IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 13, 2019, accepted May 11, 2019, date of publication May 20, 2019, date of current version June 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917889

Solving the 0-1 Knapsack Problem by Using Tissue P System With Cell Division

LIAN YE[1], JINHANG ZHENG[1], PING GUO[1], AND MARIO J. PÉREZ-JIMÉNEZ[2]
[1]College of Computer Science, Chongqing University, Chongqing 440004, China
[2]Department of Computer Science and Artificial intelligence, University of Seville, 41012 Seville, Spain

Corresponding author: Lian Ye (ylredleaf@cqu.edu.cn)

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2019CDXYJSJ0021.

ABSTRACT Membrane computing is a kind of distributed and parallel computing model inspired by a biological cell mechanism. The maximum parallelism of membrane computing improves the computational efficiency of its computational model. In this paper, a tissue P system named Π_{KP} is proposed to solve the 0-1 knapsack problem, which is one of the classic NP-hard problems. Π_{KP} could obtain the accurate solutions of knapsack problem and points out the number of accurate solutions, which mainly consists of three stages: first, generate all the solutions of knapsack problem by a cell division; then calculate the weights and total values in all the candidate membranes, which will be kept or dissolved according to the restriction of knapsack problem; and check out the final solutions. The instances are executed on a membrane simulator named UPSimulator, and the result of the experiments shows the whole searching procedure by the rules and proves the correctness and efficiency of the system.

INDEX TERMS Tissue P system, 0-1 knapsack problem, membrane computing, combinatorial optimization.

I. INTRODUCTION

Membrane computing is a new branch of nature computation. It was proposed by Păun in 1998 [1]. It is inspired by the cellular structure and behavior patterns of biological cells [2]. Membrane computing, similar to quantum computing, is a new non-traditional computing model, which has been applied in linguistics, sociology, optimization design and many other fields [3]. P system consists three categories: cell-like P system, tissue-like P system and neural-like P system [4]. Tissue-like P systems that simulate the physiological mechanisms of biological tissues have their unique characteristics and prospects [5]. The tissue P system has the function of reverse transport, and the symbol objects processed during the transport process should be transmitted along the specified channel according to the symport/antiport rules [6].

Tissue P system has attracted many researchers and its computational ability has been recognized. The independence of intermembrane operations helps to simplify complex problems such as NP problem. A tissue P system with cell division was proposed and the computing capability of this variant has been proved by solving the SAT problem in polynomial time in [7]. The subset sum was solved in linear time by using tissue P system with cell division [8]. Tissue P system was used to get a fast solution of partition problem in 2008 [9]. A general schema was proved by solving 3-coloring problem in [10] and [11]. In [12] an efficient solution of the 3-coloring problem was presented. The independent set problem was solved in linear time in [13]. A uniform family of tissue P systems was constructed to solve the partition problem and the computational power of tissue P systems with cell separation is investigated in [14]. A uniform solution for vertex cover problem was proposed by using time-free tissue P systems [15]. The bin-packing problem was solved by means of tissue P system with 2-division [16].

Knapsack problem is a typical problem of combined optimization [17] and plays an important role in the actual problems of cargo loading, demand prediction and material cutting [18]. The algorithm for solving the knapsack problem is divided into two main categories: the accurate algorithm and the approximate algorithm [19]. The accurate algorithm includes enumeration algorithm, dynamic programming algorithm and so on. The approximate algorithm contains ant colony algorithm [20], genetic algorithm [21], differential evolution algorithm [22]. Traditional algorithms

The associate editor coordinating the review of this manuscript and approving it for publication was Yu-Huei Cheng.

mentioned above have their own shortcomings such as: unable to find solutions in finite time; or trapped in local optimum [23]. It is difficult to obtain accurate solutions by using traditional way [24]. The 0-1 knapsack problem is mainly solved by approximate algorithm [25]. The solution obtained by the approximate algorithm of knapsack problem cannot completely solve practical problems [26]. There are also several membrane computing methods for solving knapsack problems. A membrane algorithm was presented to solve multidimensional 0-1 knapsack problem in linear time by recognizer P systems with input and with active membranes using 2-division [27]. The time-free efficient solution was presented to multidimensional 0-1 knapsack problem by timed recognizer tissue P systems [28]. In order to get the accurate solution of knapsack problem in linear time, we propose a new tissue P system called Π_{KP} for 0-1 knapsack problem.

The organization of the paper is as follows: the tissue P system with cell division and knapsack problem are introduced in section 2; the definition of the P system and the rules set is proposed in section 3. In section 4, the results of the experiments executed on a membrane simulator prove the correctness and feasibility of Π_{KP}. The performance of the system is analyzed and compared with other membrane computing method. The conclusions are proposed in the final section.

II. FOUNDATION

A. TISSUE P SYSTEM WITH CELL DIVISION

During the computation of the tissue P system, the membrane structure of individual membranes will not change [30], mainly depending on the material exchange between membranes and membranes, membranes and environment. Tissue P system is a network of membranes [31]. The symport/antiport rules are the modes of communication between membranes and environment [32]. The focus of the tissue P system is the collaboration between membranes. In the membrane calculation, the membranes and objects within them are considered as a complete whole, and the relative independence can perform the specific calculation independently, which makes the P system have great parallelism and non-deterministic [33]. Computational performance of P systems can be equivalent to the Turing machine and improve the calculation efficiency. P-system can generate exponential space solutions in linear time and has advantages in solving NP problems.

Tissue P system with cell division is a variant of the tissue P system. This variant increases cell division function. The rule set in tissue P system with cell division adds division rules. The progeny membranes have the same label as their parent membranes. Other activities of the same membrane are inactive temporarily at the time of mitosis in living membrane. According to this feature, other rules in the same membrane are inactive when the division rule is working. The rules in each membrane follow the principle of maximum parallelism and non-determinism.

A tissue P system (initial degree p$\geqslant$1) with cell division is a tuple:

$$\Pi_{KP} = (\Gamma, \varepsilon, w_1, w_2, \ldots, w_p, R, i_0)$$

where:

(1)Γ is a finite non empty alphabet of objects;

(2)$\varepsilon \subseteq \Gamma$, representing objects in the environment;

(3)$w_1, w_2, \ldots, w_p$, the set of objects in the cell in the initial state;

(4)R, is a set of rules of the system, and the rules of a divided organization of P systems are usually divided into two kinds:

$(i, u/v, j)$, this rule is the symport/antiport rule also called communication rule. The communication rule not only contains the objects' membranes, but also indicates the destination of the objects' movement. The object u belongs to membrane i, object v belongs to membrane j, transfers u from membrane i to membrane j, and transfers v of membrane j to membrane i, which can directly communicate between membranes and between the membranes and the environment through this rule. The unidirectional transport rule allows transmembrane transmission of the objects in the membrane in the same direction, and the reverse transport rule allows transmembrane transmission of the objects in the cell in the opposite direction. This is a reference to how the membranes work in biological tissues.

$[a]_i \rightarrow [x]_i[y]_i$, is a division rule. When there is an object a in a membrane i, this membrane can be divided into two new membranes with the same label i, and a is replaced by x or y in different sub-membranes respectively. These new membranes contain all the information in the original membranes except object a.

The execution of rules in the P system has maximum parallelism. Any object that can participate in the reaction must apply the rules as much as possible, but it should be noted that each object in the membrane can only act on one rule at each step. In a single step, all objects that satisfy the rules participate in the operation, but sometimes an object maybe in multiple rules. Therefore, there is a competitive relationship between the rules. In order to the system to run smoothly, priority is attached to rules. The priority of division rules is higher than that of communication rules. Other rules in the same membrane are suppressed when the division rule runs. In order to distinguish the priority of rules, numbers are added into the corresponding rules. A higher priority corresponds to a smaller number. Lower priority rules cannot get active when higher priority rules are applied. Priority exists only between rules in the same membrane.

