiwara. Solving SAT and Hamiltonian cycle problem using asynchronous P systems. *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Science*, E95-D(3), 2012.

[9] T. Tateishi and A. Fujiwara. Logic and arithmetic operations with a constant number of steps in membrane computing. *International Journal of Foundations of Computer Science*, 22(3):547–564, 2011.

[10] C. Zandron, G. Rozenberg, and G. Mauri. Solving NP-complete problems using P systems with active membranes. *Proceedings of the Second International Conference on Uncoventional Models of Computation*, 289–301, 2000.