International Journal of Networking and Computing – `www.ijnc.org`, ISSN 2185-2847
Volume 13, Number 2, pages 131–148, July 2023

An asynchronous P system with the Bron-Kerbosch algorithm for solving the maximum clique

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

Received: February 9, 2023
Revised: April 14, 2023
Accepted: May 25, 2023
Communicated by Sayaka Kamei and Fukuhito Ooshita

Abstract

Membrane computing, which is also known as a P system, is a computational model inspired by the activity of living cells. Several P systems, which work in a polynomial number of steps, have been proposed for solving computationally hard problems. However, most of the proposed algorithms use an exponential number of membranes, and reduction of the number of membranes must be considered in order to make a P system a more realistic model.

In the present paper, we propose asynchronous P systems based on the Bron-Kerbosch algorithm for solving the maximum clique problem with fewer membranes. The proposed P systems solve the maximum clique with n vertices in $O(n^2)$ parallel steps or $O(n^2 2^n)$ sequential steps.

We evaluate the number of membranes used in the proposed P systems by comparing with the numbers of membranes used in known P systems. Our experimental results demonstrate the validity and efficiency of the proposed P systems.

Keywords: membrane computing, asynchronous P system, maximum clique, Bron-Kerbosch algorithm

1 Introduction

Membrane computing, which was introduced in [7] in terms of P systems, is a computational model inspired by the activity of living cells. A P system is with respect to membranes and objects which represent computing cells and data storage, respectively. In a P system, each object evolves according to certain evolution rules associated with the membrane.

An exponential number of membranes can be created in a polynomial number of steps using a division rule on a P system, allowing a computationally hard problem to be solved in a polynomial number of steps. Using this feature, a number of P systems have been proposed for solving computationally hard problems [4, 6, 8, 12]. This exponential number of membranes corresponds to the number of living cells, which must be reduced when implementing a P system because living cells cannot be created in exponential numbers.

Recently, various P systems [2, 5, 10, 11] have been proposed for reducing the number of membranes. For example, a P system for the maximum independent set [11] using branch and bound, which is a well-known optimization technique, has been proposed. In this P system, only adjacent vertices are checked for partial assignment. The experimental results with this P system show that the number of membranes used is reduced by at most 97 percent relative to the previous P system.

In the present paper, we propose two asynchronous P systems for solving the maximum clique problem with the Bron-Kerbosch algorithm [1]. Bron-Kerbosch algorithm is a search algorithm for finding all maximal cliques, and consists of recursive backtracking using three disjoint sets of vertices. We show that the theoretical complexity of the proposed P systems is $O(n^2 2^n)$ sequential steps or $O(n^2)$ parallel steps using $O(n^3)$ kinds of objects.

We evaluate the number of membranes in the proposed P systems and compare it to the numbers of membranes in known P systems. The experimental results show that the number of membranes is smaller in the proposed systems.

The remainder of the paper is organized as follows. In Section 2, we describe the computational model for the membrane computing and an outline of the Bron-Kerbosch algorithm. In Section 3, we propose P systems with the Bron-Kerbosch algorithm for solving maximum clique and show examples of executions of the P systems. Then, we consider a complexity of the proposed P systems. In Section 4, we show experimental results for the previous P systems and proposed P systems. Finally, Section 5 concludes the paper.

2 Preliminaries

2.1 Computational model for membrane computing

A P system consists mainly of membranes and objects. A membrane is a computing cell in the P system and may contain objects and other membranes. Each membrane is initially labeled with a distinct integer. An object represents a memory cell that stores data in the P system. According to the evolution rules for the corresponding membrane, objects may evolve into other objects or pass through membranes. Objects may also divide or dissolve the membranes in which the objects are stored. We assume that each object is a finite string over a given set of alphabetic characters.

As an example of membranes and objects, the following expression defines a membrane structure comprising two membranes and three objects.

$$[\ [\ \alpha\ ]_2\ [\ \beta\ \gamma\ ]_3\ ]_1$$

In this example, the membrane labeled 1 contains two membranes, labeled 2 and 3, which contain sets of objects $\{\alpha\}$ and $\{\beta, \gamma\}$, respectively.

The computation of P systems is governed by a number of evolution rules. Each evolution rule is a rule for updating the membranes and objects. According to the applicable evolution rules, objects and membranes are transformed in parallel in every step of the computation. The system stops the computation if there is no applicable evolution rule for the objects.

Various types of evolution rules are assumed in membrane computing. In the present paper, we assume the following five rules as in [3]:

(1) Object evolution rule: $[\ \alpha\ ]_h \rightarrow [\ \beta\ ]_h$

Object α is transformed into object β.

(2) Send-in communication rule: $\alpha\ [\ ]_h \rightarrow [\ \beta\ ]_h$

Object α is moved into inner membrane h and is transformed into object β.

(3) Send-out communication rule: $[\ \alpha\ ]_h \rightarrow [\ ]_h\ \beta$

Object α is sent out from membrane h and is transformed into object β.

(4) Dissolution rule: $[\ \alpha\ ]_h \rightarrow \beta$

The membrane that contains object α is dissolved, and object α is transformed into object β. (Note that the outermost membrane cannot be dissolved.)

(5) Division rule: $[\ \alpha\ ]_h \rightarrow [\ \beta\ ]_h[\ \gamma\ ]_h$

The membrane that contains object α is divided into two membranes with the same label, and object α is transformed into other objects, β and γ, each in one of the created membranes.

The P system consists of the following six components:

O**:** the set of objects used in the system,

μ**:** the structure of the membrane,

ω_i**:** the set of objects initially contained in the membrane labeled i,

R_i**:** the set of evolution rules for the membrane labeled i,

i_{in}**:** the label of the input membrane, and

i_{out}**:** the label of the output membrane

Using the above components, a P system Π with m membranes is defined as follows:

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

Under the assumption that each of the evolution rules can be applied in a single step in the computational model, the complexity of the P system is defined as the number of steps executed.

In the present paper, we consider asynchronous parallelism [8] in the P system. Under the assumption of asynchronous parallelism, any number of applicable evolution rules are applied in parallel. In other words, all objects, for which there are applicable evolution rules, can be transformed in parallel, or only one of the applicable evolution rules is applied in each step of the computation. We refer to the numbers of steps in the former and latter cases as the numbers of parallel and sequential steps, respectively. The number of parallel steps is the complexity of the P system in the best case, and the number of sequential steps is the complexity in the worst case.

2.2 Bron-Kerbosch algorithm for maximum clique

I assume undirected graph $G = (V, E)$, where $V = \{1, 2, ..., n\}$ is the vertex set of G, and E is the edge set of G. A clique C is a set of vertices with the property that every pair of vertices in the set are adjacent such that $G(C)$ is complete. A maximal clique is a clique that cannot be extended by including one more adjacent vertex. A maximum clique is a maximal clique that has the maximum cardinality. The clique number of G, denoted by $\omega(G)$, is the size of the maximum clique.

The maximum clique problem, which is a well-known computationally hard problem, is the problem of finding the $\omega(G)$ in a given graph:

$$\omega(G) = \max\{|S| : S \text{ is a clique in } G\}$$

Note that the maximum clique and the maximum independent set are complementary. An independent set is a subset of V, whose elements are pairwise nonadjacent. In other words, the maximum clique of a graph G is an independent set of a complement graph of G, and vice versa.

In this paper, we propose P systems based on the Bron-Kerbosch algorithm [1] for the maximum clique problem. The Bron-Kerbosch algorithm is a recursive backtracking algorithm and is easily implemented. Using the Bron-Kerbosch algorithm, all maximal cliques are found, from which the maximum clique can be determined.

Two versions of the Bron-Kerbosch algorithm are a basic version and a pivoting version.

2.2.1 Basic

We first explain a basic Bron-Kerbosch algorithm, which is shown as Algorithm 1. The basic Bron-Kerbosch algorithm maintains three disjoint sets of vertices as input parameters P, X, and R, which are given as follows.

- P is the set of vertices that have not been considered yet.
- X is the set of vertices that have already been considered as options.
- R is the set of vertices in a clique.

In the algorithm, R and X are initially set as empty, and P is defined as the set of all vertices in an input graph. For each recursive call, the algorithm checks whether the given clique R is maximal. A recursive procedure is called for each vertex v in P. Recursive function are executed using three arguments, $R \cup \{v\}$, $P \cap N(v)$, and $X \cap N(v)$, where $N(v)$ is a set of neighbors of v. Then, v is moved from P to X to exclude v from consideration of the clique. If both P and X are empty, there is no vertex that can be added to R, which means that R is the maximal clique.

2.2.2 Pivoting

We introduce pivoting as an optimization technique in the Bron-Kerbosch algorithm shown in Algorithm. 2. In this algorithm, a pivot vertex is chosen to decrease the number of recursive calls. The idea of pivoting is based on the logic that any maximal clique must include either vertex u, which is chosen as the pivot, or one of the non-neighbors of u. If neither u nor any of the non-neighbors of u are included in R, then the clique cannot be maximal because R consists of only neighbors of u, and u can be added to R. Therefore, checks for neighbors of an arbitrary pivot u can be omitted, and the number of recursive calls is decreased by the number of omitted vertices.

3 Asynchronous P system with Bron-Kerbosch algorithm

3.1 Input and output for proposed P system

We first show an encoding for the input and output for the P system. We assume that the input for the maximum clique problem is an undirected graph $G = (V, E)$ with n vertices. An output of the problem is the maximum subset of vertices such that every pair of the vertices are adjacent. For example, we assume that the graph in Figure 1 is the input. Then, both subsets $V' = \{v_1, v_2, v_3\}$ and $V'' = \{v_3, v_4\}$ are maximal cliques, and V' is the maximum clique.