B. 0-1 KNAPSACK PROBLEM

The knapsack problem was proposed by Mekrel and Helman in 1978. Among many types of knapsack problem, 0-1 knapsack problem is the most primitive and basic knapsack problem, and other types of knapsack problem can usually be converted to 0-1 knapsack problem [34]. The 0-1 Knapsack

problem is a typical NP-complete combinatorial optimization problem [35], which aims to maximize the value of items under backpack capacity constraints. Meanwhile the 0-1 knapsack problem can be described as a decision problem. The carrying capacity of backpack is fixed. Each item is unique and has two attributes of weight and value. Compare the total weight of the item in the backpack with the target value. When the total weight is greater than the target value, the answer is ''yes'', otherwise the answer is ''no''. The mathematical model of the knapsack problem is:

$$\begin{cases} max\, f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} v_i x_i \\ s.t. \sum_{i=1}^{n} w_i x_i \leqslant W \\ x_i \subseteq \{0, 1\}, \quad i = 1, 2, \ldots n \end{cases} \tag{1}$$

the total amount of items is n; i is a positive integer and represents the amount of items, x_i is a decision variable, if the item is selected, $x_i = 1$; otherwise $x_i = 0$; v_i and w_i respectively indicate the value and weight of item i-th; W represents the total weight of the backpack. Then the knapsack problem can be described as: there are n items, the weight of each item is w_i, the value is v_i, and the total weight of all items loaded with knapsack cannot exceed W at the same time.

The knapsack problem is to find the right combination of items which meet the requirements. It is worth noting that knapsack problem is a NP problem, which means that with the increase of the amounts, the time complexity of solving knapsack problem also increases significantly. So it is a huge project to find the accurate solutions. Because of the parallelism and division operation of tissue P system with cell division, all solutions can be obtained in a short time. Moreover, the uncertainty and the maximal parallelism also do some help in screening operation. It is feasible to generate all solutions and then screen out the qualified solutions in polynomial time even linear time. Taking into account all aspects of performance the tissue P system with cell division is chosen to solve knapsack problem.

III. π_{KP} FOR SOLVING 0-1 KNAPSACK PROBLEM

To some extent, the parallelism is guaranteed by making the computation of different parts run independently when necessary. It is an important characteristic of P system. As the operations among different membranes are independent of the others, the computations are executed in maximum parallel. A membrane system consists three important parts: membrane structure, object set and the rules. Membrane structure provides the environment for calculation. The object set contains all the calculated entities. The control of objects appearance time is helpful to the full realization of response. The rules connect objects with membrane structures to guarantee the smooth progress of the calculation. All of them work together to ensure the normal operation of the system. The next two subsections introduce the membrane structure and the rule set of Π_{KP}.

A. DEFINITION

Π_{KP} is proposed for solving knapsack problem. The process of Π_{KP} mainly contains two steps: generating all solutions in a short time because of the maximum parallelism of the P system, then filtering the wrong answers from the generated solutions according to the limitation of knapsack problem. If there exists a solution satisfying the reference value, then Π_{KP} can obtain the accurate solutions of knapsack problem in linear time. This system is defined as follow: W represents the unit weight loaded by knapsack, and its superscript $|W_{max}|$ represents the numerical value of capacity. So the capacity of knapsack is $W^{|W_{max}|}$. Let $\{A_1, A_2, \ldots, A_n\}$ be a finite set of items. The reference value is v'. The tissue P system with cell division is established with an initial degree of 2, suppose there are n items that need to be placed in the knapsack:

$$\Pi_{KP}(n, W, v') = \{\Gamma, \sigma_1, \sigma_2, R, \varepsilon, i_o\}$$

where:

$$\begin{aligned} \Gamma = {} & \{A_i, B_i, \bar{B}_i, e_i, \bar{e}_i, f_i, \bar{f}_i, x_i, w_i, v_i : 1 \leq i \leq n\} \\ & \cup\{a_i : 1 \leq i \leq n + 12\} \\ & \cup\{b_i : 1 \leq i \leq n\} \\ & \cup\{c_i : 1 \leq i \leq 10\} \\ & \cup\{F_i : 1 \leq i \leq 8\} \\ & \cup\{g, h, W, v', v'_1, v'_2, v'_3, \} \\ & \cup\{\lambda, \delta, k, E, D\} \end{aligned}$$

Γ is the finite multiple set of objects in system. σ_1 and σ_2 represent membrane *1* and membrane *2* respectively. ε represents the multiple set of objects in the environment. i_o indicates the membranes which contain legal solutions.

$$\begin{aligned} \sigma_1 &= \{a_1, b_1\} \\ \sigma_2 &= \{A_1, \ldots, A_n, D, w_1^{|w_1|}, w_2^{|w_2|}, \ldots, w_n^{|w_n|}, v_1^{|v_1|}, v_2^{|v_2|}, \\ &\qquad \ldots, v_n^{|v_n|}, W_m^W ax\} \\ \varepsilon &= \Gamma \backslash \{a_1, b_1, A_i, B_i, \bar{B}_i, D, w_i^{w_i}, v_i^{v_i}, W : 1 \leq i \leq n\} \end{aligned}$$

These objects of the system could be divided into seven subsets according to their functions. Objects in the first subset are representative of items or the attributes of items: A_i means items that can be loaded into knapsack; B_i and $\bar{B}_i$ represent whether to put the i-th item into the knapsack or not respectively; x_i denotes that the i-th item will be put into the knapsack in candidate solution. Objects a_i, b_i, c_i and F_i are responsible for the control system process which helping the system to smoothly go through different stages. v'_1 represents the waiting to be solved maximum value whether to be loaded into a backpack or not when the capacity of the knapsack is constant. In the last subset, objects δ is the dissolution signs of the membrane, object λ represents nothing. $i_o = i_0$ represents environment is the output region. Assume that the amount of each object in the environment is sufficient. As long as the conditions are satisfied, the rules operate with maximum parallelism.

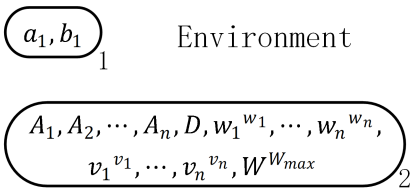

FIGURE 1. The initial configuration of Π_{KP}.

There are two membranes in the environment at initial configuration, marked *1* or *2* respectively. These two membranes work in parallel. Inside the membrane *1* are the control symbols. Membrane *2* contains the data to be solved such as weight set of items $\{w_1^{|w_1|}, w_2^{|w_2|}, \ldots, w_n^{|w_n|}\}$, value set of items $\{v_1^{|v_1|}, v_2^{|v_2|}, \ldots, v_n^{|v_n|}\}$. The calculation of the Π_{KP} system is carried out in the membranes *2*. The specific objects they contain are shown in Figure 1. A rounded rectangle represents a single membrane. The number at the lower right of each rounded rectangle represents the label of the membrane. The materials in each membrane are represented by corresponding letters. This representation is referenced in all subsequent figures in this article.

The theory of Π_{KP} is that all possible solutions can be obtained in a very short time by parallelism of P system, even there is a large amount of items. Each membrane contains a possible solution, some membranes in which the total weight exceeds the capacity and some membranes in which the total value is under the reference value are removed by rules in a few steps. The solutions that satisfy the requirements of the knapsack problem could be obtained in the remained membranes.

There are three important stages to solve knapsack problem based on tissue P system with cell division:

1) Generation stage

According to the membrane division, a lot of copies of membrane *2* are generated in this stage. The dividing process of the membranes *2* is shown in Figure 2. Activities between membranes *2* are carried out in parallel. The total amount is 2^n for those membranes *2* at the end of this phase. There are only two possibilities for each item in a membrane *2*: selected or not. The objects in every membrane represent an item selection. The set of the membranes includes all the selection like an exhaustive method. The system contains all the combinations of items at the end of the generation stage.

2) Filtering stage

In knapsack problems, the capacity of the knapsack is limited. The membranes that generated in the previous stage may be dissolved. It depends on the sum of the weight of the selected items. If the total weight of the items in the candidate solution exceeds the capacity, the corresponding membranes will be dissolved. In other words, the sum of weights in all the remained membranes is no more than the contents of knapsack.

3) Checking stage

The value of the items in each membrane has been calculated in the previous stage. Some membranes may be removed by comparing the total value with the reference value. If the total value in a membrane is no less than the reference value, this membrane contains a desired solution. If the total value in a membrane is no less than the reference value, this membrane contains a desired solution. On the contrary, if the total value in a membrane is less than the reference value, this membrane is dissolved. If there is existing any membrane, the solution that both satisfy the knapsack problem constraints and achieve the reference value can be obtained from the membrane.

B. RULE DESCRIPTION

The rules of P systems are automatically executed when the system satisfies the execution conditions of these rules. When the rules in the same membrane compete, the rules with high priority are active. The rules operate in accordance with maximum parallelism and uncertainty.

