Hindawi Publishing Corporation
Discrete Dynamics in Nature and Society
Volume 2015, Article ID 123960, 11 pages
http://dx.doi.org/10.1155/2015/123960

Research Article

Research on P System with Chain Structure and Application and Simulation in Arithmetic Operation

Jing Luan and Zhong Yao

School of Economics and Management, Beihang University, Beijing 100191, China

Correspondence should be addressed to Zhong Yao; iszhyao@buaa.edu.cn

Received 19 August 2014; Accepted 24 December 2014

Academic Editor: Zhan Zhou

Considering the advantages of distribution and maximum parallelism of membrane computing and availability of discrete Morse theory to deal with discrete structure, in this paper, combining discrete Morse theory and membrane computing, a novel membrane structure—P system with chain structure, is proposed, which is constructed on the basis of discrete gradient vector path of the discrete Morse theory. At the theoretical level, due to its unique chain structure, compared with traditional P system, its structure, object, and rule are described in details. In the practical aspect, a specific application example, P chain system for arithmetic operation, is presented to demonstrate the superiority, computational efficiency, and ability of P system with chain structure. Moreover, a simulation system of arithmetic operations based on P chain system is designed, giving a visual display of the implementation of P chain system for arithmetic operation, and verifying the feasibility and effectiveness of P chain system.

1. Introduction

Membrane computing is a new computational model proposed by Romanian scientist Păun in 1998, due to being introduced by Păun at the first time, it is also called P system [1]. Membrane computing abstracts cell as the computational unit, permitting every computational unit to calculate dependently and the whole system to operate in the way of maximum parallelism, whose computational efficiency has been improved obviously [2–5]. It has been proved that the computation capacity of membrane computing is equivalent with that of turing machines, so due to its strong parallel computing power it has been the highlight of the recent study.

Morse theory [6] is a useful tool in differential topology, applied for investigating the topology of smooth manifolds, particular for computer graphics, having been the focus of the research. Forman [7] extended it to the discrete aspect, which provides an effective tool to describe the topology of discrete object, such as simplex and simplicial complex and plays a vital role in pure mathematics and applied mathematics. Concepts in discrete Morse theory, such as simplex [8], discrete gradient vector path and so on, provide a useful tool to research the topology of discrete structure.

Generally, the structure of P system is abstracted as the nested structure of the cell wall and organelle, so it is a kind of nested and hierarchical structure. Of course, there are many other structures, such as reticular formation of neural network. Up to now, there are three main P systems, cell-like P system, tissue-like P system and neural P system [3], and the study of cell-like P system has been well-developed. In literature [4], Păun pointed out that the focus of the next stage of membrane computing study was the nonhierarchical arrangement of membranes. In literature [9], the author has proposed a P system based on simplicial complexes, which is an innovative try in the aspect of nonhierarchical membrane structure. Inspired from this, in this paper, a P system with chain structure is introduced, which combines membrane computing with discrete Morse theory and constructs a P system based on the discrete gradient vector path, forming a new kind of nonhierarchical membrane structure—P chain system. To try to arrange a novel nonhierarchical structure of P system not only makes a contribution to knowledge but also makes a clear methodological contribution.

The organization of the reminder in this paper is described as follows: Section 2 is the part of theoretical discussion, reviewing theories and properties of discrete

Morse theory and P system, which are the foundation of P system with chain structure, and then giving the specific description of structure, object, and rule of P chain system. In Section 3, a practical application of P chain system, P chain system for arithmetic operation, is proposed, displaying $+, -, *, /$ four kinds of P chain systems. In Section 4, by a computer simulation, the specific implementation of P chain system for arithmetic operation is demonstrated, showing the computational efficiency and power of P system with chain structure. In Section 5, the summary and prospect are included.

2. P System with Chain Structure

2.1. Foundation of P System with Chain Structure

2.1.1. Discrete Morse Theory. Here are some core definitions in discrete Morse theory, which are also essentials in this paper [6–9].

Definition 1 (Q-simplex). Assuming that $a^0, a^1, a^2, \ldots, a^q$ are $q+1$ points which cover the widest position in n-dimensional Euclidean space and $q \le n$, that is to say, vectors $e^1 = a^1 - a^0$, $e^2 = a^2 - a^0, \ldots, e^q = a^q - a^0$ are linearly independent relationship with each other, then defining $x = \lambda_0 a^0 + \lambda_1 a^1 + \cdots + \lambda_q a^q$, written as $x = \sum_{i=0}^{k} \lambda_i \alpha^i$, and real number $\lambda_0, \ldots, \lambda_q$ satisfying two factors: (1) $\sum_{i=0}^{q} \lambda_i = 1$; (2) $\lambda_0, \lambda_1, \ldots, \lambda_q \ge 0$, the set of x in E^n is called q-simplex, marked as $\underline{s}^q$, while $a_0, a_1, a_2, \ldots, a_q$ are called vertices of the simplex. A simplex is uniquely denoted by its vertices; therefore it can be expressed as $[a_0, a_1, a_2, \ldots, a_q]$ or $a_0 a_1 a_2 \cdots a_q$ simply.