The above input is given by the following set of objects O_E in the P system:

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

In the above set of input objects, an edge (v_i, v_j) is represented by the objects $\langle e_{i,j}, W\rangle$. If edge (v_i, v_j) is in the graph, object W is set to T, otherwise set to F. For example, the following set of objects is given as an input of the P system, O_E, for the graph in Figure 1:

$$\begin{aligned} O_E =\{&\langle e_{1,1}, F\rangle, \langle e_{1,2}, T\rangle, \langle e_{1,3}, T\rangle, \langle e_{1,4}, F\rangle, \\ &\langle e_{2,1}, T\rangle, \langle e_{2,2}, F\rangle, \langle e_{2,3}, T\rangle, \langle e_{2,4}, F\rangle, \\ &\langle e_{3,1}, T\rangle, \langle e_{3,2}, T\rangle, \langle e_{3,3}, F\rangle, \langle e_{3,4}, T\rangle, \\ &\langle e_{4,1}, F\rangle, \langle e_{4,2}, F\rangle, \langle e_{4,3}, T\rangle, \langle e_{4,4}, F\rangle\} \end{aligned}$$

We assume that a computation on the P system starts if the above O_E is given as input from the outside region into the outermost membrane. The output of the P system is the following set of objects, which represents a subset of vertices:

$$O_C = \{\langle R_i, W\rangle \mid 1 \leq i \leq n, W \in \{T, F\}\}$$

Algorithm 1: A basic Bron-Kerbosch algorithm

```
R={}, P=V, X={}

BK(R, P, X){
  if P and X are both empty:
    return R
  for each vertex v in P:
    BK(R∪{v}, P∩N(v), X∩N(v))
    P = P-{v}
    X = X∪{v}
}
```

Algorithm 2: A pivoting Bron-Kerbosch algorithm

```
R={}, P=V, X={}

BK-Pivot(R, P, X){
  if P and X are both empty:
    return R
  choose a pivot vertex u in P∪X
  for each vertex v in P-N(u):
    BK-Pivot(R∪{v}, P∩N(v), X∩N(v))
    P = P-{v}
    X = X∪{v}
}
```

Object R_i in $\langle R_i, W\rangle$ indicates the i-th vertex, and the object W is set to T when the i-th vertex is included in the maximum clique, otherwise set to F.

For example, the following is an output of the P system, which represents the maximum clique for the graph in Figure 1:

$$O_C = \{\langle R_1, T\rangle, \langle R_2, T\rangle, \langle R_3, T\rangle, \langle R_4, F\rangle\}$$

3.2 Outline of the P system with basic Bron-Kerbosch algorithm

We first explain an outline of a P system based on a basic Bron-Kerbosch algorithm. The proposed P system consists of two membranes $[\ [\ ]_2\ ]_1$, i.e., an inner membrane labeled 2 contained in an outer membrane labeled 1. The P system consists of the following 6 steps:

Step 1: Move a set of input objects into an inner membrane and create objects denoting R, P and X.

Step 2: In the inner membrane, choose a candidate vertex in P. The inner membrane is divided into two membranes in case that the vertex is included in the clique or not included.

Step 3: In the inner membrane that a new vertex is added to the clique, an objects that represent P, R and X are updated. In the other membrane, objects that representing P and X are updated.

Step 4: In each inner membrane, check whether the vertex are remained in P. In case that no vertex is in P, determine whether the clique is maximal or not by checking objects denoting X. In the other case, the above procedure is repeated from Step 2.

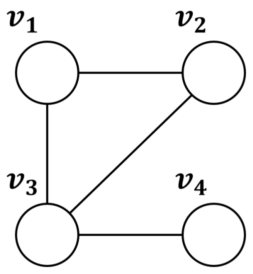


Figure 1: Example of an input undirected graph

Step 5: In each inner membrane that contains the maximal clique, send out objects that denote the clique size to the outer membrane. Dissolve the other inner membranes, which contains no maximal clique.

Step 6: Send out objects that denote the maximum clique from the outer membrane after results are sent out from all inner membranes.

3.3 Details of P system with the basic Bron-Kerbosh algorithm

We now explain the details of each step of the P system with the basic Bron-Kerbosch algorithm. In the following description, $R_{i,j,k}$ denotes a set of evolution rules applied to membrane i in Step j. k is a number to distinguish each operation. O_E, which denotes an input formula, is given to the outer membrane.

Step 1: A set of input objects is moved into an inner membrane and generate objects denoting R, P and X. This step is executed using the following evolution rules.

(Evolution rules for outer membrane)

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

(Evolution rules for inner membranes)

$$\begin{aligned} R_{2,1,1} &= \{[\langle M_{1,j}\rangle]_2 \rightarrow []_2\langle M_{1,j}\rangle\langle C_R, j, 0\rangle \mid 2 \leq j \leq n\} \\ &\cup\{[\langle M_{i,j}\rangle]_2 \rightarrow []_2\langle M_{i,j}\rangle \mid 2 \leq i \leq n, 1 \leq j \leq n\} \\ &\cup\{\langle M_{i,n+1}\rangle \rightarrow \langle M_{i+1,1}\rangle \mid 1 \leq i \leq n\} \\ &\cup\{\langle M_{n+1,1}\rangle \rightarrow \langle MakR\rangle\} \\ R_{2,1,2} &= \{\langle MakR\rangle \rightarrow \langle R_1, F\rangle\langle R_2\rangle\} \\ &\cup\{\langle R_i\rangle \rightarrow \langle R_i, F\rangle\langle R_{i+1}\rangle \mid 2 \leq i \leq n\} \\ &\cup\{\langle R_{n+1}\rangle \rightarrow \langle MakP\rangle\} \\ R_{2,1,3} &= \{\langle MakP\rangle \rightarrow \langle P_1, T\rangle\langle P_2\rangle\} \\ &\cup\{\langle P_i\rangle \rightarrow \langle P_i, T\rangle\langle P_{i+1}\rangle \mid 2 \leq i \leq n\} \\ &\cup\{\langle P_{n+1}\rangle \rightarrow \langle MakX\rangle\} \\ R_{2,1,4} &= \{\langle MakX\rangle \rightarrow \langle X_1, F\rangle\langle X_2\rangle\} \\ &\cup\{\langle X_i\rangle \rightarrow \langle X_i, F\rangle\langle X_{i+1}\rangle \mid 2 \leq i \leq n\} \\ &\cup\{\langle X_{n+1}\rangle \rightarrow \langle MakC\rangle\} \\ R_{2,1,5} &= \{\langle MakC\rangle \rightarrow \langle C_P, n\rangle\langle C_X, 0\rangle\langle k, n\rangle\langle ChoP\rangle\} \end{aligned}$$

The computation using the above evolution rules is executed as follows. All input objects in the outer membrane are moved into the inner membrane using object $\langle M_{i,j}\rangle$ according to evolution rules in $R_{1,1}$ and $R_{2,1,1}$. After objects representing edges of the first vertex are moved, the objects $\langle C_R, j, 0\rangle$, which constitute a counter giving the size and number of maximal cliques in R, are generated. For example, object $\langle C_R, 4, 2\rangle$ indicates 2 maximal cliques of size 4 are in the membrane. The objects are used in Step 6.

After all input objects are moved, objects $\langle R_i, F\rangle$, $\langle P_i, T\rangle$, and $\langle X_i, F\rangle$ are created according to evolution rules in $R_{2,1,2}$, $R_{2,1,3}$ and $R_{2,1,4}$. These objects represent R, P, and X in the P system. For example, object $\langle R_2, T\rangle$ represents that the second vertex is in R.

At the end of Step 1, objects $\langle C_P, n\rangle$ and $\langle C_X, 0\rangle$ are created for counting the numbers of vertices in P and X according to evolution rules in $R_{2,1,5}$. Object $\langle C_P, n\rangle$ indicates that n vertices are in P.

Step 2: A candidate vertex in P is chosen in the inner membrane for adding to the clique. Then, the inner membrane is divided into two membranes. The candidate vertex is included in the clique in one membrane, and the vertex is not included in the clique in the other membrane.

This step is executed using the following evolution rules.

(Evolution rules for inner membranes)

$$\begin{aligned}
R_{2,2,1} \;=\; & \{\langle ChoP\rangle\langle C_P, i\rangle\langle P_1, T\rangle \to \langle Set, 1\rangle\langle C_P, i\rangle\langle P_1, T\rangle\langle Div\rangle \mid 2 \le i \le n\} \\
& \cup\{\langle ChoP\rangle\langle C_P, 1\rangle\langle P_1, T\rangle \to \langle Set, 1\rangle\langle C_P, 1\rangle\langle P_1, T\rangle\langle NoDiv\rangle\} \\
& \cup\{\langle ChoP\rangle\langle P_1, F\rangle \to \langle P_1, F\rangle\langle ChoP, 2\rangle\} \\
& \cup\{\langle ChoP, j\rangle\langle C_P, i\rangle\langle P_j, T\rangle \to \langle Set, j\rangle\langle C_P, i\rangle\langle P_j, T\rangle\langle Div\rangle \mid 2 \le i \le n, 2 \le j \le n\} \\
& \cup\{\langle ChoP, j\rangle\langle C_P, 1\rangle\langle P_j, T\rangle \to \langle Set, j\rangle\langle C_P, 1\rangle\langle P_j, T\rangle\langle NoDiv\rangle \mid 2 \le j \le n\} \\
& \cup\{\langle ChoP, j\rangle\langle P_j, F\rangle \to \langle P_j, F\rangle\langle ChoP, j+1\rangle \mid 2 \le j \le n\} \\
R_{2,2,2} \;=\; & \{[\langle Div\rangle\langle k, i\rangle]_2 \to [\langle k, i-1\rangle\langle SV\rangle]_2[\langle k, i-1\rangle\langle NoSV\rangle]_2 \mid 1 \le i \le n\} \\
& \cup\{\langle NoDiv\rangle \to \langle SV\rangle\}
\end{aligned}$$

The computation using the above evolution rules is executed as follows. First, a vertex in P is chosen using the objects $\langle ChoP, j\rangle$ and $\langle P_j, T\rangle$, and then a trigger object $\langle Div\rangle$ and a object $\langle Set, j\rangle$ is generated according to evolution rules in $R_{2,2,1}$. The object $\langle Set, j\rangle$ means j-th vertex is selected from the set P. In the exceptional case that the count of P (j of $\langle C_P, j\rangle$) is 1, that is, only one vertex can be selected, division is not executed for the membrane, and the vertex must be contained in a maximal clique.

The membrane is divided into two membranes according to evolution rules in $R_{2,2,2}$. In the membrane containing the selected vertex, a trigger object $\langle SV\rangle$ is generated. In the other membrane, the trigger object $\langle NoSV\rangle$ is generated. The object $\langle k, i\rangle$ represents the number of unchecked vertices, e.g., three vertices have not been checked if the object is $\langle k, 3\rangle$.