The rules in Π_{KP} could be classified into four categories to describe.

(1) Global rules

$$r_{1,i} \equiv (1, a_i/a_{i+1}, 0), \quad for\ i = 1, 2, 3, \ldots, n+11$$

$r_{1,i}$ are global rules to control the whole process, and it is always active when the Π_{KP} is calculating. In these rules, a_i is a control symbol and works as a global counter of the amount of steps performed. The subscript of the object a_i will be added one while the system execute onestep that also can be called timeunit Π_{KP} system terminates the operation when a_{n+12} appears in the membrane *1*. Until that time, every solution satisfying the constraint will be in one membrane.

(2) Generation rules

$$\begin{aligned}
r_{2,i} &\equiv [A_i]_2 \rightarrow [B_i]_2[\bar{B}_i]_2, \quad for\ i = 1, 2, 3, \ldots, n \\
r_{3,i} &\equiv (1, b_i/b_{i+1}, 0), \quad for\ i = 1, 2, 3, \ldots, n \\
r_4 &\equiv (1, b_{n+1}/D, 2) \\
r_5 &\equiv (1, D/F_1, 0) \\
r_{6,i} &\equiv (1, F_i/F_{i+1}, 0),\ 2, \quad for\ i = 1, 2, 3, \ldots, 7 \\
r_7 &\equiv (1, F_4/c_4\ E, 2), \quad 1 \\
r_8 &\equiv (1, F_8/c_8\ v_1', 2)
\end{aligned}$$

$r_{2,i}$ is a division rule. The original membrane *2* in initial configuration eventually produced 2^n membranes *2* by this rule. In generated membrane, only an object B_i or $\bar{B}_i$ will be in each membrane. B_i means choosing i-th item and $\bar{B}_i$ indicates that i-th item is not selected.

The amount of generating membranes associates with the amount of items n. In order to control the division process, symbol b_i becomes a pedometer for membrane division in rule $r_{3,i}$. This process will consume n time unit and generate 2^n membranes. At the same time, the amount of b_i is increased. When the objects b_{n+1} appear in membrane *1*, the rule r_4 will be activated, which will transfer b_{n+1} to every membrane *2* and accept D from every membranes *2*.

When the membrane *1* receive objects D, that means all candidate solutions are generated and stored in membranes *2*

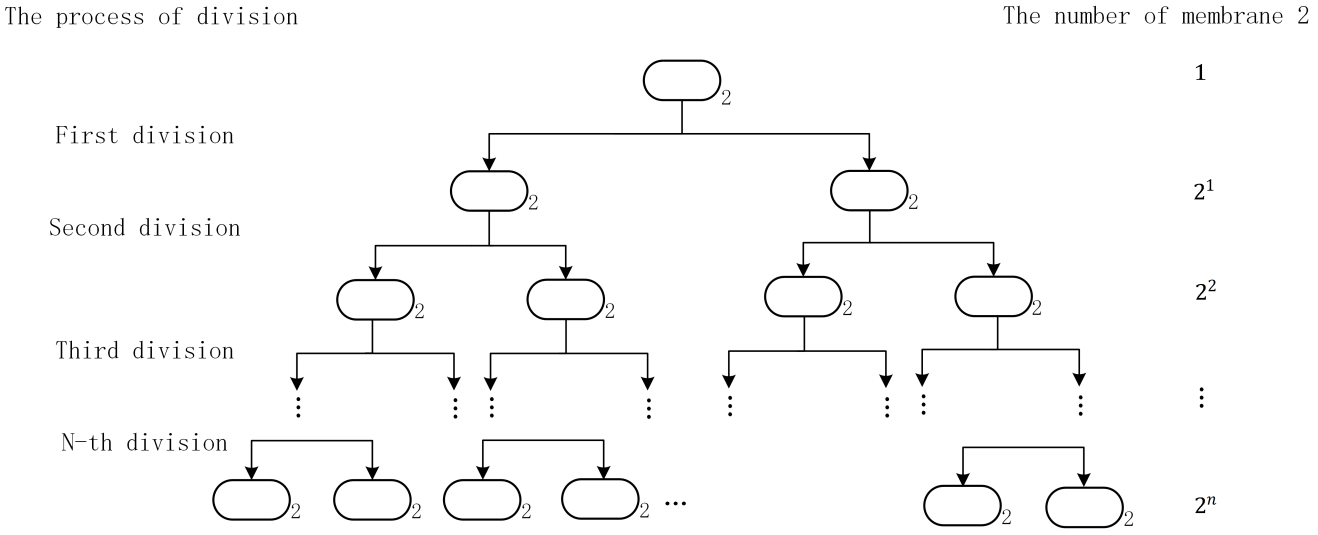

FIGURE 2. The division process of membrane 2.

through cell division. In other words, the generation stage has finished and the filtration stage will begin. The membrane *1* will obtain the new control symbols F_i from the environment by r_5. F_i is a controller in both filtration stage and checking stage. It is used to count the amount of membranes satisfying the conditions in the final result. The superscript of F_8 represents the amount of sub membranes which meet the condition at each configuration. There is a priority between $r_{6,i}$ and r_7 in membrane *1*, that means when the substances in membrane *1* satisfy both the running conditions of $r_{6,i}$ and r_7, only r_7 can work normally. r_8 is to remove solutions less than reference value

(3) Filtration rules

$$
\begin{aligned}
r_9 &\equiv (2, b_{n+1}/c_1\ d^n, 0)\\
r_{10,i} &\equiv (2, c_i/c_{i+1}, 0), \quad for\ i = 1, 2, 3, \ldots, 5, 7, 8\\
r_{11,i} &\equiv (2, \bar{B}_i d/\bar{e}_i^{W_{imax}} \bar{f}_i^{v_{imax}}, 0), \quad for\ i = 1, 2, 3, \ldots, n\\
r_{12,i} &\equiv (2, B_i d/x_i e_i^{W_{imax}} f_i^{v_{imax}}, 0), \quad for\ i = 1, 2, 3, \ldots, n\\
r_{13,i} &\equiv (2, \bar{e}_i w_i/\lambda, 0), \quad for\ i = 1, 2, 3, \ldots, n\\
r_{14,i} &\equiv (2, \bar{f}_i v_i/\lambda, 0), \quad for\ i = 1, 2, 3, \ldots, n\\
r_{15,i} &\equiv (2, e_i w_i/g, 0), \quad for\ i = 1, 2, 3, \ldots, n\\
r_{16,i} &\equiv (2, f_i v_i/h, 0), \quad for\ i = 1, 2, 3, \ldots, n\\
r_{17} &\equiv (2, gW/k, 0), \quad 1\\
r_{18} &\equiv (2, g/E, 0), \quad 2\\
r_{19} &\equiv (2, F_4/\delta, 0)
\end{aligned}
$$

When the membrane *2* receive object b_{n+1}. The rule r_9 will be activated, which will obtain object c_1 and d^n from the environment. c_i is used as a counter, and d^n is used as the amount of items.

$r_{11,i}$ and $r_{12,i}$ are executed at the same time, obtaining the maximum weight and maximum value of a single item from the environment. The object x_i is introduced to preserve the selection of objects. If x_i appears in a membrane *2*, $i_t h$ item will be placed in the knapsack.

$r_{13,i}$ $r_{16,i}$ are executed simultaneously. There may bring two different situations according to the choice. If one item is chosen, $r_{13,i}$ and $r_{14,i}$ will dissolve the value and weight of this item. Otherwise, $r_{15,i}$ and $r_{16,i}$ will produce uniform objects g and h, the amount of these object represents total value and total weight of the items in the knapsack respectively.

As the priority of r_{17} is higher than that of r_{18}, the objects gW will be replaced by k by r_{17}. If there the amount of g is more than that of W, r_{18} will execute to get object E from environment. E is used to represent overweight membranes which will be dissolved by r_{19}. Through r_7 each illegal membranes *2* consume one F_4. Correct the superscript of the F_4 to record the amount of legitimate membranes *2* at this time.

(4) Check rules

$$
\begin{aligned}
r_{20} &\equiv (2, c_6/c_7\ v_1^{\prime v_{max}} v_2^{\prime total-v_{max}}, 0)\\
r_{21} &\equiv (2, v_1'\ h/v', 0)\\
r_{22} &\equiv (2, v_2^{\prime total-v_{max}}/v_3^{\prime total-v_{max}}, 0)\\
r_{23} &\equiv (2, F_8/\delta, 0)\\
r_{24} &\equiv (2, v_3'\ h/v', 0)
\end{aligned}
$$

In the checking stage, the reference value $v_1^{\prime v_{max}}$ will be obtained from the environment by r_{20} at first, then compare the total value and the reference value of items in each membrane *2* by r_{21}.The object v_3' is added as controlling the transformation of value by r_{22}.