Definition 2 (simplex with orientation). For a q-simplex $\underline{s}^q$, there are $(q+1)!$ permutations of different sequences for its $q+1$ vertices $a_0, a_1, a_2, \ldots, a_q$; when $q > 0$, there are two kinds of permutations; any two permutations of the same kind differ in even commutations, while any two permutations in different group differ in odd commutations. These two kinds are called two orientations of $\underline{s}^q$. The simplex which has been given an orientation is called simplex with orientation, one denoted as s^q and the other as $-s^q$.

Definition 3 (Q-chain). Suppose a fundamental constituent $\{s_i^q\}$ of n-complex K, for an integer g_i, there is $g_i s_i^q = (-g_i)(-s_i^q)$, and a linear combination with integer coefficient $x_q = g_1 s_1^q + g_2 s_2^q + \cdots + g_{a_q} s_{a_q}^q$ is called q-chain of K.

Definition 4 (discrete gradient vector field). For the gradient vector field V on a n-complex K, it is a series of ordered pair sets denoted as $\{\alpha^{(p)}, \beta^{(P+1)}\}$ and subjected to (1) $\alpha \prec \beta$, (2) each simplex of K belongs to one pair in V at most. Specifically, a gradient vector field is called discrete gradient vector field if and only if there is no nontrivial closed V-path on it, which is an alternating sequence of simplexes $\alpha_0^{(p)}, \beta_0^{(p+1)}, \ldots, \alpha_r^{(p)}, \beta_r^{(p+1)}, \alpha_{r+1}^{(p)}$, where $\alpha_i^{(p)} \prec \beta_i^{(p+1)}$ and $\{\alpha_i^{(p)} \prec \beta_i^{(p+1)}\} \in V$.

2.1.2. P System Theory. Membrane computing is a novel computational model abstracted from biochemical reactions in living cells, whose merit is internal maximum parallelism. The essential components of P system are membrane structure, objects, and rules. The formalization definition of P system is shown as follows:

$$\prod = (O, T, C, u, w_1, \ldots, w_m, (R_1, \rho_1), \ldots, (R_m, \rho_m)). \quad (1)$$

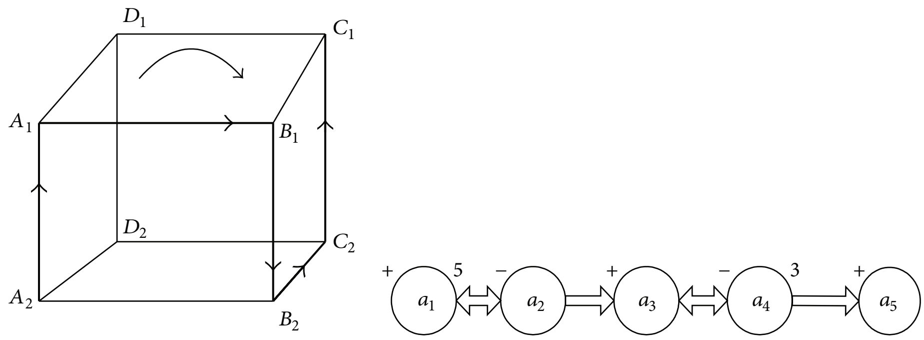
Here O is the alphabet, representing the object; T is the output alphabet and $T \in O$; C is the catalyst and $C \in O - T$; u is the set of membranes; w_i is the object multisets, i is the membrane label; R_i is the set of evolutionary rules, some rules are applied to reflect the chemical reactions, such as rewriting rules, and others are employed to simulate biological processes, such as communication rules, and ρ_i is the priority set of these rules.

2.2. Description of the P Chain System

2.2.1. Definitions and Properties of P Chain System

Definition 5 (P system with chain structure). Generalized P system with chain structure is written as $X^P = g_1 s_1^p + g_2 s_2^p + \cdots + g_{a_p} s_{a_p}^p$, where g_i is integer and $|g_i|$ represents the number of the membranes. Additionally, if $\forall g_i > 0$, X^P represents positive generalized P system with chain structure, especially if $\forall g_i = 1$, X^P denotes positive standard P chain system. Moreover, if $\forall g_i < 0$, X^P represents negative generalized P chain system, particularly if $\forall g_i = -1$, x^P denotes negative standard P chain system. Furthermore, if $\exists g_i > 0$ or $\exists g_i < 0$, X^P represents multiply generalized P chain system, especially if $\exists g_i = 1$ or $\exists g_i = -1$, X^P denotes multiply standard P chain system.

Generally, what we call P system with chain structure is referred to the standard P chain system. Here is an example for the above definition. There is a complex $A_1B_1C_1D_1 - A_2B_2C_2D_2$ in Figure 1(a), and 0-simplex are these vertices A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, 1-simplex are these segments A_1B_1, C_1D_1, A_2B_2, C_2D_2, …, 2-simplex are these surfaces $A_1B_1C_1$, $A_2B_2C_2$, …, 3-simplex are these bodies $B_2 - A_1B_1C_1$, $A_1 - A_2B_2D_2$ and so on. There is a discrete gradient vector path $A_2 \rightarrow A_2A_1 \rightarrow A_1 \rightarrow A_1B_1 \rightarrow B_1 \rightarrow B_1B_2 \rightarrow B_2 \rightarrow B_2C_2 \rightarrow C_2 \rightarrow C_2C_1 \rightarrow C_1$ on the 3-complex, composed by $S^0 = \{A_2, A_1, B_1, B_2, C_2, C_1\}$ and $S^1 = \{A_2A_1, A_1B_1, B_1B_2, B_2C_2, C_2C_1\}$, so there are 0-chain $A_2 + A_1 + B_1 + B_2 + C_2 + C_1$ whose units come from S^0 and 1-chain $A_2A_1 + A_1B_1 + B_1B_2 + B_2C_2 + C_2C_1$ whose units belong to S^1. And then membranes from 0-chain or from 1-chain can compose a P system with chain structure. $A_2 + A_1 + B_1 + B_2 + C_2 + C_1$ is an example of positive standard P chain system. $5A_2A_1 + (-A_1B_1) + B_1B_2 + (-3B_2C_2) + C_2C_1$ can be called multiply generalized P chain system shown in Figure 1(b), where two-way arrow represents the repeated appearance of membrane, figure denotes the number of repeated appearance of its front membrane, that is to say, a_1 appears 5 times and a_3 3 times, usually 1 is omitted.