Step 3: The objects that represent P, R, and X are updated in the membrane such that a new vertex is added to the clique. The new vertex v is added to R. In addition, P and X are updated such that $P \cap N(v)$ and $X \cap N(v)$. In the other membrane, the objects that represent the sets P and X are updated. Vertex v is removed from P and added to set X.

This step is executed using the following evolution rules.

(Evolution rules for inner membranes)

$$\begin{aligned}
R_{2,3,1} \;=\; & \{\langle SV\rangle \to \langle AltR\rangle\} \\
& \cup\{\langle AltR\rangle\langle Set, i\rangle\langle R_i, F\rangle \to \langle Set, i\rangle\langle R_i, T\rangle\langle AltP\rangle \mid 1 \le i \le n\} \\
& \cup\{\langle AltP\rangle\langle Set, i\rangle\langle e_{i,1}, T\rangle\langle P_1, T\rangle\langle C_P, m\rangle \to \langle Set, i\rangle\langle e_{i,1}, T\rangle\langle P_1, T\rangle\langle C_P, 1\rangle\langle AltP, 2\rangle \\
& \qquad \mid 1 \le i \le n, 1 \le m \le n\} \\
& \cup\{\langle AltP\rangle\langle Set, i\rangle\langle e_{i,1}, W\rangle\langle P_1, X\rangle\langle C_P, m\rangle \to \langle Set, i\rangle\langle e_{i,1}, W\rangle\langle P_1, F\rangle\langle C_P, 0\rangle\langle AltP, 2\rangle \\
& \qquad \mid 1 \le i \le n, 1 \le m \le n, W, X \in \{T, F\}, W \wedge X \ne T\} \\
& \cup\{\langle AltP, j\rangle\langle Set, i\rangle\langle e_{i,j}, T\rangle\langle P_j, T\rangle\langle C_P, m\rangle \to \langle Set, i\rangle\langle e_{i,j}, T\rangle\langle P_j, T\rangle\langle C_P, m+1\rangle\langle AltP, j+1\rangle \\
& \qquad \mid 1 \le i \le n, 2 \le j \le n, 1 \le m \le n\} \\
& \cup\{\langle AltP, j\rangle\langle Set, i\rangle\langle e_{i,j}, W\rangle\langle P_j, X\rangle \to \langle Set, i\rangle\langle e_{i,j}, W\rangle\langle P_j, F\rangle\langle AltP, j+1\rangle \\
& \qquad \mid 1 \le i \le n, 2 \le j \le n, 1 \le m \le n, W, X \in \{T, F\}, W \wedge X \ne T\} \\
& \cup\{\langle AltP, n+1\rangle \to \langle AltX\rangle\}
\end{aligned}$$

$$\begin{aligned}
R_{2,3,2} \;=\;& \{\langle AltX\rangle\langle Set,i\rangle\langle e_{i,1},T\rangle\langle X_1,T\rangle\langle C_X,m\rangle \to \langle Set,i\rangle\langle e_{i,1},T\rangle\langle X_1,T\rangle\langle C_X,1\rangle\langle AltX,2\rangle \\
&\quad \mid 1 \le i \le n, 1 \le m \le n\} \\
&\cup\{\langle AltX\rangle\langle Set,i\rangle\langle e_{i,1},W\rangle\langle X_1,X\rangle\langle C_X,m\rangle \to \langle Set,i\rangle\langle e_{i,1},W\rangle\langle X_1,F\rangle\langle C_X,0\rangle\langle AltX,2\rangle \\
&\quad \mid 1 \le i \le n, 1 \le m \le n, W,X \in \{T,F\}, W \wedge X \ne T\} \\
&\cup\{\langle AltX,j\rangle\langle Set,i\rangle\langle e_{i,j},T\rangle\langle X_j,T\rangle\langle C_X,m\rangle \to \langle Set,i\rangle\langle e_{i,j},T\rangle\langle X_j,T\rangle\langle C_X,m+1\rangle\langle AltX,j+1\rangle \\
&\quad \mid 1 \le i \le n, 2 \le j \le n, 1 \le m \le n\} \\
&\cup\{\langle AltX,j\rangle\langle Set,i\rangle\langle e_{i,j},W\rangle\langle X_j,X\rangle \to \langle Set,i\rangle\langle e_{i,j},W\rangle\langle X_j,F\rangle\langle AltX,j+1\rangle \\
&\quad \mid 1 \le i \le n, 2 \le j \le n, 1 \le m \le n, W,X \in \{T,F\}, W \wedge X \ne T\} \\
&\cup\{\langle AltX,n+1\rangle\langle Set,i\rangle \to \langle CheC_{PX}\rangle \mid 1 \le i \le n\} \\
R_{2,3,3} \;=\;& \{\langle NoSV\rangle\langle Set,i\rangle\langle P_i,T\rangle\langle X_i,F\rangle \to \langle P_i,F\rangle\langle X_i,T\rangle\langle -\, C_P\rangle \mid 1 \le i \le n\} \\
&\cup\{\langle -\, C_P\rangle\langle C_P,i\rangle \to \langle C_P,i-1\rangle\langle CheC_P\rangle \mid 1 \le i \le n\}
\end{aligned}$$

The computation using the above evolution rules is executed as follows. In the membrane that a new vertex is added, an object $\langle SV\rangle$ must be included. First, an object $\langle AltR\rangle$ is generated from object $\langle SV\rangle$. Object $\langle R_i,F\rangle$ is turned into $\langle R_i,T\rangle$ using objects $\langle Set,i\rangle$ and $\langle AltR\rangle$, which means that the i-th vertex is added to set R. Second, the set P is updated using loop counter objects $\langle AltP\rangle$ and $\langle AltP,j\rangle$. Object $\langle P_j,T\rangle$ remains TRUE only when both $\langle P_j,T\rangle$ and $\langle e_{i,j},T\rangle$ are TRUE, otherwise FALSE, according to evolution rules in $R_{2,3,1}$. The counter of P are initialized when the loop counter is 1. The Set X is updated similarly according to evolution rules in $R_{2,3,2}$.

In the other membrane, i.e., the vertex selected from the set P is not included, object $\langle NoSV\rangle$ must be included. The i-th vertex is removed from P and added to X using objects $\langle NoSV\rangle$ and $\langle Set,i\rangle$ according to evolution rules in $R_{2,3,3}$. Then, the count of P is decremented. (The counter of X does not need to be updated.)

Step 4: Existence of the vertex in P is checked in each inner membrane. If P contains no vertices, then confirming that the clique is maximal by checking the objects representing X. Otherwise, the procedure is repeated from Step 2.

This step is executed using the following evolution rules.

(Evolution rules for inner membranes)

$$\begin{aligned}
R_{2,4,1} \;=\;& \{\langle CheC_P\rangle\langle C_P,0\rangle \to \langle C_P,0\rangle\langle DelR\rangle\} \\
&\cup\{\langle CheC_P\rangle\langle C_P,i\rangle \to \langle C_P,i\rangle\langle ChoP\rangle \mid 1 \le i \le n\} \\
R_{2,4,2} \;=\;& \{\langle CheC_{PX}\rangle\langle C_P,0\rangle\langle C_X,0\rangle \to \langle MakC_R\rangle\} \\
&\cup\{\langle CheC_{PX}\rangle\langle C_P,i\rangle\langle C_X,j\rangle \to \langle C_P,i\rangle\langle C_X,j\rangle\langle CheC_P\rangle \mid 0 \le i \le n, 0 \le j \le n, 1 \le i+j\}
\end{aligned}$$

In the above set of evolution rules, object $\langle CheC_P\rangle$ executes a check of the number of elements in P according to evolution rules in $R_{2,4,1}$. If the number of elements in P is 0, then object $\langle DelR\rangle$ is generated for dissolution of the membrane. Otherwise, another trigger object, $\langle ChoP\rangle$ described in Step 2, is generated and the procedure returns to Step 2. Object $\langle CheC_{PX}\rangle$ executes a check of the numbers of elements in P and X according to evolution rules in $R_{2,4,2}$. When these numbers are both 0, object $\langle MakC_R\rangle$ is generated for counting the number of elements in R.

Step 5: Objects that representing clique size are sent out from the inner membrane that contains the maximal clique to the outer membrane. The other membranes, which contain no maximal clique, are dissolved.

This step is executed using the following evolution rules.

(Evolution rules for inner membranes)

$$\begin{aligned}
R_{2,5,1} \;=\;& \{\langle MakC_R\rangle\langle R_1,T\rangle \to \langle R_1,T\rangle\langle C_R,1\rangle\langle MakC_R,2\rangle\} \\
&\cup\{\langle MakC_R\rangle\langle R_1,F\rangle \to \langle R_1,F\rangle\langle C_R,0\rangle\langle MakC_R,2\rangle\} \\
&\cup\{\langle C_R,j\rangle\langle MakC_R,i\rangle\langle R_i,T\rangle \to \langle C_R,j+1\rangle\langle MakC_R,i+1\rangle\langle R_i,T\rangle \\
&\quad \mid 2 \le i \le n, 0 \le j \le n-1\}
\end{aligned}$$

$$\begin{aligned}
& \cup\{\langle C_R, j\rangle\langle MakC_R, i\rangle\langle R_i, F\rangle \to \langle C_R, j\rangle\langle MakC_R, i+1\rangle\langle R_i, F\rangle \mid 2 \le i \le n, 0 \le j \le n-1\} \\
& \cup\{\langle MakC_R, n+1\rangle \to \langle DupC_R\rangle\} \\
R_{2,5,2} \quad = \quad & \{\langle DupC_R\rangle\langle C_R, j\rangle \to \langle C_R, j\rangle\langle C_R, j\rangle\langle StBy\rangle\langle SenC_R\rangle \mid 1 \le j \le n\} \\
& \cup\{[\langle SenC_R\rangle\langle C_R, j\rangle\langle k, i\rangle]_2 \to []_2\langle Che, 2^i\rangle\langle{+}{+}C_R, j\rangle \mid 0 \le i \le n, 1 \le j \le n\} \\
R_{2,5,3} \quad = \quad & \{\langle DelR\rangle\langle R_1, W\rangle \to \langle DelR, 2\rangle \mid W \in \{T, F\}\} \\
& \cup\{\langle DelR, i\rangle\langle R_i, W\rangle \to \langle DelR, i+1\rangle \mid 2 \le i \le n, W \in \{T, F\}\} \\
& \cup\{\langle DelR, n+1\rangle \to \langle Dis\rangle\} \\
R_{2,5,4} \quad = \quad & \{[\langle Dis\rangle\langle k, i\rangle\langle e_{1,1}, F\rangle]_2 \to \langle Che, 2^i\rangle \mid 0 \le i \le n\}
\end{aligned}$$

$R_{2,5,1}$ and $R_{2,5,2}$ are executed in the membrane that contains the maximal clique. First, the number of elements in R is counted using loop counter object $\langle MakC_R\rangle$ and $\langle MakC_R, i\rangle$ according to evolution rules in $R_{2,5,1}$. Then, the counter object $\langle C_R, j\rangle$ is duplicated according to evolution rules in $R_{2,5,2}$. One of the duplicated objects is sent to the outer membrane with a object $\langle Che, 2^i\rangle$ and the other is left in the membrane. Object $\langle Che, 2^i\rangle$ is used in Step 6 to check whether all inner membranes has finished up to Step 5.