The function of r_8 is to transport F_8 into membranes *2* which less than reference value. Then these membrane *2* will be dissolved according to r_{23}. r_{24} converts h in membranes *2* larger than or equal to the reference value to v'. After r_{21} and r_{24}, the total value of the items in membrane is expressed by v'.

TABLE 1. The weight and value of the items.

Items	Weight	The form of weight in Π_{KP}	Value	The form of value in Π_{KP}
1	4	w_1^4	9	v_1^9
2	5	w_2^5	11	v_2^{11}
3	2	w_3^2	4	v_3^4
4	1	w_4^1	2	v_4^2
5	6	w_5^6	13	v_5^{13}

Finally, the existing membranes are the solutions that satisfies both the limited condition of the knapsack and the reference value. The set of x_i is the combination of items in backpack, and the superscript of v' is the total value of this combination. The superscript of F_8 in membrane *1* indicates the sum of legal solutions.

IV. INSTANCE AND ANALYSIS

The basic structure and operation rules of Π_{KP} have been constructed as above. In this section, An instance is used to describe the whole process of Π_{KP} by rules and executed on a membrane simulator, UPSimulator. According to the results of the experiment, we analyze the performance and complexity of Π_{KP} and compare it with other membrane computing methods.

A. INSTANCE DESCRIPTION

A knapsack problem with five items was chosen to illustrate the system. The specific data information for these items is shown in Table 1. The capacity of knapsack is 8. The reference value is 15. The objective of the system is to find a combination of items with a total value greater than 15 when the content of knapsack is no more than 8. There are two important numbers in subsequent operations: the maximum weight 6 and the maximum value 13.

At the beginning of Π_{KP}, there is a global controller a_1 and a generating stage controller b_1 in membrane *1* according to the definition. Meanwhile, the value and weight of each item have been placed in the membrane *2*, as shown in Figure 3. A_i represents the *i*-th item in this instance. W_8 is the loading weight of knapsack. The object D is used as control symbol at the end of generation stage.

The rules $r_{1,i}$ start counting the steps of execution. In this instance, the amount of items is five, so the system will take five timeunits to generate 32 membranes *2* represent exponential selections by rules $r_{2,i}$. Each membrane *2* has itself unique candidate solution. When the membrane is dividing, other activities in the same membrane cease. The symbol D

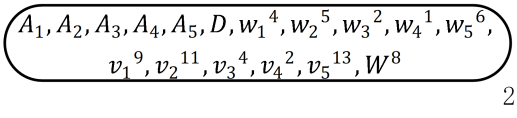

FIGURE 3. Initial membrane *2* of the instance.

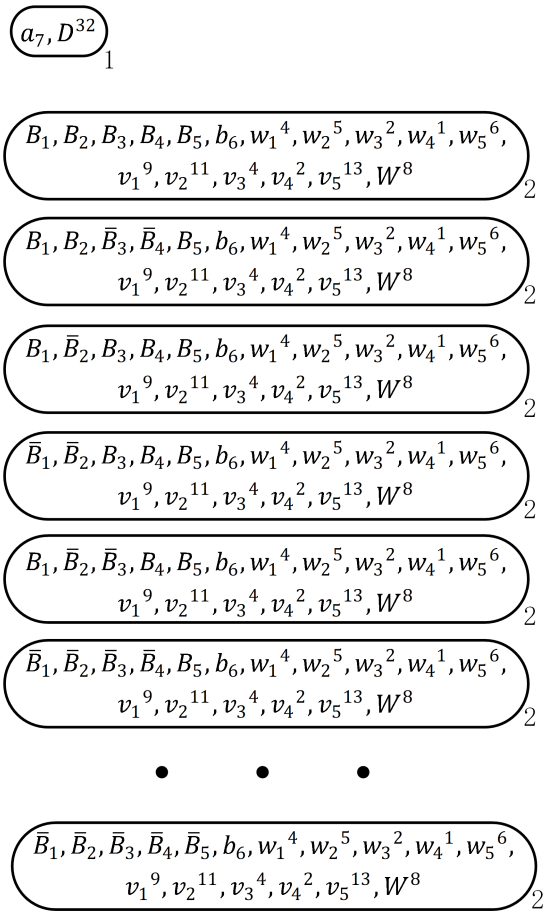

FIGURE 4. The configuration of the example after generation stage in Π_{KP} system.

in each membrane *2* are transported to membrane *1* according to r_4 in the sixth timeunit, which indicates the end of the generation stage. Figure 4 shows the state of system configuration after six timeunits. In membrane *1*, the controller a_7 means that it has passed six timeunits. The symbols D which have reached in membrane *1* means the end of the generation phase, in other word, all the membranes *2* are no longer duplicated. We only list 7 membranes represented exponential selections among 32 membranes in this picture.

The seventh timeunit separates the generation stage and filtering stage. In this unit, all the membranes bring new control symbols. In membrane *1*, the object F_1 from the environment replaces of D by r_5, the object c_1 and d_5 from the environment replaces of b_6 by r_9. The subscript of F_i will increase gradually in another seven timeunits, meanwhile each membrane *2* will carry out a series of complex calculations.

Two random membranes *2* are chosen as an instance to illustrate the procedure that last the next five timeunits, shown in Figure 5. In this process, the same layer of membranes are at the same point. In order to distinguish the membranes in the figure, different membranes are marked with (a), (b) and (c) at the bottom of the figure.(a) is membrane *1*, (b) and (c) labeled different membranes *2* respectively. The first