(a) Complex $A_1B_1C_1D_1 - D_2A_2B_2C_2$

(b) P chain system based on discrete gradient vector path $A_2A_1 \rightarrow A_1B_1 \rightarrow B_1B_2 \rightarrow B_2C_2 \rightarrow C_2C_1$

Figure 1: The structure of P chain system.

Property 1 (oriented property of P chain system). There are three orientations of P chain system due to the orientation of simplex. What we entitle as P chain system with the same direction is that all the membranes are the same orientation, specifically in order to determine the direction of the P chain system, we need to predefine the orientation of complex, that is to say, if we stipulate the positive surface S^q as positive, all the membranes from positive simplex of $\{S^q\}$ form P chain system with positive orientation, marked by "+", its membrane called positive membrane. On the contrary, all the membranes from the negative simplex of $\{-S^q\}$ constitute the P chain system with negative orientation, marked by "−", corresponding membrane called negative membrane. Moreover, if there are membranes from positive simplex or negative simplex at the same time in a P chain system, it is called P chain system with multiply orientation, marked by "×," that is to say, in the P chain system with multiply orientation there are both positive membranes and negative membranes.

Take the P chain system $A_2A_1 \rightarrow A_1B_1 \rightarrow B_1B_2 \rightarrow B_2C_2 \rightarrow C_2C_1$ in Figure 1 for example. If we define clocklike orientation as the positive direction, P chain system $A_2A_1 \rightarrow A_1B_1 \rightarrow B_1B_2 \rightarrow B_2C_2 \rightarrow C_2C_1$ is positive one, and P chain system $A_1A_2 \rightarrow B_1A_1 \rightarrow B_2B_1 \rightarrow C_2B_2 \rightarrow C_1C_2$ is with negative orientation, while P chain system $A_2A_1 \rightarrow B_1A_1 \rightarrow B_1B_2 \rightarrow B_2C_2 \rightarrow C_1C_2$ is the example of multiply orientation.

Property 2 (additive property of P chain system). Supposing that $x_q = g_1s_1^q + g_2s_2^q + \cdots + g_{a_q}s_{a_q}^q$ and $y_q = h_1s_1^q + h_2s_2^q + \cdots + h_{a_q}s_{a_q}^q$ are any two q-chain, their addition $x_q + y_q = (g_1 + h_1)s_1^q + (g_2 + h_2)s_2^q + \cdots + (g_{a_q} + h_{a_q})s_{a_q}^q$ is also a q-chain. For the addition, all the q-chain of complex K are a free Abelian group, which is called q-chain group of K.

Property 3 (precursory and subsequent relationships between adjacent membranes in P chain system). Because P chain system is based on a discrete gradient vector path, the order of membranes is determined; that is, for a given P system with chain structure, the relationship between adjacent membranes is precursor or subsequent. For a P chain system $x^p = r_1\alpha_1^p + r_2\alpha_2^p + \cdots + r_n\alpha_n^p$, where r_i is integer and $i = 1, \ldots, n$, except the first membrane, usually the input membrane, for each membrane α_i^p where $i = 2, \ldots, n$, its ahead membrane α_{i-1}^p is called its precursor; the relationship between α_{i-1}^p and α_i^p is entitled as precursory relationship. Moreover, except for the last membrane, usually the output membrane, for arbitrary membrane α_i^p, where $i = 1, \ldots, (n-1)$, its back membrane α_{i+1}^p is called its subsequence; the relationship between α_i^p and α_{i+1}^p is entitled as subsequent relationship.

The algorithm for constructing the P system with chain structure is shown in Algorithm 1.

2.2.2. Description of Structure, Object, and Rule of P Chain System. For the structure of P chain system, we know that it is nonhierarchical, specifically if it is a chain structure based on discrete gradient vector path, while for the object, it is similar with the former P system, which is denoted by multisets, meaning that $w = a_1^{M(a_1)}a_2^{M(a_2)}\cdots a_n^{M(a_n)}$ is used to represent objects of each membrane. But considering the oriented property of P chain system, the membrane is divided into positive and negative one, so objects in membrane with different orientations are different too; for example, supporting that object in positive membrane is marked as a, then object in negative membrane is marked as a' correspondingly. a and a' are antimatter, meaning that they cannot coexist; when they encounter each other, they will counteract immediately. This is similar with positive and negative spikes in spiking neural P system, where rule $\alpha\alpha' \rightarrow \lambda$ makes their coexistence impossible.