$R_{2,5,3}$ and $R_{2,5,4}$ are executed in the membrane that does not contain a maximal clique. Set R is deleted by the object $\langle DelR\rangle$, otherwise set R from various membranes is send to the outer membrane, which causes the procedure to be stopped. Then, object $\langle e_{1,1}, F\rangle$ is deleted for the same reason and the inner membranes are dissolved using object $\langle Dis\rangle$ according to evolution rules in $R_{2,5,4}$.

Step 6: After the results are sent out from all the inner membranes, objects that represent the maximum clique are sent out from the outer membrane.

This step is executed using the following evolution rules.

(Evolution rules for outer membrane)

$$\begin{aligned}
R_{1,6,1} \quad = \quad & \{\langle{+}{+}C_R, j\rangle\langle C_R, j, i\rangle \to \langle C_R, j, i+1\rangle \mid 0 \le i \le n-1, 2 \le j \le n\} \\
& \cup\{\langle Che, 2^i\rangle\langle Che, 2^i\rangle \to \langle Che, 2^{i+1}\rangle \mid 0 \le i \le n-1\} \\
& \cup\{\langle Che, 2^n\rangle \to \langle StaRes\rangle\} \\
R_{1,6,2} \quad = \quad & \{\langle StaRes\rangle \to \langle CheMax, n\rangle\} \\
& \cup\{\langle CheMax, j\rangle\langle C_R, j, i\rangle \to \langle MaxC_R, j\rangle \mid 1 \le i \le n, 2 \le j \le n\} \\
& \cup\{\langle CheMax, j\rangle\langle C_R, j, 0\rangle \to \langle CheMax, j-1\rangle \mid 2 \le j \le n\} \\
& \cup\{[]_2\langle MaxC_R, j\rangle \to [\langle MaxC_R, j\rangle]_2 \mid 2 \le j \le n\} \\
R_{1,6,3} \quad = \quad & \{\langle SenCli\rangle\langle R_1, W\rangle \to \langle SenR_1, W\rangle\langle SenCli, 2\rangle \mid W \in T, F\} \\
& \cup\{\langle SenCli, i\rangle\langle R_i, W\rangle \to \langle SenR_i, W\rangle\langle SenCli, i+1\rangle \mid 2 \le i \le n, W \in T, F\} \\
& \cup\{[\langle SenR_i, W\rangle]_1 \to []_1\langle R_i, W\rangle \mid 2 \le i \le n, W \in T, F\} \\
& \cup\{\langle SenCli, n+1\rangle \to \langle END\rangle\}
\end{aligned}$$

(Evolution rules for inner membranes)

$$\begin{aligned}
R_{2,6,1} \quad = \quad & \{\langle StBy\rangle\langle MaxC_R, j\rangle\langle C_R, i\rangle \to \langle SenCli\rangle \mid 1 \le i \le n, 2 \le j \le n, i = j\} \\
& \cup\langle StBy\rangle\langle MaxC_R, j\rangle\langle C_R, i\rangle \to \langle StBy\rangle\langle C_R, i\rangle\langle SenMaxC_R, j\rangle \mid 1 \le i \le n, 2 \le j \le n, i \ne j\} \\
& \cup\{[\langle SenCli\rangle\langle e_{1,1}, F\rangle]_2 \to \langle SenCli\rangle\} \\
& \cup\{[\langle SenMaxC_R, j\rangle]_2 \to []_2\langle MaxC_R, j\rangle \mid 2 \le j \le n\}
\end{aligned}$$

$R_{1,6,1}$ is executed to total the object submitted by each inner membrane. $\langle C_R, j, i\rangle$ of i are incremented and the addition of $\langle Che, 2^i\rangle$ is performed aiming at $\langle Che, 2^n\rangle$, which means that all inner membranes finish their execution up to Step 5.

The size of the maximum clique is checked using object $\langle CheMax, j\rangle$ according to evolution rules in $R_{1,6,2}$. Since the outer membrane do not know which number is the largest, $\langle C_R, j, i\rangle$ is checked from the case j is n. If i of the object $\langle C_R, j, i\rangle$ is not 0, then a maximal clique of size j exists in the membrane, and the maximum clique size is determined uniquely.

To execute the check, object $\langle MaxC_R, j\rangle$ is sent into a random inner membrane according to evolution rules in $R_{1,6,2}$. In the randomly selected inner membrane, the clique size in the membrane is verified. If the clique size in the membrane is not equal to the determined clique size, then object $\langle MaxC_R, j\rangle$ is sent out to the outer membrane to redo selection using object $\langle SenMaxC_R, j\rangle$ according to evolution rules in $R_{2,6,1}$. Otherwise, the inner membrane is dissolved using objects $\langle SenCli\rangle$ and $\langle e_{1,1}, F\rangle$; that is, the set R representing maximum clique is sent out to the outer membrane according to evolution rules in $R_{2,6,1}$. Set R represents the maximum clique and the result is sent out using objects $\langle SenCli, i\rangle$ and $\langle SenR_i, W\rangle$ according to evolution rules in $R_{1,6,3}$.

We now summarizes the asynchronous P system $\Pi_{\text{BK-basic}}$.

$$\Pi_{\text{BK-basic}} = (O_{meaning}, O_{state}, \mu, \omega_1, \omega_2, R_1, R_2, i_{in}, i_{out})$$

$$\begin{aligned}
O_{meaning} &= \{\langle e_{i,j}, W\rangle \mid 1 \le i \le n, 1 \le j \le n, W \in \{T,F\}\}\\
&\quad \cup\{\langle R_i, W\rangle\langle P_i, W\rangle\langle X_i, W\rangle\langle C_R, i, j\rangle \mid 1 \le i \le n, , 0 \le j \le n, W \in \{T,F\}\}\\
&\quad \cup\{\langle C_R, i\rangle\langle C_P, i\rangle\langle C_X, i\rangle \mid 0 \le i \le n\}\\
&\quad \cup\{\langle k, i\rangle \mid 0 \le i \le n\} \cup \{\langle Set, i\rangle \mid 0 \le i \le n\}\\
&\quad \cup\{\langle CHECK, 2^i\rangle \mid 0 \le i \le n\} \cup \{\langle maxC_R, i\rangle \mid 0 \le i \le n\}\\
O_{state} &= \{\langle M_{i,j}\rangle \mid 1 \le i \le n, 1 \le j \le m\}\\
&\quad \cup\{\langle MakeR\rangle\langle R_i\rangle\langle MakeP\rangle\langle P_i\rangle\langle MakeX\rangle\langle X_i\rangle \mid 1 \le i \le n\} \cup \{\langle MakeCounter\rangle\}\\
&\quad \cup\{\langle ChooseP\rangle\langle ChooseP, i\rangle \mid 1 \le i \le n\}\\
&\quad \cup\{\langle Div\rangle\langle NoDiv\rangle\} \cup \{\langle SV\rangle\langle NoSV\rangle\}\\
&\quad \cup\{\langle AlterR\rangle\langle AlterP\rangle\langle AlterP, i\rangle\langle AlterX\rangle\langle AlterX, i\rangle \mid 1 \le i \le n\}\\
&\quad \cup\{\langle CheckCountPX\rangle\} \cup \{\langle CheckCountP\rangle\langle - \, CountP\rangle\}\\
&\quad \cup\{\langle MakeCountR\rangle\langle makeC_R, i\rangle\langle DuplicateC_R\rangle \mid 1 \le i \le n\}\\
&\quad \cup\{\langle Standby\rangle \cup \{\langle DeleteR\rangle\} \cup \{\langle Dissolution\rangle\} \cup \{\langle StaRes\rangle\}\\
&\quad \cup\{\langle {+}{+}C_R, i\rangle \mid 1 \le i \le n\} \cup \{\langle maxcheck, i\rangle \mid 1 \le i \le n\}\\
&\quad \cup\{\langle sendbackmaxC_R, i\rangle \mid 1 \le i \le n\}\\
&\quad \cup\{\langle SendClique\rangle\langle sendR_i, W\rangle\langle sendresult, i\rangle \mid 1 \le i \le n, W \in \{T,F\}\}\\
&\quad \cup\{\langle END\rangle\}\\
\mu &= [\,[\,]_2\,]_1\\
\omega_1 &= \omega_2 \;=\; \phi\\
R_1 &= R_{1,1} \cup R_{1,6,1} \cup R_{1,6,2} \cup R_{1,6,3}\\
R_2 &= R_{2,1,1} \cup R_{2,1,2} \cup R_{2,1,3} \cup R_{2,1,4} \cup R_{2,1,5} \cup R_{2,2,1} \cup R_{2,2,2} \cup R_{2,3,1} \cup R_{2,3,2}\\
&\quad \cup R_{2,3,3} \cup R_{2,4,1} \cup R_{2,4,2} \cup R_{2,5,1} \cup R_{2,5,2} \cup R_{2,5,3} \cup R_{2,5,4} \cup R_{2,6,1}\\
i_{in} &= i_{out} = 1
\end{aligned}$$

3.4 Details of P system with the pivoting Bron-Kerbosch algorithm

We next explain a P system based on a pivoting Bron-Kerbosch algorithm. Since only Step 1 and Step 2 mainly differ from the P system with $\Pi_{\text{BK-basic}}$, we explain these two steps in detail below.

Step 1: A set of input is moved into an inner membrane and objects for the sets R, P, X, and Piv are generated.

This step is executed using the following additional evolution rules. ($R_{2,1,4}$ and $R_{2,1,5}$ are replaced with $R'_{2,1,4}$ and $R'_{2,1,5}$ from $\Pi_{\text{BK-basic}}$.)