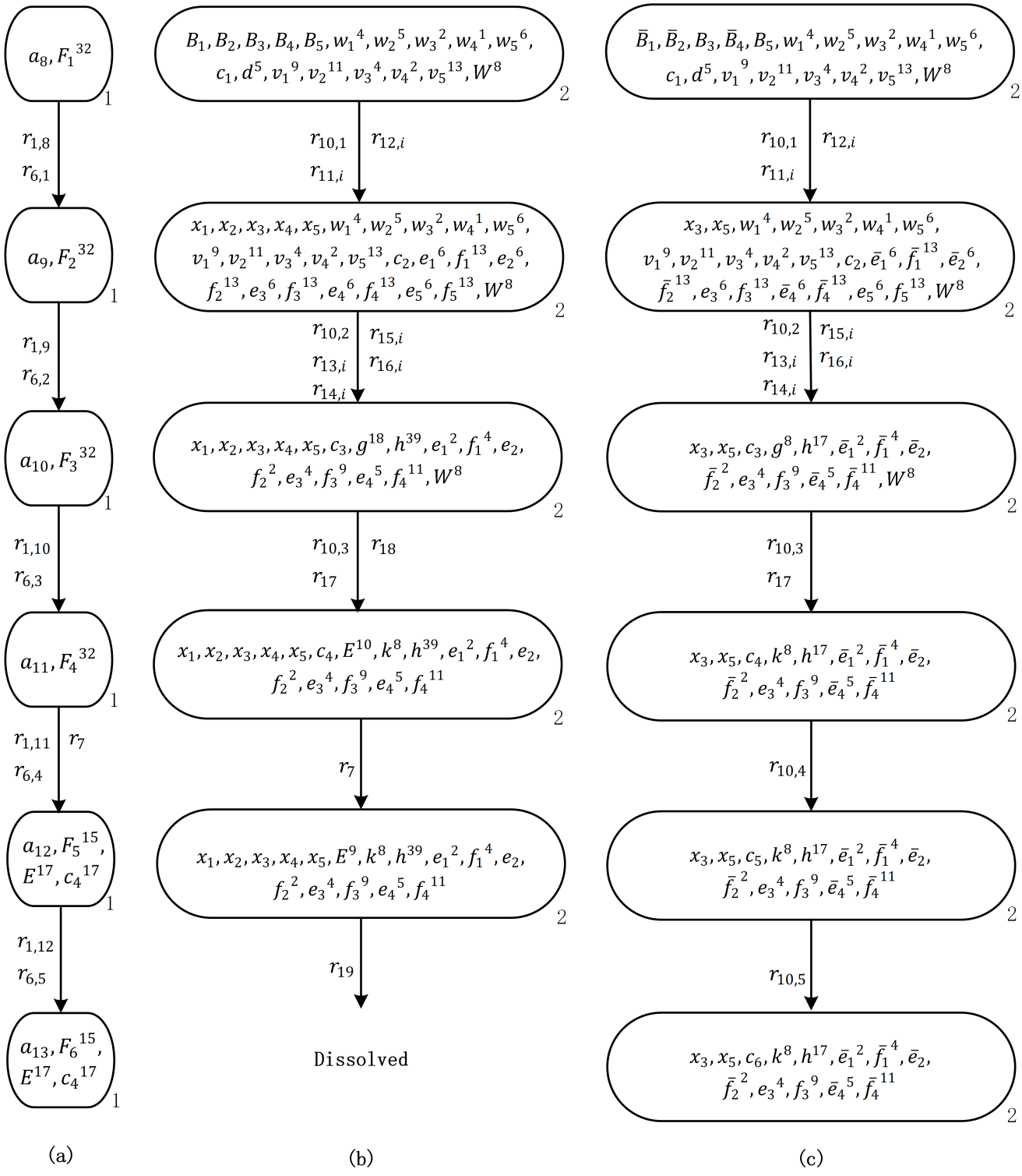

FIGURE 5. The change of two sample membranes during filtering stage in Π_{KP} system.

row in this picture represents the start of filtering phase. The last row in this picture represents the state after the twelfth timeunits, At that time, some membranes may be dissolved like the membrane in (b), the others may be remained like the membrane in (c).

The filtering phase begin at the eighth timeunit, $r_{6,i}$ became active on F_i in membrane *1*, simultaneously $r_{10,i}$, $r_{11,i}$ and $r_{12,i}$ are activated in parallel in membranes *2*.The eighth timeunit is between a_8 and a_9. $r_{10,i}$ is responsible for the operation of counter c_i, so this rule has been active since c_1

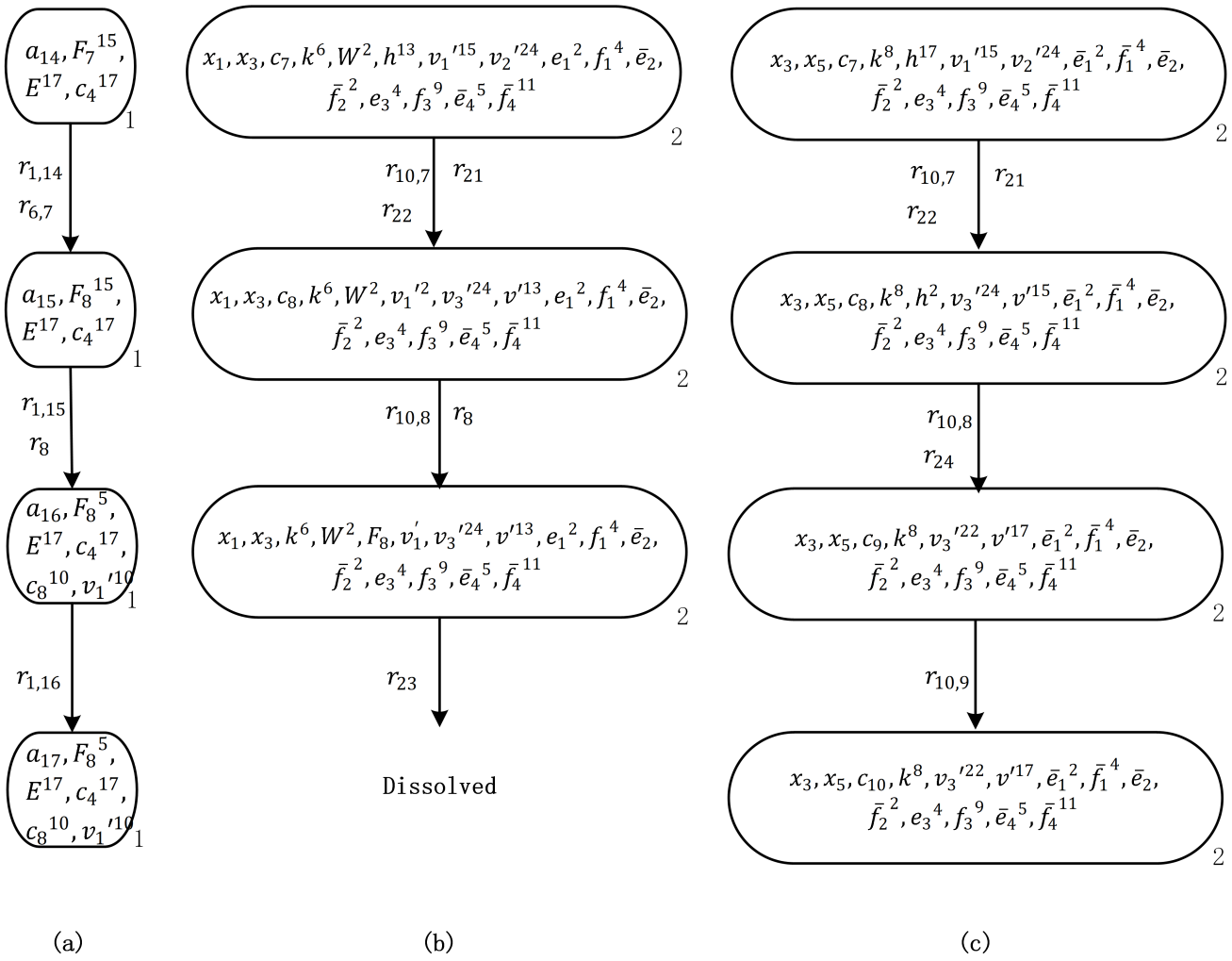

FIGURE 6. The change of two sample membranes during checking stage in Π_{KP} system.

was introduced. There must be the same amount of items as b because b reacts one-to-one with B_i or $\bar{B}_i$. Then through $r_{11,i}$ and $r_{12,i}$ maximum weight and maximum value of the items are introduced into membranes *2* from the environment, and the status of the object selection is recorded by x_i.

In the ninth timeunit, in membranes *2* the rules $r_{13,i}$ $r_{16,i}$ come to active simultaneously to calculate the total weight and total value of the items in membrane *2*. The amount of g is the total weight, and the amount of h is the total value.

The total weight of the selected items in each membrane *2* is compare with the capacity of backpack in the tenth timeunit. The part of weight that does not exceed the backpack capacity is converted to k by r_{17}. If the total weight in membrane *2* is more than the capacity, the extra object g is transported to get E by r_{18}. E is an object for filtering.

In the next two timeunit, there are two circumstances for membranes *2*. If the total weight is no more than the capacity, like the membrane *2* labled (c) in Figure 5, in these membrane, only $r_{10,i}$ is executing. Otherwise, the symbols E and c_4 are exchanging with the symbol F_4 in membrane *1* by r_7 as shown in (b) in Figure 5, and the overweight membranes will be dissolved by r_{19}.

In the thirteenth timeunit, the pedometer c_7, v_1' and v_2' are introduced into membranes *2* at the same time by r_{20}. v_1' stands for the reference value and is used to help identify membranes smaller than the reference value. v_2' is a temporary value for auxiliary computation. This timeunit is the beginning of checking stage.

Figure 6 shows the procedure of the fourteenth timeunit to the sixth timeunit. In this picture, we also select two membranes *2* from the remaining membranes in the system in order to complete the instructions of the checking stage to cover all cases.

In the fourteenth timeunit, the total value in the candidate membranes is compared with the reference value by r_21. v_1' and h react one-to-one to obtain a corresponding amount of v' by transporting. v' is used to represent the final unit value. All v_2' converts to v_3' by r_{22}. What v_3' represents is partial value which is less than the total value in the candidate solution and greater than the reference value.