Based on the rules of former P system, combining the particularities of P system with chain structure, there are three main rules of P chain system: rewriting rules, communication rules, and forgetting rules.

Rewriting rule is with the form of $u \rightarrow v$, where u and v, from the alphabet O, are the objects which represent the multisets, and this rule can be used in membrane a when and only when the object set w in a satisfies $w \supseteq u$. Rewriting rules are used to control the type and number

INPUT: discrete Morse function M, the number of the membranes in P chain system N
OUTPUT: P system with chain structure
(1) Find out a discrete gradient vector path from the given discrete Morse function M by the algorithm proposed in literature [10];
(2) Find out sets $\underline{s}^q$ consisted by all the q-simplex and $\underline{s}^{q-1}$ consisted by all the $(q-1)$-simplex from the discrete gradient vector path, noting that $\underline{s}^q$ and $\underline{s}^{q-1}$ are ordered sets, ordered by the sequence of simplexes located in the discrete gradient vector path;
(3) According to the number of membranes N in the P chain system, choosing adjacent N simplexes from $\underline{s}^q$ or $\underline{s}^{q-1}$ as membranes in the P chain system, constituting q-dimension or $(q-1)$-dimension P system with chain structure.

Algorithm 1: Construction algorithm of P system with chain structure.

of objects in membrane. Here are some target indications used to manage the movement of objects. They include $\text{tar}_1 = \{\text{here}, \text{out}, \text{in}\}$, where "here" indicates that object remains in the membrane, "out" denotes that object leaves the membrane into its subsequence, and "in" shows that object is sent out to its precursor. So rewriting rule with target indication is formed as $u \rightarrow v$, where u is a string representing multisets of objects from a given set O, and v is a string over O^* $\{\text{here}, \text{out}, \text{in}\}$. Usually they are formed as (a, tar), where a is the object from O and tar is here (usually omitted) or in or out. For example, P chain system $x^P = r_1\alpha_1^P + r_2\alpha_2^P + r_3\alpha_3^P + r_4\alpha_4^P$; there are object $a^3b^2c^4$ and rule $a^2bc^3 \rightarrow a^2bc(da, \text{out})(ca, \text{in})$ in membrane a_2; the result of the execution of the rule is to produce $4a, 1b, 2c, 1d$, where $1d$ and $1a$ will be sent into subsequence a_3 and $1c$ and $1a$ will be sent into precursor a_1, eventually remaining $a^3b^2c^2$ in membrane a_2.

Communication rules are used to manage the communication across the membranes, including symport rules and antiport rules. Here another target indication $\text{tar}_2 = \{\text{pre}, \text{sub}\}$ is introduced, where "pre" indicates that which move to the precursor of the membrane and "sub" denotes that which move to its subsequence. So in communication rules, target indication is shown in pair, formed as $(\text{tar}_1, \text{tar}_2)$, where $\text{tar}_1 = \{\text{here}, \text{out}, \text{in}\}$ and $\text{tar}_2 = \{\text{pre}, \text{sub}\}$ and where tar_2 can be omitted, meaning that objects will be sent into the precursor or subsequence randomly. In a P chain system where the structure is determined, defining that the membrane where rule resides is α_i^P, its precursor is α_{i-1}^P, and its subsequence is α_{i+1}^P. Symport rules, formed as (u, in) or $(u, (\text{in}, \text{pre}))$ or $(u, (\text{in}, \text{sub}))$, express that object u from membrane α_{i-1}^P or membrane α_{i+1}^P is sent into membrane α_i^P, "pre" or "sub" specifying the source of object u. Also there is the form of (u, out) or $(u, (\text{out}, \text{pre}))$ or $(u, (\text{out}, \text{sub}))$, which means that object u from membrane α_i^P is sent out into membrane α_{i-1}^P or membrane α_{i+1}^P randomly or determined by the target "pre" or "sub". Antiport rules is formed as $(u, \text{out}; v, \text{in})$ or $(u, (\text{out}, \text{tar}_2); v, (\text{in}, \text{tar}_2))$, where $(u, \text{out}; v, \text{in})$ means that object u from membrane α_i^P is sent out into membrane α_{i-1}^P or membrane α_{i+1}^P randomly, at the same time, object v from the membrane α_{i-1}^P or membrane α_{i+1}^P is sent into membrane α_i^P, while rule $(u, (\text{out}, \text{tar}_2); v, (\text{in}, \text{tar}_2))$ elaborates the specific precursor and subsequence. In fact, the target indication $(\text{tar}_1, \text{tar}_2)$ can be simplified as $\text{tar}_{1\text{tar}_2}$, for example, symport rule $(u, (\text{out}, \text{sub}))$ in membrane α_i^P can be expressed as $(u, \text{out}_{\alpha_{i+1}^P})$, in the same way, antiport rule $(u, (\text{out}, \text{tar}_2); v, (\text{in}, \text{tar}_2))$ can be shown as $(u, \text{out}_{\alpha_{i+1}^P}; v, \text{in}_{\alpha_{i-1}^P})$, which means that object v from precursory membrane α_{i-1}^P is sent into membrane α_i^P, at the same time object u from membrane α_i^P is sent out into subsequent membrane α_{i+1}^P.