(Evolution rules for inner membranes)

$$\begin{aligned}
R'_{2,1,4} &= \{\langle MakX\rangle \to \langle X_1, F\rangle\langle X_2\rangle\}\\
&\quad \cup\{\langle X_i\rangle \to \langle X_i, F\rangle\langle X_{i+1}\rangle \mid 1 \le i \le n\}\\
&\quad \cup\{\langle X_{n+1}\rangle \to \langle MakPiv\rangle\}\\
R_{2,1,6} &= \{\langle MakPiv\rangle \to \langle Piv_1, T\rangle\langle u, 1\rangle\langle Piv, 2\rangle\}
\end{aligned}$$

$$\begin{aligned}
& \cup\{\langle Piv, i\rangle \rightarrow \langle Piv_i, T\rangle\langle Piv, i+1\rangle \mid 2 \leq i \leq n\} \\
& \cup\{\langle Piv, n+1\rangle \rightarrow \langle MakC\rangle\} \\
R'_{2,1,5} \quad = \quad & \{\langle MakC\rangle \rightarrow \langle C_P, n\rangle\langle C_X, 0\rangle\langle k, n\rangle\langle ChoP\rangle\langle C_{Piv}, n\rangle\langle ChoPiv\rangle\}
\end{aligned}$$

After the objects representing R, P, and X are generated, objects $\langle u, 1\rangle$ and $\langle Piv_i, T\rangle$ are generated according to evolution rules in $R_{2,1,6}$. Object $\langle u, 1\rangle$ indicates that the first vertex is a pivot, and object $\langle Piv_i, T\rangle$ represents an element of the set $P - N(u)$, which is used in Step 2.

Step 2: In the inner membrane, a pivot vertex u is chosen from P or X. Then, a vertex from a set $Piv = P - N(u)$ is chosen. The inner membrane is divided into two membranes depending on whether the vertex is included in the maximal clique.

This step is executed using the following additional evolution rules. ($R_{2,2,1}$ is replaced with $R'_{2,2,1}$.)

(Evolution rules for inner membranes)

$$\begin{aligned}
R_{2,2,3} \quad = \quad & \{\langle ChoPiv\rangle\langle P_1, T\rangle\langle u, i\rangle \rightarrow \langle P_1, T\rangle\langle u, 1\rangle\langle UpdPiv\rangle \mid 1 \leq i \leq n\} \\
& \cup\{\langle ChoPiv\rangle\langle P_1, F\rangle \rightarrow \langle P_1, F\rangle\langle ChoPiv, 2\rangle \\
& \cup\{\langle ChoPiv, j\rangle\langle P_j, T\rangle\langle u, i\rangle \rightarrow \langle P_j, T\rangle\langle u, i\rangle\langle UpdPiv\rangle \mid 1 \leq i \leq n, 2 \leq j \leq n\} \\
& \cup\{\langle ChoPiv, j\rangle\langle P_j, F\rangle \rightarrow \langle P_j, F\rangle\langle ChoPiv, j+1\rangle \mid 2 \leq j \leq n\} \\
& \cup\{\langle ChoPiv, n+1\rangle \rightarrow \langle ChoPivX, 1\rangle\} \\
& \cup\{\langle ChoPivX, j\rangle\langle X_j, T\rangle\langle u, i\rangle \rightarrow \langle X_j, T\rangle\langle u, i\rangle\langle UpdPiv\rangle \mid 1 \leq i \leq n, 1 \leq j \leq n\} \\
& \cup\{\langle ChoPivX, j\rangle\langle X_j, F\rangle \rightarrow \langle X_j, F\rangle\langle ChoPivX, j+1\rangle \mid 1 \leq j \leq n\} \\
R_{2,2,4} \quad = \quad & \{\langle UpdPiv\rangle\langle P_1, T\rangle\langle u, i\rangle\langle e_{i,1}, F\rangle\langle Piv_1, W\rangle \rightarrow \langle P_1, T\rangle\langle u, i\rangle\langle e_{i,1}, F\rangle\langle Piv_1, T\rangle\langle UpdPiv, 2\rangle \\
& \quad \mid 1 \leq i \leq n, W \in \{T, F\}\} \\
& \cup\{\langle UpdPiv\rangle\langle P_1, W'\rangle\langle u, i\rangle\langle e_{i,1}, W''\rangle\langle Piv_1, W\rangle \rightarrow \langle P_1, T\rangle\langle u, i\rangle\langle e_{i,1}, W''\rangle\langle Piv_1, F\rangle\langle UpdPiv, 2\rangle \\
& \quad \mid 1 \leq i \leq n, W, W', W'' \in \{T, F\}, (W', W'') \neq (T, F)\} \\
& \cup\{\langle UpdPiv, j\rangle\langle P_j, T\rangle\langle u, i\rangle\langle e_{i,j}, F\rangle\langle Piv_j, W\rangle \rightarrow \langle P_j, T\rangle\langle u, i\rangle\langle e_{i,j}, F\rangle\langle Piv_j, T\rangle\langle UpdPiv, j+1\rangle \\
& \quad \mid 1 \leq i \leq n, 2 \leq j \leq n, W \in \{T, F\}\} \\
& \cup\{\langle UpdPiv, j\rangle\langle P_j, W'\rangle\langle u, i\rangle\langle e_{i,j}, W''\rangle\langle Piv_j, W\rangle \rightarrow \langle P_j, T\rangle\langle u, i\rangle\langle e_{i,j}, W''\rangle\langle Piv_j, F\rangle\langle UpdPiv, j+1\rangle \\
& \quad \mid 1 \leq i \leq n, 2 \leq j \leq n, W, W', W'' \in \{T, F\}, (W', W'') \neq (T, F)\} \\
& \cup\{\langle UpdPiv, n+1\rangle \rightarrow \langle UpdC_{Piv}\rangle \\
R_{2,2,5} \quad = \quad & \{\langle UpdC_{Piv}\rangle\langle Piv_1, T\rangle\langle C_{Piv}, i\rangle \rightarrow \langle Piv_1, T\rangle\langle C_{Piv}, 1\rangle\langle UpdC_{Piv}, 2\rangle \mid 1 \leq i \leq n\} \\
& \cup\{\langle UpdC_{Piv}\rangle\langle Piv_1, F\rangle\langle C_{Piv}, i\rangle \rightarrow \langle Piv_1, F\rangle\langle C_{Piv}, 0\rangle\langle UpdC_{Piv}, 2\rangle \mid 0 \leq i \leq n\} \\
& \cup\{\langle UpdC_{Piv}, j\rangle\langle Piv_j, T\rangle\langle C_{Piv}, i\rangle \rightarrow \langle Piv_j, T\rangle\langle C_{Piv}, i+1\rangle\langle UpdC_{Piv}, j+1\rangle \\
& \quad \mid 0 \leq i \leq n, 2 \leq j \leq n\} \\
& \cup\{\langle UpdC_{Piv}, j\rangle\langle Piv_j, F\rangle\langle C_{Piv}, i\rangle \rightarrow \langle Piv_j, F\rangle\langle C_{Piv}, i\rangle\langle UpdC_{Piv}, j+1\rangle \mid 0 \leq i \leq n, 2 \leq j \leq n\} \\
& \cup\{\langle UpdC_{Piv}, n+1\rangle \rightarrow \langle CheC_{Piv}\rangle\} \\
R'_{2,2,1} \quad = \quad & \{\langle ChoP\rangle\langle Piv_1, T\rangle\langle C_{Piv}, j\rangle \rightarrow \langle Piv_1, F\rangle\langle Set, 1\rangle\langle C_{Piv}, j-1\rangle\langle Div\rangle \mid 2 \leq j \leq n\} \\
& \cup\{\langle ChoP\rangle\langle Piv_1, T\rangle\langle C_{Piv}, 1\rangle \rightarrow \langle Piv_1, F\rangle\langle Set, 1\rangle\langle C_{Piv}, 0\rangle\langle NoDiv\rangle\} \\
& \cup\{\langle ChoP\rangle\langle Piv_1, F\rangle \rightarrow \langle Piv_1, F\rangle\langle ChoP, 2\rangle\} \\
& \cup\{\langle ChoP, i\rangle\langle Piv_i, T\rangle\langle C_{Piv}, j\rangle \rightarrow \langle Piv_i, F\rangle\langle Set, i\rangle\langle C_{Piv}, j-1\rangle\langle Div\rangle \mid 2 \leq i \leq n, 2 \leq j \leq n\} \\
& \cup\{\langle ChoP, i\rangle\langle Piv_i, T\rangle\langle C_{Piv}, 1\rangle \rightarrow \langle Piv_i, F\rangle\langle Set, i\rangle\langle C_{Piv}, 0\rangle\langle NoDiv\rangle \mid 2 \leq i \leq n\} \\
& \cup\{\langle ChoP, i\rangle\langle Piv_i, F\rangle \rightarrow \langle Piv_i, F\rangle\langle ChoP, i+1\rangle\} \mid 2 \leq i \leq n\}
\end{aligned}$$

A pivot vertex is selected from P using loop counter object $\langle ChoPiv, i\rangle$ according to evolution rules in $R_{2,2,3}$. When object $\langle P_j, T\rangle$ is discovered, the vertex is selected as a pivot. Then, set Piv is updated according to evolution rules in $R_{2,2,4}$. Since set Piv equals $P - N(u)$, the adjacent vertices of u are removed from P. In other words, object $\langle P_j, T\rangle$ remains $\langle P_j, T\rangle$ only when P_j is TRUE and object $e_{i,j}$ is FALSE. Object C_{Piv} is also updated according to the update of Piv according to evolution rules in $R_{2,2,5}$.

A vertex to add to the solution is selected from Piv using object $\langle ChoP, i\rangle$ according to evolution rules in $R'_{2,2,1}$. If object C_{Piv} indicates 1, then the membrane is not divided because the remaining vertex is in the maximal clique.

In addition to the above modifications for Step 1 and Step 2, the following minor modifications are applied to Step 3 and Step 4. (The following sets of evolution rules are replacements of the evolution rules for Step 3 and Step 4.)