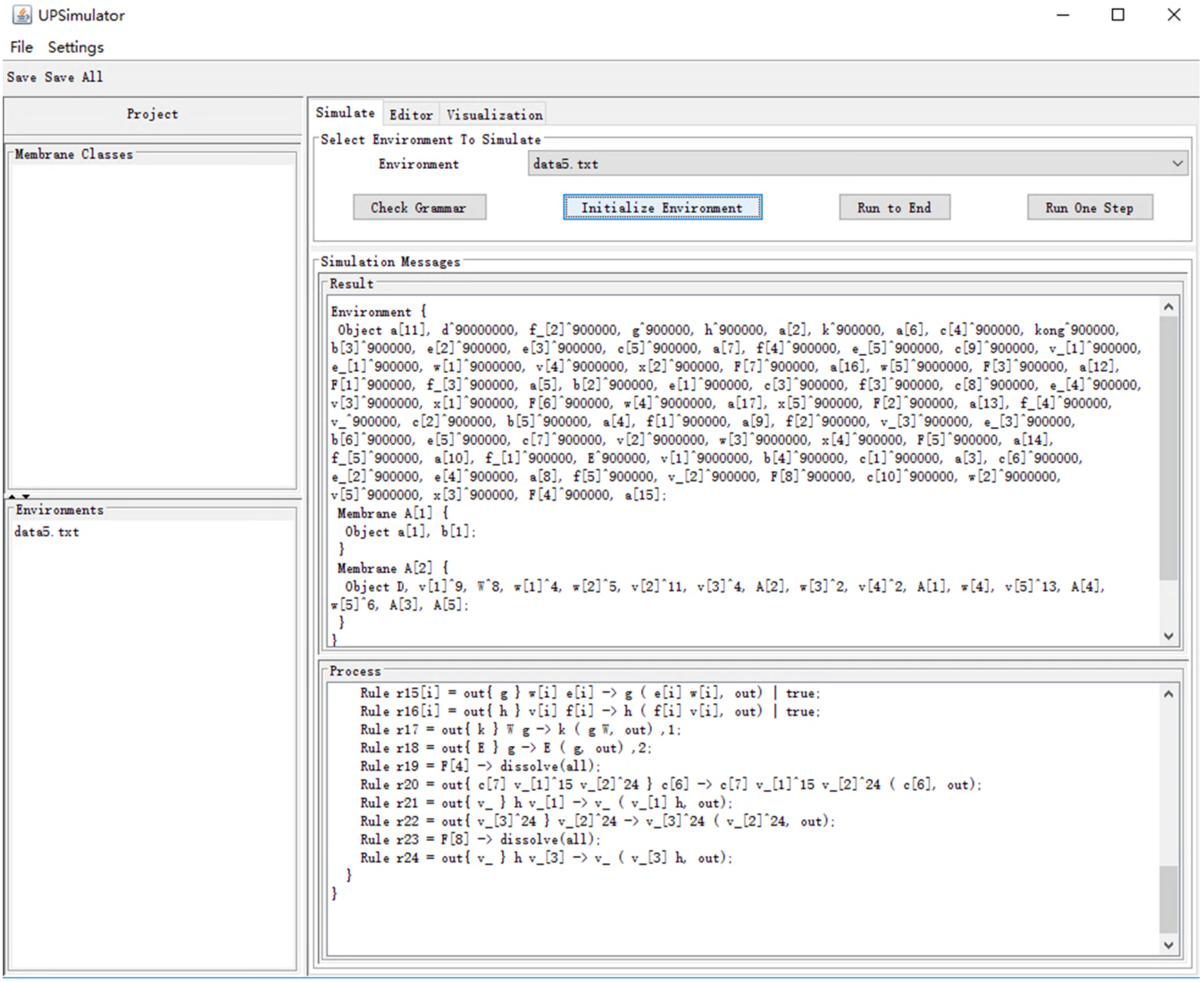

FIGURE 7. The initialization in UPSimulator.

There are two situations in the fifteen timeunit according to the total value of membranes *2*. If the total value in the membrane is less than the reference value, r_8 between membrane *1* and membranes *2* begins to activate. F_8 will appear in membranes *2* which less than the reference value according to the rule r_8. The second case is the same as the membrane *2* labeled (c) in Figure 6. The total value of the membrane is no less than the reference value. In this time unit, there are $r_{10,i}$ and r_{24} running in the membranes under the same situation as (c).

In the sixteen timeunit, the subscript of a_i is equal to 17, this is the final operation of Π_{KP}. The membranes which received F_8 are dissolved in this moment. The accurate solutions of the knapsack problem exist in the membranes which still exist in the environment. The superscript of F_8 in membrane *1* is the amount of legitimate solutions. Combination of items contained in these membranes represented by x_i. And the optimal solution can be calculated by the superscript of v'.

B. EXPERIMENTS

In order to verify the feasibility of Π_{KP}, some experiments are carried out on a special membrane simulator named UPSimulator. UPSimulator designs different concept interfaces to implement different rule types and different membrane structures, and could recognize and simulate the membrane classes and the environment defined in one or multiple structured text files [36].

Firstly, the example in the previous subsection is simulated in the simulator. Initialization is shown in Figure 7. The initial configuration of the system is displayed in the ''Simulation Message'' column. ''Membrane A[1]'' is membrane *1*. The objects in membrane *1* are stored in curly braces which follow the ''Membrane A[1]''. Membrane *2* and its objects are represented in a similar way to the membrane *1*. The objects in environment are stored in the curly braces which follow the ''Environment''. An object denotes as object name, subscript and cardinality. Object names are represented by letters or

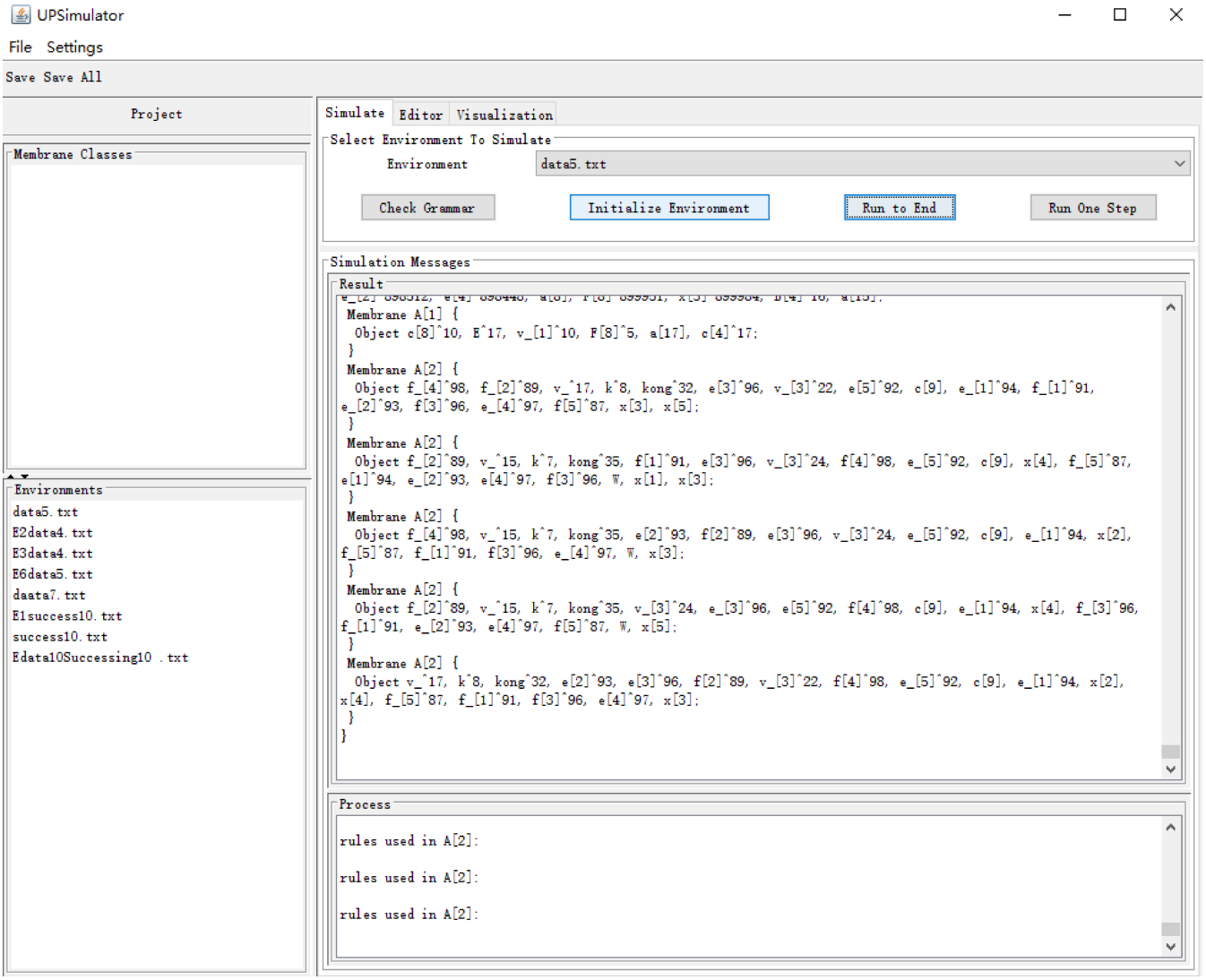

FIGURE 8. The result in UPSimulator.

letters with "_". Subscripts are represented by numbers in square brackets after letters. The amount of an object is represented by numbers segregated by superscript operator. Some objects may not have numeric values or subscripts. The amount of some objects in the environment are very large in order to let the reaction be sufficient, so the amount of each object in the environment is given a huge number, such as 900000. In Figure 7, the "Result" column shows the initial state of the system. The "Process" column shows the rules that the system can run at each step.