Forgetting rule is defined as the form of $u \rightarrow \lambda$, where u is object over alphabet O which is used to represent multisets, and λ is null. The function of forgetting rule is to disappear some certain objects in membrane. A typical example of forgetting rule is objects a, a' which are antimatter in membranes with different orientation, when they encounter with each other, they will counteract immediately, resulting from the execution of rule $aa' \rightarrow \lambda$. In most situations, forgetting rule $aa' \rightarrow \lambda$ does not appear explicitly, but is defaulted to execute with the highest priority. Note that when execute this rule, if there are object a or a' in membrane where the rule resides, its objects a or a' will be used at first, if there is no object a or a', supporting that there are enough object a or a' in environment can be used to counteract the antimatter.

2.3. Formalization Definition of P Chain System. The formalization definition of standard P system with chain structure is described as followed:

$$\prod = (O, T, C, u, \sigma_0, \sigma_1, \ldots, \sigma_m, \text{syn}, \text{in}, \text{out}). \qquad (2)$$

(1) O is the alphabet, and the element from O is called object;

(2) T is output alphabet, and $T \in O$;

(3) C is catalyst ($C \in O - T$); the element of C is unchanged during the evolutionary process of system; that is to say, the char of C does not disappear and there is no new char generated, but its participation is necessary for some certain rules to execute;

(4) u is the set of membrane structures, each membrane and its enclosed area are expressed as label set H, $H = \{1, 2, \ldots, m\}$;

(5) $\sigma_1, \ldots, \sigma_m$ denote that there are m membranes with chain structure $\sigma_1, \ldots, \sigma_m$, and membrane σ_i is represented as $\sigma_i = (n_i, R_i)$, $0 \le i \le m$, where

① $n_i \ge 0$ denotes the number of objects of membrane σ_i at the beginning of calculation;

② R_i shows the limited set of rules in membrane σ_i, there are three main forms.

(i) Rewriting rules are showed as $u \to v$, where u from the alphabet O represents the multisets; v is the string over set O^*\{here, out, in\}, written as (a, tar), where a is the object from O and tar is here (usually omitted), out or in.

(ii) Communication Rules include symport rules and antiport rules. Symport rules are formed as $(u, (\mathrm{tar}_1, \mathrm{tar}_2))$, and antiport rules are showed as $(u, (\mathrm{tar}_1, \mathrm{tar}_2)$; $v, (\mathrm{tar}_1, \mathrm{tar}_2))$, where u, v are the objects of alphabet O representing the multisets, and $\mathrm{tar}_1 = \{\mathrm{here}, \mathrm{out}, \mathrm{in}\}$, $\mathrm{tar}_2 = \{\mathrm{pre}, \mathrm{sub}\}$, tar_1 is used to control the direction of object in or out of the membrane, and tar_2 explains the object will be sent into which membrane.

(iii) Forgetting Rules are formed as $u \to \lambda$, where u is the object over O representing the multisets, and λ represents null. The execution of forgetting rules makes the number of objects in membrane less.

(6) $\mathrm{syn} = \{1, 2, \ldots, m\}$ is used to show the relationship among membranes in P chain system, for each $1 \le i \le m$, there is $\sigma_1 \to \sigma_2 \to \cdots \to \sigma_m$;

(7) $\mathrm{in}, \mathrm{out} \in \{1, 2, \ldots, m\}$ denote the input and output membranes.

3. P Chain System for Arithmetic Operation

The calculating power of P system has been concerned and researched generally, and previous study has proved that NP-hard problem can be resolved by P system in polynomial time [5]. So there are crucial theoretical value and heightened practical significance if we can take advantage of the maximum parallelism of P system to improve computational efficiency for a series of calculation problems, such as arithmetic operation. The fundamental arithmetic operation includes addition, subtraction, multiplication, and division, which is the base of other complex operations. We try to achieve arithmetic operation with P chain system in order to explore more complex operation fulfilled by P chain system. Literature [11] has proved the possibility to fulfill arithmetic operation by P system. Based on this, P chain system for arithmetic operation is proposed here; compared with the former method, there are some improvements of both time performance and space performance.

When designing the P chain system for arithmetic, standard P system with chain structure is chosen; that is to say, each membrane in P chain system appears only once. Moreover, we suppose that there are enough objects in environment which can be used for the execution of certain rules. Note that the P chain system for arithmetic operation designed now can only achieve the arithmetic operation between any two natural numbers, and the result of the calculation is shown as the number of object in output membrane.

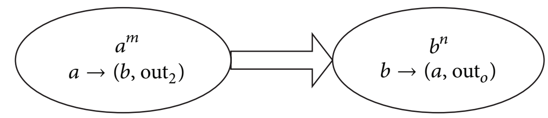


Figure 2: P chain system for addition.

3.1. P Chain System for Addition. In order to fulfill addition by P chain system, we select P system with single direction; that is to say, there is only membrane with positive orientation or negative orientation. Here is the example of P chain system with positive orientation to carry out the addition operation of P chain system. P system with chain structure for addition is designed as Figure 2, and its formalization expression is described as follows:

$$\prod{}^{+}(\mathrm{Addition}) = \left(O, \sigma_1, \sigma_2, \mathrm{in}, \mathrm{out}\right). \quad (3)$$

+: show that it is a P chain system with positive orientation; $O = \{a, b\}$; $\sigma_1 = \{n_1, R_1\}$, $\sigma_2 = \{n_2, R_2\}$; $n_1 = \{m \mid a\}$, $n_2 = \{n \mid b\}$; $R_1 = \{a \to (b, \mathrm{out}_2)\}$, $R_2 = \{b \to (a, \mathrm{out}_o)\}$; in $= 1$, out $= o$, here output membrane does not appear in Figure 2 and is represented by alphabet o.