Step 3

(Evolution rules for inner membranes)

$$\begin{aligned} R'_{2,3,3} &= \{\langle NoSV\rangle\langle Set,i\rangle\langle P_i,T\rangle\langle X_i,F\rangle \to \langle P_i,F\rangle\langle X_i,T\rangle\langle - C_P\rangle\langle ++\ C_X\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle - C_P\rangle\langle ++C_X\rangle\langle C_P,i\rangle\langle C_X,j\rangle \to \langle C_P,i-1\rangle\langle C_X,j+1\rangle\langle CheC_{Piv}\rangle \mid 1 \le i \le n, 0 \le j \le n-1\} \end{aligned}$$

Step 4

(Evolution rules for inner membranes)

$$\begin{aligned} R'_{2,4,1} &= \{\langle CheC_{Piv}\rangle\langle C_{Piv},0\rangle \to \langle C_{Piv},0\rangle\langle DelR\rangle\} \\ &\quad \cup\{\langle CheC_{Piv}\rangle\langle C_{Piv},i\rangle \to \langle C_{Piv},i\rangle\langle ChoP\rangle \mid 1 \le i \le n\} \\ R'_{2,4,2} &= \{\langle CheC_{PX}\rangle\langle C_P,0\rangle\langle C_X,0\rangle \to \langle MakC_R\rangle\} \\ &\quad \cup\{\langle CheC_{PX}\rangle\langle C_P,i\rangle\langle C_X,j\rangle \to \langle C_P,i\rangle\langle C_X,j\rangle\langle ChoPiv\rangle \mid 0 \le i \le n, 0 \le j \le n, 1 \le i+j\} \end{aligned}$$

We now summarize the asynchronous P system $\Pi_{\text{BK-pivot}}$.

$$\Pi_{\text{BK-pivot}} = (O_{meaning}, O_{state}, \mu, \omega_1, \omega_2, R_1, R_2, i_{in}, i_{out})$$

$$\begin{aligned} O_{meaning} &= \{\langle e_{i,j},W\rangle \mid 1 \le i \le n, 1 \le j \le n, W \in \{T,F\}\} \\ &\quad \cup\{\langle R_i,W\rangle\langle P_i,W\rangle\langle X_i,W\rangle\langle C_R,i,j\rangle \mid 1 \le i \le n, , 0 \le j \le n, W \in \{T,F\}\} \\ &\quad \cup\{\langle C_R,i\rangle\langle C_P,i\rangle\langle C_X,i\rangle \mid 0 \le i \le n\} \\ &\quad \cup\{\langle k,i\rangle \mid 0 \le i \le n\} \cup \{\langle Set,i\rangle \mid 0 \le i \le n\} \\ &\quad \cup\{\langle CHECK,2^i\rangle \mid 0 \le i \le n\} \cup \{\langle maxC_R,i\rangle \mid 0 \le i \le n\} \\ O_{state} &= \{\langle M_{i,j}\rangle \mid 1 \le i \le n, 1 \le j \le m\} \\ &\quad \cup\{\langle MakeR\rangle\langle R_i\rangle\langle MakeP\rangle\langle P_i\rangle\langle MakeX\rangle\langle X_i\rangle \mid 1 \le i \le n\} \cup \{\langle MakeCounter\rangle\} \\ &\quad \cup\{\langle ChooseP\rangle\langle ChooseP,i\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle Div\rangle\langle NoDiv\rangle\} \cup \{\langle SV\rangle\langle NoSV\rangle\} \\ &\quad \cup\{\langle AlterR\rangle\langle AlterP\rangle\langle AlterP,i\rangle\langle AlterX\rangle\langle AlterX,i\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle CheckCountPX\rangle\} \cup \{\langle CheckCountPivot\rangle\langle - CountP{+}{+}CountX\rangle\} \\ &\quad \cup\{\langle MakeCountR\rangle\langle makeC_R,i\rangle\langle DuplicateC_R\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle Standby\rangle \cup \{\langle DeleteR\rangle\} \cup \{\langle Dissolution\rangle\} \cup \{\langle StaRes\rangle\} \\ &\quad \cup\{\langle ++C_R,i\rangle \mid 1 \le i \le n\} \cup \{\langle maxcheck,i\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle sendbackmaxC_R,i\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle SendClique\rangle\langle sendR_i,W\rangle\langle sendresult,i\rangle \mid 1 \le i \le n, W \in \{T,F\}\} \\ &\quad \cup\{\langle UpdatePivot\rangle\langle updatePivot,i\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle UpdateCountPivot\rangle\langle updatecountpivot,i\rangle \mid 1 \le i \le n\} \\ &\quad \cup\{\langle END\rangle\} \\ \mu &= [\ [\ ]_2\ ]_1 \\ \omega_1 &= \omega_2 = \phi \\ R_1 &= R_{1,1} \cup R_{1,6,1} \cup R_{1,6,2} \cup R_{1,6,3} \\ R_2 &= R_{2,1,1} \cup R_{2,1,2} \cup R_{2,1,3} \cup R'_{2,1,4} \cup R'_{2,1,5} \cup R_{2,1,6} \cup R'_{2,2,1} \cup R_{2,2,2} \end{aligned}$$

$$
\begin{aligned}
& \cup R_{2,2,3} \cup R_{2,2,4} \cup R_{2,2,5} \cup R_{2,3,1} \cup R_{2,3,2} \cup R'_{2,3,3} \cup R'_{2,4,1} \cup R'_{2,4,2} \\
& \cup R_{2,5,1} \cup R_{2,5,2} \cup R_{2,5,3} \cup R_{2,5,4} \cup R_{2,6,1} \\
i_{in} &= i_{out} = 1
\end{aligned}
$$

3.5 Examples of executions of the proposed P systems

We show examples of executions of $\Pi_{\text{BK-basic}}$ and $\Pi_{\text{BK-pivot}}$. First, we show an example for $\Pi_{\text{BK-basic}}$. The behaviors for an input graph in Figure 1 are shown in Figure 2 to Figure 3.

Figure 2 (a) shows the initial state. The input object set O_E is given in membrane 1. Figure 2 (b) shows the state at the end of Step 1. In this step, O_E is moved into the membrane 2 and O_R, O_P and O_X are created. Figure 2 (c) shows the state at the end of Step 2. In this step, a vertex in P is selected and the membrane division is executed.

Figure 2 (d) shows the state at the end of Step 3. In this step, vertex 1 is added to R, and the set P and the counter of P are updated according to the number of P in the left membrane. In addition, vertex 1 is added to X in the right membrane. Figure 2 (e) shows the state at the end of Step 4. In this case, the value of counter of P are not zero, and both inner membranes generate $\langle ChoP \rangle$, which is a trigger object to Step 2. After that, each membrane execute from Step 2 through Step 4 until the number of P become zero.

Figure 3 (a) shows the case all membranes become the set P is empty. Next, the membranes execute from Step 4 through Step 5. Figure 3 (b) shows the state at the end of Step 5. In this step, the membrane that X is empty sends out counter of R. The membrane that X is not empty is dissolved after deleting the set R. Figure 3 (c) shows the state at the end of Step 6. In this step, the membrane with maximum clique are dissolved with O_R and the result are sent out from the the outer membrane.

Second, we show the example with pivoting method. The behavior given the graph in Figure 1 is shown in Figure 4. Figure 4 (a) shows the state at the end of Step 1. In this step, the object O_{Piv} and counter of pivot are added compared to $\Pi_{\text{BK-basic}}$. Figure 4 (b) shows the state at the end of Step 2. In this step, vertex 1 is selected as a pivot and O_{Piv} is updated to the pivot and the non-adjacent vertices of pivot. After that, the membrane execute Step 4 and backtrack to Step 2. Figure 4 (c) shows the state at the end of the second Step 2. In this step, the membrane division is executed after checking counter of pivot.

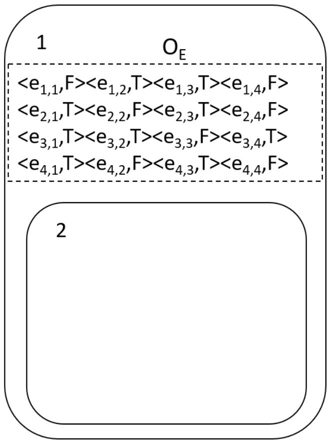
Figure 4 (d) shows the state at the end of Step 3. In this step, the execution is almost same as one with $\Pi_{\text{BK-basic}}$. Figure 4 (e) shows the state at the end of Step 4. In this step, trigger object that selects a new pivot is generated because O_P is not empty in the left membrane. Trigger object that selects a vertex to add the set R is generated because O_{Piv} is not empty. After that, each membrane executes from Step 2 through Step 4 until the number of P becomes zero.

Execution after Step 5 is the same as one with $\Pi_{\text{BK-basic}}$.

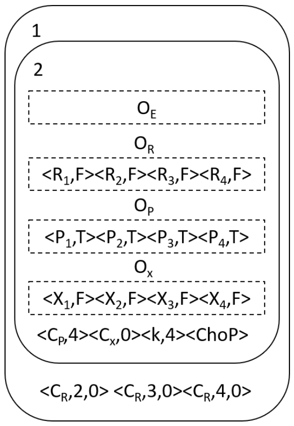
3.6 Complexity of P system

We now explain the complexities of P systems $\Pi_{\text{BK-basic}}$ and $\Pi_{\text{BK-pivot}}$. Since these two P systems mainly differ in Step 1 and Step 2, we explain the difference of the complexities in the two steps. First of all, the number of membranes is $O(2^n)$ because the inner membrane is divided in case that the vertex is included in the clique or not included in Step 2. Since 2^n instances are obtained by the division, the number of inner membranes is 2^n in the worst case.

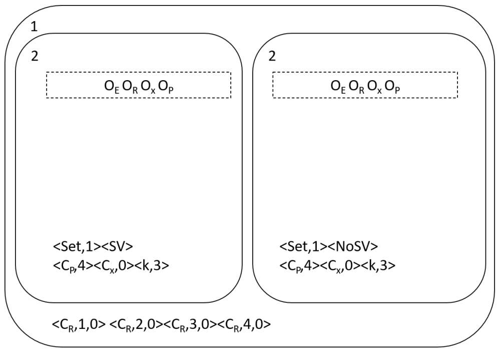
Next, we consider complexities for Steps 1 to 6. Step 1 of $\Pi_{\text{BK-basic}}$ is executed in $O(n^2)$ parallel steps or $O(n^2)$ sequential steps since $O(n^2)$ objects move sequentially. $O(n^2)$ kinds of objects and $O(n^2)$ kinds of evolution rules are used in Step 1. The complexity, kinds of objects and kinds of evolution rules of $\Pi_{\text{BK-pivot}}$ are the same because similar movement is executed in $\Pi_{\text{BK-pivot}}$. Step 2 of $\Pi_{\text{BK-basic}}$ is executed in $O(n)$ parallel steps or $O(n2^n)$ sequential steps since the set P is checked and the number of membranes is $O(2^n)$. $O(n^2)$ kinds of objects and $O(n^2)$ kinds of evolution rules are used in Step 2. Step 2 of $\Pi_{\text{BK-pivot}}$ is executed in $O(n)$ parallel steps or $O(n2^n)$ sequential steps since the set $Pivot$ is checked. $O(n^2)$ kinds of objects and $O(n^2)$ kinds of evolution rules are used in Step 2.