The simulation results can be obtained directly by clicking on the "Run to End" button. The simulation results of the example in the previous summary are shown in Figure 8. The symbol $\times[i]$ in Figure 8 corresponding to x_i in Π_{KP} means the i-th item is chosen in this membrane *2*. The object a[i] in membrane *1* is a control symbol, the value of its subscript is increased in the procedure. In this example, the stopping condition of the system is that the subscript of a[i] is up to 17. The amount of objects F[8] in membrane *1* in Figure 8 is 5, there are five solutions that ultimately reach reference value 15. Correspondingly, the amount of membrane *2* in Figure 8 is 5. The amount of "v_" shows the total value of items in membrane *2*. According to the order from top to bottom, the combination and total value of the items in the five solutions are shown in Figure 8. There are two optimum solutions: $\times[2]$, $\times[3]$, $\times[4]$ and $\times[3]$, $\times[5]$. The value of these two combinations is 17. This is the maximum total value of this 0-1 knapsack problem.

In order to prove the feasibility and correctness of the system, we have done more simulation experiments based on the other six instances. The experimental parameters including the weight of items, the value of items and the reference value are shown in Table 2. In Table 3, the third column shows the maximum value. The items that are put into the

TABLE 2. The parameters of six test problems.

Experiments	The amount of items	The weight of knapsack	The weight of items	The value of items	The reference value
1	10	300	95, 75, 23, 73, 50, 22, 6, 57, 89, 98	89, 59, 19, 43, 100, 72, 44, 16, 7, 64	388
2	4	20	6, 5, 9, 7	9, 11, 13, 15	30
3	4	11	2, 4, 6, 7	6, 10, 12, 13	22
4	10	60	30, 25, 20, 18, 17, 11, 5, 2, 1, 1	20, 18, 17, 15, 15, 10, 5, 3, 1, 1	52
5	7	50	31, 10, 20, 19, 4, 3, 6	70, 20, 39, 37, 7, 5, 10	100
6	5	80	15, 20, 17, 8, 31	33, 24, 36, 37, 12	128

TABLE 3. The simulation results of six test problems.

Experiments	The amount of membrane *2*	Maximum value	Optimum solution
1	5	388	x[1], x[2], x[6], x[9], x[5], x[10], x[8], x[7], x[4]; x[1], x[2], x[6], x[9], x[5], x[10], x[3], x[4]; x[1], x[6], x[9], x[5], x[10], x[8], x[4]; x[1], x[2], x[6], x[9], x[5], x[8], x[7], x[3]; x[1], x[2], x[6], x[9], x[5], x[8], x[7], x[4];
2	2	35	x[1], x[2], x[4]
3	2	23	x[2], x[4]
4	4	52	x[6], x[9], x[5], x[10], x[8], x[3], x[7]; x[3], x[4], x[5], x[7]; x[9], x[5], x[10], x[8], x[3], x[4]; x[6], x[9], x[10], x[8], x[4], x[3], x[7];
5	4	107	x[1], x[4]
6	1	130	x[1], x[2], x[3], x[4]

knapsack could make the maximum value and are shown in the fourth column. So the Π_{KP} system can obtain all the optimal solutions.

C. ANALYSIS

The experimental results show that the system can quickly find the best selection to make the maximum value. In other words, if there exists a combinations of items which satisfy the reference value, then all the selections will be obtained in the system in linear time. So the best solution is a membrane which contains the maximum value under the knapsack with certain capacity.

Reference [27] proposed uniform recognizer P systems with active membranes to solve the multidimensional 0-1 knapsack problem in linear time. In this system: all instance of the problem that have the same size are processed by the same P system, the answer is affirmative, an object *yes* will be sent to the environment changing the polarization of the skin to neutral. Otherwise, an object *no* will be sent to the environment changing the polarization of the skin to neutral. Reference [28] also presented a time-free solution to the multidimensional 0-1 knapsack problem. In this method, *yes* may be sent to the environment to exchange for object *no*. The system halts with object *yes* in the environment. On the contrary, object *no* will exist in the environment when the computation halts.

The two proposed method based on Tissue P Systems consider the 0-1 knapsack problem as a decision problem. They both output the solution, yes or no, to represent the answer, positive or negative. In this paper, if the best solution exists, it could be found by checking the remained membrane in linear time.

The execution time of Π_{KP} is $n + 12$, which contains: (1) generation stage, $n + 2$ steps; (2) filtering stage, 6 steps; (3) checking stage, 4 steps. n is the amount of items that will be put into the knapsack. The time consumed in the generation stage is related to n, because this stage is affected by division rules. In filtering stage and checking stage, only communication rules become active. Because the work between membranes in tissue P system is parallel, the total operation steps of these two stages are not affected by n. Obviously, Π_{KP} could solve the knapsack problem in linear time, and its time complexity is $O(n)$. It is much better than the methods that needs polynomial time or exponential time.

V. CONCLUSIONS

In this paper, a new P system Π_{KP} is proposed for solving 0-1 knapsack problem, a classical NP problem. Based on the parallelism of membrane computing, the existence of solutions that meet the requirements are judged and the specific contents of solutions are found in linear time. In other words, it could seek out the best solution in the remained membrane if the optimal solution exists. It is different from the previous membrane methods which decide whether the optimal solution exists or not. The proposed system has been implemented on a membrane simulator. The results of simulation prove the feasibility and correctness of this system. The idea of using Π_{KP} to solve knapsack problem is also applied to solve other NP problems, that's what we're going to do in the further work.