As shown in Figure 2, there are two membranes with positive orientation, where m and n represent two natural numbers to add, and a and b are the objects of the P chain system, using the number change of object a and b to achieve the addition between any two natural numbers, here the number of object a are used to represent the calculating result. The specific process of P system with chain structure to fulfill addition is stated as followed. First, due to that there are operable rules in both membrane σ_1 and σ_2, in membrane σ_1, rule R_1 is executed and there will generate an object b sent into membrane σ_2, at the same time, rule R_2 in membrane σ_2 is performed too and produces an object a sent into output membrane. Reactions in membrane σ_1 and σ_2 are done at the same time in the mode of maximum parallelism, and it will continue until all the object a in membrane σ_1 are transformed into b and sent into membrane σ_2. Then membrane σ_1 will go into stable condition, and in membrane σ_2 the rules will be used until all object b from membrane σ_1 or σ_2 are converted into a and sent into output membrane. Now membrane σ_2 reaches stable, meaning that the whole P system is stable and the calculation ends, left $m+n$ object a in output membrane. Until then, the P chain system completes the addition $m+n$ of arbitrary two natural numbers m and n.

3.2. P Chain System for Subtraction. In order to achieve subtraction by P chain system, we select P chain system with multiplied orientation; that is, there are both membrane with

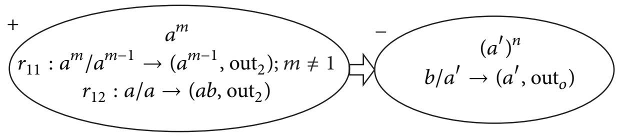


Figure 3: P chain system for subtraction.

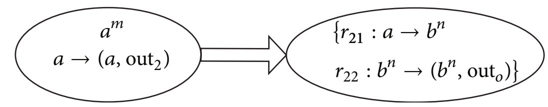


Figure 4: P chain system for multiplication.

positive orientation and negative orientation. P chain system for subtraction is designed as Figure 3 and its formalization expression is described as followed.

$$\prod{}^{\times} (\text{Subtraction}) = (O, \sigma_1, \sigma_2, \text{in}, \text{out}) \tag{4}$$

$\times$: show that it is a P chain system with multiply orientation; $O = \{a, a', b\}$; $\sigma_1 = \{n_1, R_1\}$, $\sigma_2 = \{n_2, R_2\}$; $n_1 = \{m \mid a\}$, $n_2 = \{n \mid a'\}$; $R_1 = \{\{r_{11} : a^m/a^{m-1} \to (a^{m-1}, \mathrm{out}_2), m \neq 1\}$; $\{r_{12} = a/a \to (ab, \mathrm{out}_2)\}\}$; $R_2 = \{b/a' \to (a', \mathrm{out}_0)\}$; in $= 1$, out $= o$, here output membrane isn't shown in Figure 3 and is represented by alphabet o.

As Figure 3 showed, there is one membrane with positive orientation and the other is with negative orientation, where m and n represent any two natural numbers to subtract, specifically m is subtrahend and n is minuend, and b, a and a' which are antimatter are the objects in the P chain system, using the number variation of object a and a' to achieve the subtraction between any two natural numbers, here the number of object a' are used to represent the calculating result. The process of P system with chain structure to achieve subtraction is described as followed. At first, rule in membrane σ_1 with positive orientation will be used, where "/" denotes the condition for the rule to be executed, that is, only when the condition is met, the rule can be used. So rule r_{11} will be performed, producing $m - 1$ object a sent into membrane σ_2 with negative orientation, and when a and a' encounter, they will disappear immediately by the defaulted rule $aa' \to \lambda$. And then there is only one object a left in membrane σ_1, which satisfies the executive condition of rule r_{12}, and objects a and b generated will be sent into membrane σ_2. The appearance of object b activates the rule R_2, and $n - m$ object a' left in membrane σ_2 will be sent into output membrane. Until then, the P chain system completes the subtraction $n - m$ of any two natural numbers m and n.

3.3. P Chain System for Multiplication. In order to complete multiplication by P chain system, we select P chain system with single orientation, membrane with positive orientation or negative orientation. Here take the P chain system with positive orientation for example, and P chain system for multiplication is designed as Figure 4 and its formalization expression is described as followed.

$$\prod{}^{+} (\text{Multiplication}) = (O, \sigma_1, \sigma_2, \text{in}, \text{out}) \tag{5}$$

$+$: show that it is a P chain system with positive orientation; $O = \{a, b\}$; $\sigma_1 = \{n_1, R_1\}$, $\sigma_2 = \{n_2, R_2\}$; $n_1 = \{m \mid a\}$, $n_2 = \{0 \mid \phi\}$; $R_1 = \{a \to (a, \mathrm{out}_2)\}$; $R_2 = \{r_{21} : a \to b^n; r_{22} : b^n \to (b^n, \mathrm{out}_0)\}$; in $= 1$, out $= o$, here output membrane does not appear in Figure 4 and is represented by alphabet o.