1
O_E
$\langle e_{1,1},F\rangle\langle e_{1,2},T\rangle\langle e_{1,3},T\rangle\langle e_{1,4},F\rangle$
$\langle e_{2,1},T\rangle\langle e_{2,2},F\rangle\langle e_{2,3},T\rangle\langle e_{2,4},F\rangle$
$\langle e_{3,1},T\rangle\langle e_{3,2},T\rangle\langle e_{3,3},F\rangle\langle e_{3,4},T\rangle$
$\langle e_{4,1},T\rangle\langle e_{4,2},F\rangle\langle e_{4,3},T\rangle\langle e_{4,4},F\rangle$
2

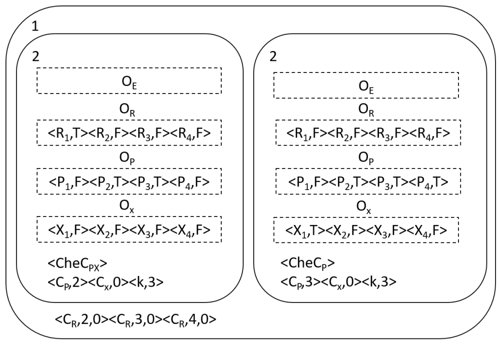
(a) Initial state


1
2
O_E
O_R
$\langle R_1,F\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,T\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,T\rangle$
O_x
$\langle X_1,F\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle C_P,4\rangle\langle C_x,0\rangle\langle k,4\rangle\langle ChoP\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

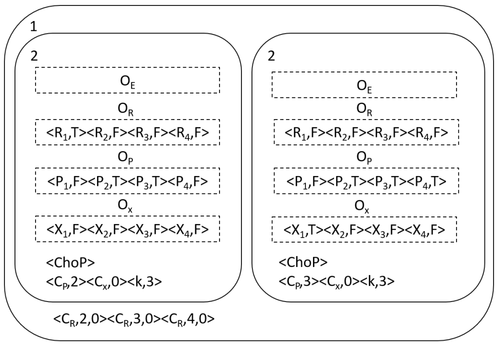
(b) Step 1


1
2
$O_E\ O_R\ O_x\ O_P$
$\langle Set,1\rangle\langle SV\rangle$
$\langle C_P,4\rangle\langle C_x,0\rangle\langle k,3\rangle$
2
$O_E\ O_R\ O_x\ O_P$
$\langle Set,1\rangle\langle NoSV\rangle$
$\langle C_P,4\rangle\langle C_x,0\rangle\langle k,3\rangle$
$\langle C_R,1,0\rangle\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

(c) Step 2

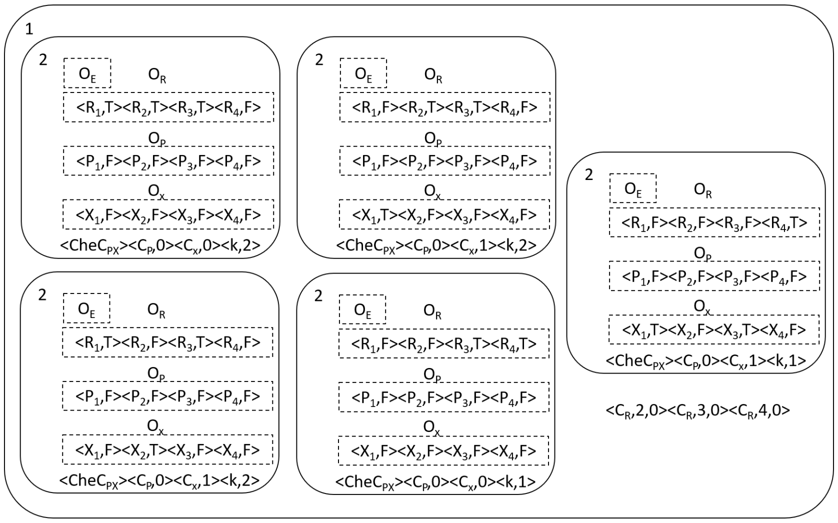

1
2
O_E
O_R
$\langle R_1,T\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,F\rangle$
O_x
$\langle X_1,F\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle CheC_{PX}\rangle$
$\langle C_P,2\rangle\langle C_x,0\rangle\langle k,3\rangle$
2
O_E
O_R
$\langle R_1,F\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,T\rangle$
O_x
$\langle X_1,T\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle CheC_P\rangle$
$\langle C_P,3\rangle\langle C_x,0\rangle\langle k,3\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

(d) Step 3

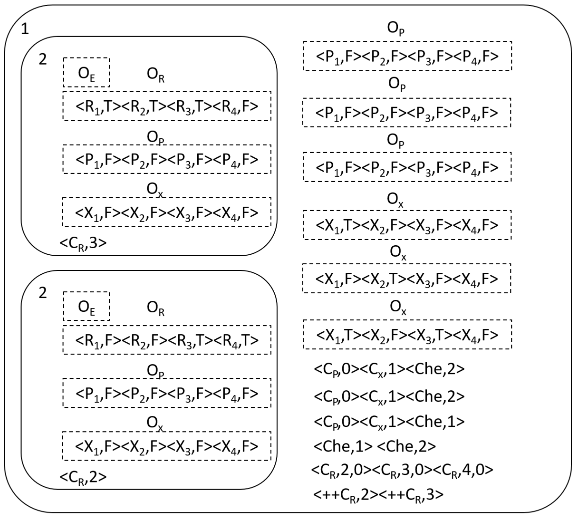

1
2
O_E
O_R
$\langle R_1,T\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,F\rangle$
O_x
$\langle X_1,F\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle ChoP\rangle$
$\langle C_P,2\rangle\langle C_x,0\rangle\langle k,3\rangle$
2
O_E
O_R
$\langle R_1,F\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,T\rangle$
O_x
$\langle X_1,T\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle ChoP\rangle$
$\langle C_P,3\rangle\langle C_x,0\rangle\langle k,3\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

(e) Step 4

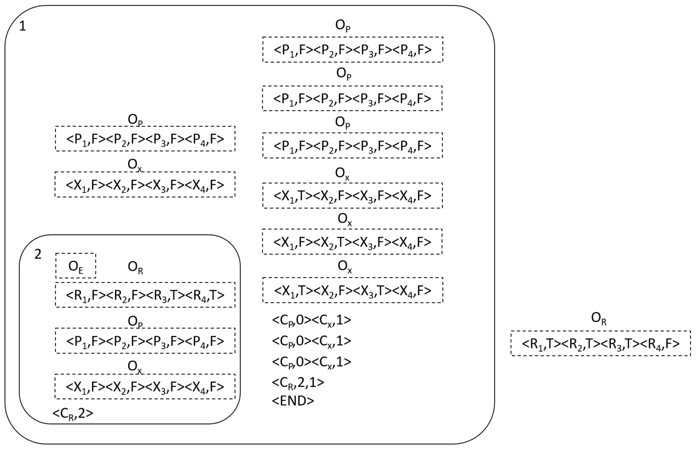
Figure 2: An example of execution of $\Pi_{\text{BK-basic}}$ (the first half)



(a) Step 3

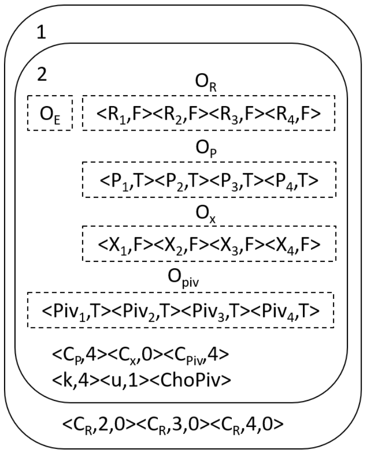


(b) Step 5

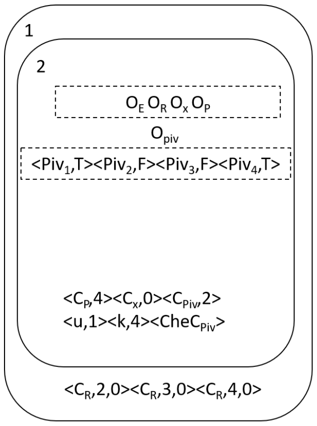


(c) Step 6

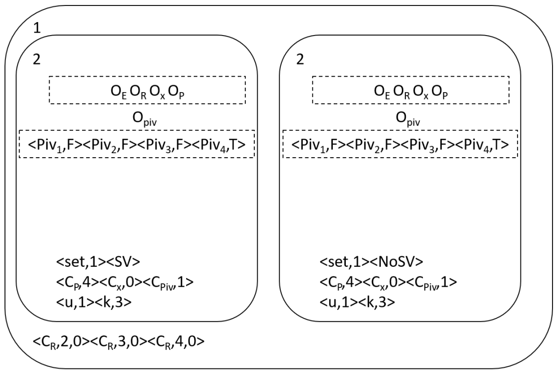
Figure 3: An example of execution of $\Pi_{\text{BK-basic}}$ (the second half)


1
2
O_R
O_E
$\langle R_1,F\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,T\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,T\rangle$
O_X
$\langle X_1,F\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
O_{piv}
$\langle Piv_1,T\rangle\langle Piv_2,T\rangle\langle Piv_3,T\rangle\langle Piv_4,T\rangle$
$\langle C_P,4\rangle\langle C_X,0\rangle\langle C_{Piv},4\rangle$
$\langle k,4\rangle\langle u,1\rangle\langle ChoPiv\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