REFERENCES

[1] G. Paŭn, "Computing with membranes," *J. Comput. Syst. Sci.*, vol. 61, no. 1, pp. 108–143, Aug. 2000.
[2] G. Paŭn, "Computing with membranes: An introduction," *Bull. EATCS*, no. 67, no. 2, pp. 139–152, Feb. 1999.
[3] T. Wang *et al.*, "Application of neural-like P systems with state values for power coordination of photovoltaic/battery microgrids," *IEEE Access*, vol. 6, pp. 46630–46642, 2018.
[4] S. Pang, T. Ding, A. Rodríguez-Patón, T. Song, and Z. Phen, "A parallel bioinspired framework for numerical calculations using enzymatic P system with an enzymatic environment," *IEEE Access*, vol. 6, pp. 65548–65556, 2018.
[5] P. Sosík and L. Cienciala, "A limitation of cell division in tissue P systems by PSPACE," *J. Comput. Syst. Sci.*, vol. 81, no. 2, pp. 473–484, Mar. 2015.
[6] P. Andrei and G. Păun, "The power of communication: P systems with symport/antiport," *New Generat. Comput.*, vol. 20, no. 3, pp. 295–305, 2002.
[7] G. Păun, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "Tissue P systems with cell division," *Int. J. Comput. Commun. Control*, vol. 3, no. 3, pp. 295–303, Sep. 2008.
[8] B. Song, T. Song, and L. Pan, "A time-free uniform solution to subset sum problem by tissue P systems with cell division," *Math. Struct. Comput. Sci.*, vol. 27, no. 1, pp. 17–32, Jan. 2017.
[9] D. Díaz-Pernil, M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "A fast solution to the partition problem by using tissue-like P systems," in *Proc. 3rd Int. Conf. Bio-Inspired Comput., Theories Appl. (BIC-TA)*, Adelaide, SA, Australia, Sep./Oct. 2008, pp. 43–48.
[10] D. Díaz-Pernil, M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "A linear–time tissue P system based solution for the 3–coloring Problem," *Electron. Notes Theor. Comput. Sci.*, vol. 171, no. 2, pp. 81–93, Jul. 2007.
[11] D. Díaz-Pernil, M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, and A. Riscos-Núñez, "A uniform family of tissue P systems with cell division solving 3-COL in a linear time," *Theor. Comput. Sci.*, vol. 404, no. 1, pp. 76–87, Sep. 2008.
[12] A. H. Christinal, D. Díaz-Pernil, and T. Mathu, "A uniform family of tissue P systems with protein on cells solving 3-coloring in linear time," *Natural Comput.*, vol. 17, no. 2, pp. 311–319, Jun. 2018.
[13] D. Díaz-Pernil, M. A. Gutiérrez-Naranjo, M. J. Pérez-Jiménez, and A. Riscos-Núñez, *Solving the Independent Set Problem by Using Tissue-Like P Systems With Cell Division*. Berlin, Germany: Springer-Verlag, 2009, no. 1, pp. 213–222.
[14] X. Zhang, S. Wang, Y. Niu, and L. Pan, "Tissue P systems with cell separation: Attacking the partition problem," *Sci. China Inf. Sci.*, vol. 54, no. 2, pp. 293–304, Feb. 2011.
[15] Y. Niu, Z. Wang, and J. Xiao, "A uniform solution for vertex cover problem by using time-free tissue P systems," in *Bio-Inspired Computing—Theories and Applications* (Communications in Computer and Information Science), vol. 562, M. Gong, P. Linqiang, S. Tao, K. Tang, and X. Zhang, Eds. Berlin, Germany: Springer, 2015, pp. 306–314.
[16] H. A. Christinal, R. R. John, D. A. Chandy, and M. A. Gutiérrez-Naranjo, "Solving the bin-packing problem by means of tissue P system with 2-division," in *Unconventional Computation and Natural Computation* (Lecture Notes in Computer Science), vol. 10240, 2017, pp. 170–181.
[17] T.-C. Lu and G.-R. Yu, "An adaptive population multi-objective quantum-inspired evolutionary algorithm for multi-objective 0/1 knapsack problems," *Inf. Sci.*, vol. 243, pp. 39–56, Sep. 2013.
[18] X. Zhang, S. Huang, Y. Hu, Y. Zhang, S. Mahadevan, and Y. Deng, "Solving 0-1 knapsack problems based on amoeboid organism algorithm," *Appl. Math. Comput.*, vol. 219, no. 19, pp. 9959–9970, Jun. 2013.
[19] A. Layeb, "A hybrid quantum inspired harmony search algorithm for 0–1 optimization problems," *J. Comput. Appl. Math.*, vol. 253, pp. 14–25, Dec. 2013.
[20] B. Duhart, F. Camarena, J. C. Ortiz-Bayliss, I. Amaya, and H. Terashima-Marín, "An experimental study on ant colony optimization hyper-heuristics for solving the knapsack problem," in *Proc. Mex. Conf. Pattern Recognit.*, J. Martínez-Trinidad, J. Carrasco-Ochoa, J. Olvera-López, and S. Sarkar, Eds. Cham, Switzerland: Springer, May 2018, pp. 62–71.
[21] A. Rezoug, M. Bader-El-Den, and D. Boughaci, "Guided genetic algorithm for the multidimensional Knapsack problem," *Memetic Comput.*, vol. 10, no. 1, pp. 29–42, 2018.
[22] I. M. Ali, D. Essam, and K. Kasmarik, "An efficient differential evolution algorithm for solving 0–1 knapsack problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Rio de Janeiro, Brazil, Jul. 2018, pp. 1–8.
[23] T. Saraç and A. Sipahioglu, "Generalized quadratic multiple knapsack problem and two solution approaches," *Comput. Oper. Res.*, vol. 43, pp. 78–89, Mar. 2014.
[24] C. Patvardhan, S. Bansal, and A. Srivastav, "Solving the 0–1 quadratic knapsack problem with a competitive quantum inspired evolutionary algorithm," *J. Comput. Appl. Math.*, vol. 285, pp. 86–99, Sep. 2015.
[25] R. Julien, R. F. José, and D. S. Yves, "The inverse 0,1-knapsack problem: Theory, algorithms and computational experiments," *Discrete Optim.*, vol. 10, no. 2, pp. 181–192, May 2013.
[26] E. Bas, "A capital budgeting problem for preventing workplace mobbing by using analytic hierarchy process and fuzzy 0-1 bidimensional Knapsack model," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 12415–12422, 2011.
[27] L. Pan and M.-V. Carlos, "Solving multidimensional 0–1 knapsack problem by P systems with input and active membranes," *J. Parallel Distrib. Comput.*, vol. 65, no. 12, pp. 1578–1584, Dec. 2005.
[28] X. Liu, Z. Li, J. Suo, Y. Ju, J. Liu, and X. Zeng, "Solving multidimensional 0-1 knapsack problem with time-free tissue P systems," *J. Appl. Math.*, vol. 2014, no. 5, pp. 1–6, Mar. 2014.
[29] P. Guo, J. Zhu, and J. Chen, "A family of P systems for 0-1 knapsack problem," *J. Residuals Sci. & Technol.*, vol. 14, no. 3, pp. 256–270, Sep. 2017.
[30] A. Leporati, L. Manzoni, G. Mauri, and C. Zandron, "Tissue P systems can be simulated efficiently with counting oracles," in *Proc. Int. Conf. Membrane Comput. (CMC)*, in Lecture Notes in Computer Science, vol. 9504, Aug. 2015, pp. 251–261.
[31] H. Adorna, A. Alhazov, L. Pan, and B. Song, "Simulating evolutional symport/antiport by evolution-communication and vice versa in tissue P systems with parallel communication," in *Proc. Int. Conf. Membrane Comput. (CMC)*, in Lecture Notes in Computer Science, vol. 10725, Jul. 2017, pp. 1–14.
[32] H. Peng and J. Wang, "A hybrid approach based on tissue P systems and artificial bee colony for IIR system identification," *Neural Comput. Appl.*, vol. 28, no. 9, pp. 2675–2685, Sep. 2016.
[33] L. Pan, B. Song, and G. Zhang, "Tissue P systems with rule production/removal," in *Membrane Computing* (Lecture Notes in Computer Science), vol. 10725, M. Gheorghe, G. Rozenberg, A. Salomaa, and C. Zandron, Eds. Cham, Switzerland: Springer, 2017, pp. 230–244.
[34] Q. Yu, E. Xu, and S. Cui, "Streaming algorithms for news and scientific literature recommendation: Submodular maximization with a d-knapsack constraint," in *Proc. IOOC-ECOC*, Boston, MA, USA, 2016, pp. 585–590.
[35] S.-H. Zhan, Z.-J. Zhang, L.-J. Wang, and Y.-W. Zhong, "List-based simulated annealing algorithm with hybrid greedy repair and optimization operator for 0–1 knapsack problem," *IEEE Access*, vol. 6, pp. 54447–54458, 2018.
[36] P. Guo, C. Quan, and L. Ye, "A simulator for cell-like P system," in *Bio-Inspired Computing: Theories and Applications*, J. Qiao *et al.*, Eds. Singapore: Springer, 2018, pp. 108–143.

LIAN YE was born in Chongqing, China, in 1980. She received the Ph.D. degree in computer science from Chongqing University, China, in 2012, where she is currently a Teacher. Her research interests include the different aspects of artificial intelligence, artificial immune systems, and membrane computing.

JINHANG ZHENG was born in Da'an, China, in 1994. She received the bachelor's degree in communication engineering from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2016. She is currently pursuing the master's degree in computer technology with the School of Computer Science, Chongqing University. Her research interests include membrane computing and artificial immune systems.

PING GUO was born in Meishan, China, in 1963. He received the Ph.D. degree in computer science from Chongqing University, China, in 2004, where he is currently a Professor. His research interests include the different aspects of artificial intelligence and biological computing models.

MARIO J. PÉREZ-JIMÉNEZ received the degree in mathematics from the University of Barcelona and the Ph.D. degree in mathematics from the University of Seville, in 1992. In the past, he was a Lecturer and also a Teaching Assistant with the University of Barcelona. He is currently a Full Professor with the Department of Computer Science and Artificial Intelligence. He is also the Head of the Research Group on Natural Computing, University of Seville. From 2005 to 2007, he was a Numerary Member with the Academia Europaea (The Academy of Europe) in the Section of Informatics.