As shown in Figure 4, there are two membranes with positive orientation, where m and n represent any two natural numbers to multiply, and a and b are the objects of the P chain system, using the number change of a and b to achieve the multiplication between any two natural numbers, here the number of object b are used to represent the calculating result. The process of P system with chain structure to fulfill multiplication is explained as followed. First, the existence of object a needed by rule R_1 makes it usable in membrane σ_1, due to the execution of rule $a \to (a, \mathrm{out}_2)$, the object a produced will be sent into membrane σ_2, which will trigger rule r_{21} in membrane σ_2. Here rule R_2 is a chained rule, which contains two rule vectors. Referred to the literature [12], chained rule R is a vector set of rules, in the membrane if the first rule in the vector of chained rules R is applied, in the next step the rest of the rules from R will be applied in order in consecutive steps. And if any one rule from a vector of chained rules R which has already started to carry out cannot be used, the execution of R is dropped, that is, for the current application of R, the remaining rules are not executed anymore. So by rule r_{21}, generate n object b, and then the executive condition of rule r_{22} is satisfied. So in membrane σ_2 rule r_{22} will be used continuously until m object a in membrane σ_1 are used up, and the whole system reaches stable, calculation ending and $m * n$ object b left in output membrane. Until then, the P chain system completes the multiplication $m * n$ of arbitrary two natural numbers m and n. Specific implementation is shown in Table 1.

3.4. P Chain System for Division. In order to achieve division by P chain system, we use P chain system with single orientation. Here taking the P chain system with positive orientation for example, P chain system for division is designed as Figure 5 and its formalization definition is described as followed.

$$\prod{}^{+} (\text{Division}) = (O, \sigma_1, \sigma_2, \text{in}, \text{out}) \tag{6}$$

$+$: show that it is a P chain system with positive orientation; $O = \{a, b\}$; $\sigma_1 = \{n_1, R_1\}$, $\sigma_2 = \{n_2, R_2\}$; $n_1 = \{m \mid a\}$, $n_2 = \{0 \mid \phi\}$; $R_1 = \{a^n \to (a^n, \mathrm{out}_2)\}$, $R_2 = \{r_{21} : a^n \to b; r_{22} : b \to (b, \mathrm{out}_o)\}$; in $= 1$, out $= o$, here output membrane does not appear in Figure 5 and is represented by alphabet o.

As shown in Figure 5, there are two membranes with positive orientation, where m and n represent any two natural numbers to divide, and a and b are the objects of the P chain system, using the number variation of object a and b to achieve the division between any two natural numbers, here the number of object b are used to represent the calculating result. Because at present the object defined in P chain system can only represent nonnegative integer, division taken into consideration here ignores the remainder. The process of P system with chain structure to fulfill division m/n is explained

Table 1: The implementation of P chain system for multiplication.

Step	Membrane σ_1 Start	Membrane σ_1	Membrane σ_1 End	Membrane σ_2 Start	Membrane σ_2	Membrane σ_2 End	Output
1	a^m	$a \to (a, \mathrm{out}_2)$	a^{m-1}	$a(\sigma_1)$		$a(\sigma_1)$	
2	a^{m-1}	$a \to (a, \mathrm{out}_2)$	a^{m-2}	$a(\sigma_1)$	$a \to b^n$	b^n	
3	a^{m-2}	$a \to (a, \mathrm{out}_2)$	a^{m-3}	$a(\sigma_1)$ b^n	$a \to b^n$ $b^n \to (b^n, \mathrm{out})$	b^n	b^n
				$\vdots$			
m	a	$a \to (a, \mathrm{out}_2)$	ϕ	$a(\sigma_1)$ b^n	$a \to b^n$ $b^n \to (b^n, \mathrm{out})$	b^n	$b^{(m-2)n}$
$m+1$				$a(\sigma_1)$ b^n	$a \to b^n$ $b^n \to (b^n, \mathrm{out})$	b^n	$b^{(m-1)n}$
$m+2$				$a(\sigma_1)$ b^n	$a \to b^n$ $b^n \to (b^n, \mathrm{out})$	b^n	b^{mn}

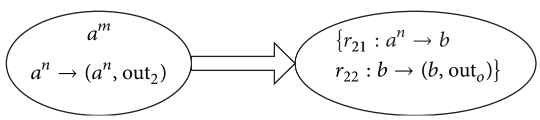


Figure 5: P chain system for division.

as followed. First, there are operable objects for rule R_1 in membrane σ_1, so the rule $a^n \to (a^n, \mathrm{out}_2)$ is used, generation n object a sent into membrane σ_2, which activates the rule r_{21}. Here rule R_2 is a chained rule too. So by rule r_{21} produce one object b, and then the executive condition of rule r_{22} is met. So in membrane σ_2 rule r_{22} will be performed continuously until the object a is used up or less than n and rule R_1 isn't executed any more, the whole system reaches stable, calculation ending and result is the number of object b left in output membrane. Until then, the P chain system completes the division m/n of arbitrary two natural numbers m and n. Specific implementation of exact division is shown in Table 2, and Table 3 gives the computational process of an example of nonexact division 9/2.