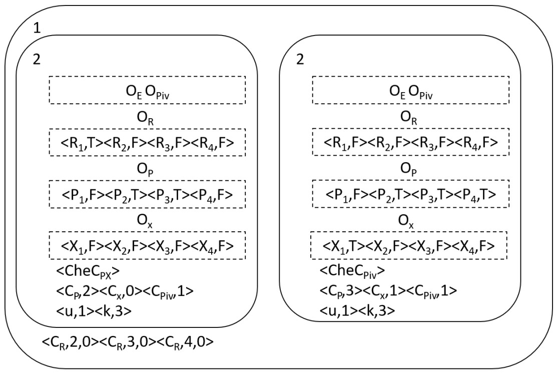
(a) Step 1


1
2
$O_E\,O_R\,O_X\,O_P$
O_{piv}
$\langle Piv_1,T\rangle\langle Piv_2,F\rangle\langle Piv_3,F\rangle\langle Piv_4,T\rangle$
$\langle C_P,4\rangle\langle C_X,0\rangle\langle C_{Piv},2\rangle$
$\langle u,1\rangle\langle k,4\rangle\langle CheC_{Piv}\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

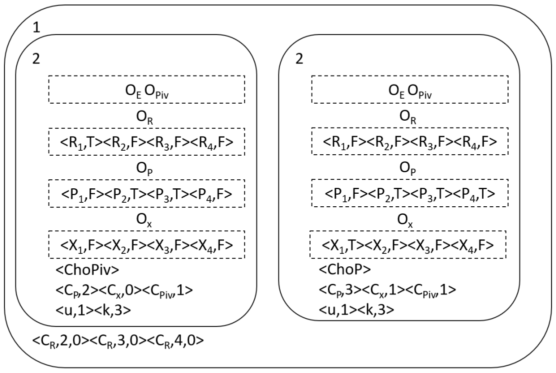
(b) Step 2


1
2
$O_E\,O_R\,O_X\,O_P$
O_{piv}
$\langle Piv_1,F\rangle\langle Piv_2,F\rangle\langle Piv_3,F\rangle\langle Piv_4,T\rangle$
$\langle set,1\rangle\langle SV\rangle$
$\langle C_P,4\rangle\langle C_X,0\rangle\langle C_{Piv},1\rangle$
$\langle u,1\rangle\langle k,3\rangle$
2
$O_E\,O_R\,O_X\,O_P$
O_{piv}
$\langle Piv_1,F\rangle\langle Piv_2,F\rangle\langle Piv_3,F\rangle\langle Piv_4,T\rangle$
$\langle set,1\rangle\langle NoSV\rangle$
$\langle C_P,4\rangle\langle C_X,0\rangle\langle C_{Piv},1\rangle$
$\langle u,1\rangle\langle k,3\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

(c) The second Step 2


1
2
$O_E\,O_{Piv}$
O_R
$\langle R_1,T\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,F\rangle$
O_X
$\langle X_1,F\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle CheC_{PX}\rangle$
$\langle C_P,2\rangle\langle C_X,0\rangle\langle C_{Piv},1\rangle$
$\langle u,1\rangle\langle k,3\rangle$
2
$O_E\,O_{Piv}$
O_R
$\langle R_1,F\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,T\rangle$
O_X
$\langle X_1,T\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle CheC_{Piv}\rangle$
$\langle C_P,3\rangle\langle C_X,1\rangle\langle C_{Piv},1\rangle$
$\langle u,1\rangle\langle k,3\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

(d) Step 3


1
2
$O_E\,O_{Piv}$
O_R
$\langle R_1,T\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,F\rangle$
O_X
$\langle X_1,F\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle ChoPiv\rangle$
$\langle C_P,2\rangle\langle C_X,0\rangle\langle C_{Piv},1\rangle$
$\langle u,1\rangle\langle k,3\rangle$
2
$O_E\,O_{Piv}$
O_R
$\langle R_1,F\rangle\langle R_2,F\rangle\langle R_3,F\rangle\langle R_4,F\rangle$
O_P
$\langle P_1,F\rangle\langle P_2,T\rangle\langle P_3,T\rangle\langle P_4,T\rangle$
O_X
$\langle X_1,T\rangle\langle X_2,F\rangle\langle X_3,F\rangle\langle X_4,F\rangle$
$\langle ChoP\rangle$
$\langle C_P,3\rangle\langle C_X,1\rangle\langle C_{Piv},1\rangle$
$\langle u,1\rangle\langle k,3\rangle$
$\langle C_R,2,0\rangle\langle C_R,3,0\rangle\langle C_R,4,0\rangle$

(e) Step 4

Figure 4: An example of execution of $\Pi_{\text{BK-pivot}}$

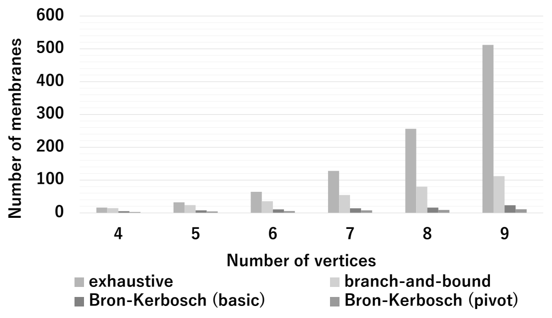


Figure 5: Number of membranes for different methods

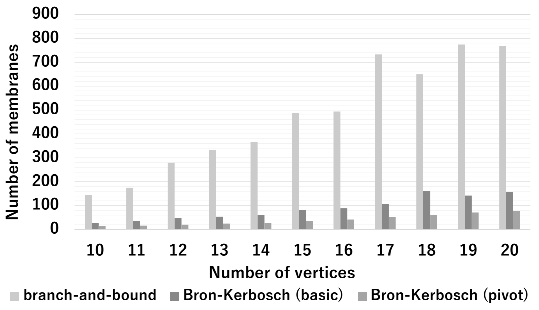


Figure 6: Number of membranes for larger numbers of vertices

Step 3 is executed in $O(n)$ parallel steps or $O(n2^n)$ sequential steps since the set R and P are updated in Step 3. $O(n^2)$ kinds of objects and $O(n^3)$ kinds of evolution rules are used in Step 3 and the number of membranes is $O(2^n)$. Step 4 is executed in $O(1)$ parallel steps or $O(2^n)$ sequential steps since the counter is checked in Step 4 and the number of membranes is $O(2^n)$. $O(n)$ kinds of objects and $O(n^2)$ kinds of evolution rules are used in Step 4. Steps 2 through 4 are repeated at worst n times.

Step 5 is executed in $O(n)$ parallel steps or $O(n2^n)$ sequential steps since the set is checked in Step 5 and the number of membranes is $O(2^n)$. $O(n)$ kinds of objects and $O(n^2)$ kinds of evolution rules are used in Step 5. Step 6 is executed in $O(n)$ parallel steps or $O(n)$ sequential steps since the set R is moved in Step 6. $O(n^2)$ kinds of objects and $O(n^2)$ kinds of evolution rules are used in Step 6.

From the above, we obtain the following theorem regarding the complexity of the proposed two asynchronous P systems, $\Pi_{\text{BK-basic}}$ and $\Pi_{\text{BK-pivot}}$.

Theorem 1 *Two asynchronous P systems, $\Pi_{BK\text{-}basic}$ and $\Pi_{BK\text{-}pivot}$, solve the maximum clique problem with n vertices and operate in $O(n^2)$ parallel steps or $O(n^2 2^n)$ sequential steps using $O(n^2)$ types of objects, $O(n^3)$ kinds of evolution rules and $O(2^n)$ membranes.* □

4 Experimental simulation

For evaluating the validity of the proposed P systems, we use our original simulator for the asynchronous P system. The simulator is built using Python 3 and is executed on CentOS 7. The input formula is randomly created for a given number of n vertices.

We compare number of membranes between the proposed P system and two existing P systems [9, 11]. A P system in [9] executes an exhaustive search for finding the maximum independent set, and a P system in [11] executes the same exhaustive search with branch and bound. Although the two P systems were proposed for solving the maximum independent set problem, the maximal clique can be reduced to the maximum independent set of the complementary graph, and we can compare number of membranes using this reduction.

The first simulation is executed for a small number of vertices. Figure 5 shows the average numbers of membranes with 10 trials for the first simulation. As shown, the numbers of membranes obtained on the P system in [9] increase exponentially, whereas the numbers of membranes obtained on the other P systems seem to be linear in the number of vertices.

Figure 6 shows the average numbers of membranes of the three P systems for larger numbers of vertices. Although the complexities of the P systems are the same, it is verified that there are differences between the numbers of membranes. (Note that the numbers are decreased for the number of membranes in $17 \leq n \leq 19$. We guess that the decreases are caused by the small number of trials.)

5 Conclusions

In the present paper, we proposed an asynchronous P system for solving the maximum clique problem using a Bron-Kerbosch algorithm. The results of simulations show that the number of membranes used in the proposed P system is significantly smaller than the numbers of membranes used in existing P systems.

In our future research, we intend to consider reducing the number of membranes for other computationally hard problems.

Acknowledgment

This research was supported in part by JSPS KAKENHI through a Grant-in-Aid for Scientific Research (C) (No. 20K11681).

References

[1] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[2] Y. Jimen and A. Fujiwara. Asynchronous P systems for solving the satisfiability problem. *International Journal of Networking and Computing*, 8(2):141–152, 2018.

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

[4] 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.

[5] Y. Nakano and A. Fujiwara. An asynchronous P system with branch and bound for solving the knapsack problem. In *Workshop on Parallel and Distributed Algorithms and Applications*, pages 242–248, 2020.

[6] 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.

[7] G. Păun. Computing with membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000.

[8] H. Tagawa and A. Fujiwara. Solving SAT and Hamiltonian cycle problem using asynchronous p systems. *IEICE Transactions on Information and Systems (Special section on Foundations of Computer Science)*, E95-D(3), 2012.

[9] K. Tanaka and A. Fujiwara. Asynchronous P systems for hard graph problems. *International Journal of Networking and Computing*, 4(1):2–22, 2014.

[10] K. Umetsu and A. Fujiwara. P systems with branch and bound for solving two hard graph problems. *International Journal of Networking and Computing*, 10(2):159–173, 2020.

[11] K. Umetsu and A. Fujiwara. An asynchronous p system using branch and bound for maximum independent set. *Bulletin of Networking, Computing, Systems, and Software*, 10(1):10–16, 2021.

[12] 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*, pages 289–301, 2000.