3.5. The Analysis of System Performance. Time complexity used in literature [11] is referred to the number of execution of all rules. Following this way, the comparison of time complexity of the two different methods is shown in Table 4. We can see that the time complexity of arithmetic operation fulfilled by P system with chain structure has been improved obviously, especially the division, which can testify the efficiency and improvement of the new method. Meanwhile, the design of membrane structure and rule is simpler, more effective and more conducive to simulate than the former P system.

4. Simulation of Arithmetic Operation P Chain System

The simulation of P system has a vital practical significance, and there has been a series of success P system simulation software which can be obtained from P system web page [13]. Based on this, in this paper we try to do the simulation program of P chain system for arithmetic operation, as a result, it has stood out its lower time-cost and higher efficiency significantly. The development platform of the system is Microsoft Visual Studio 2008 on Windows 7 and the development language is C#. The whole system can operate normally, and can produce the executable file, which is with portability and robustness.

4.1. Description of Storage Mode of the Simulation System. During the process to develop system, we need to choose the appropriate storage mode for different elements, and the specification is demonstrated as followed [14].

We have known that we need to store the membrane structure, object and rule. Here we abstract those contents in each P system instance as an input file, marked "pcs" as the expanded name, for example, the input file of a P system to achieve the addition operation is named as "Addition.pcs". And taking the addition operation for example, the input file which includes the definition of the membrane structure, object and rule is showed in Table 5.

After explain the structure, object and rule in a P chain system which need stored, how to store them is the next consideration, which is also the problem that needs to be resolved in the process of system initialization.

For the object multisets in the P chain system, it is stored by the form of nonnegative integer. The reason that we do not choose char is when the number of the object is large, the length of char may be very long, not only occupying large storage space, but also consuming much time when read. However, by using the INT array, huge amount of data can be denoted briefly.

For the rules in the P chain system, they are input into the system in the form of strings. If we want to fulfill the correspondence between the rules and object multisets, some rule transformations are needed. The way is that, first, determine the alphabets used in the P chain system, taking the P chain system for addition operation for example, the alphabet set $\{a, b\}$ and the rule $a \to (b, \mathrm{out}_2)$ involved in membrane 1, we need divide the rule into two parts, namely the former and the latter, and store them respectively.

Table 2: The implementation of P chain system for exact division.

Step	Membrane σ_1 Start		End	Membrane σ_2 Start		End	Output
1	$a^{m-(t-3)n}$	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^{m-n}	$a^n(\sigma_1)$		$a^n(\sigma_1)$	
2	a^{m-n}	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^{m-2n}	$a^n(\sigma_1)$	$a^n \rightarrow b$	b	
3	a^{m-2n}	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^{m-3n}	$a^n(\sigma_1)$ b	$a^n \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	b
4	a^{m-3n}	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^{m-4n}	$a^n(\sigma_1)$ b	$a^n \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	b^2
				$\vdots$			
$t-2$ $(t = m/n)$	$a^{m-(t-3)n}$	$a^n \rightarrow (a^n, \mathrm{out}_2)$	$a^{m-(t-2)n}$	$a^n(\sigma_1)$ b	$a^n \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	$b^{(t-4)}$
$t-1$ $(t = m/n)$	$a^{m-(t-2)n}$	$a^n \rightarrow (a^n, \mathrm{out}_2)$	$a^{m-(t-1)n}$	$a^n(\sigma_1)$ b	$a^n \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	$b^{(t-3)}$
t $(t = m/n)$	$a^{m-(t-1)n}$	$a^n \rightarrow (a^n, \mathrm{out}_2)$	$a^{m-tn} = \phi$	$a^n(\sigma_1)$ b	$a^n \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	$b^{(t-2)}$
$t+1$ $(t = m/n)$	ϕ		ϕ	$a^n(\sigma_1)$ b	$a^n \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	$b^{(t-1)}$
$t+2$ $(t = m/n)$	ϕ		ϕ	b	$b \rightarrow (b, \mathrm{out}_0)$	b	b^t

Table 3: The implementation of P chain system for nonexact division 9/2.

Step	Membrane σ_1 Start		End	Membrane σ_2 Start		End	output
1	a^9	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^7	$a^2(\sigma_1)$		$a^2(\sigma_1)$	
2	a^7	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^5	$a^2(\sigma_1)$	$a^2 \rightarrow b$	b	
3	a^5	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^3	$a^2(\sigma_1)$ b	$a^2 \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	b
4	a^3	$a^n \rightarrow (a^n, \mathrm{out}_2)$	a^1	$a^2(\sigma_1)$ b	$a^2 \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	b^2
5	a^1		a^1	$a^2(\sigma_1)$ b	$a^2 \rightarrow b$ $b \rightarrow (b, \mathrm{out}_0)$	b	b^3
6	a^1		a^1	b	$b \rightarrow (b, \mathrm{out}_0)$	b	b^4

The former part of the rule is presented as $[1, 0]$, meaning that there are 1 object a and 0 object b, and the latter part of the rule is denoted as $[0, 0; 1]$ and $[0, 1; 2]$, noting that the number 1 and 2 after semicolon display the membrane 1, 2, and the part before semicolon shows the change of numbers of objects. For the rule $a \rightarrow (b, \mathrm{out}_2)$, we know that after the operation of the rule, there are 1 object a consumed and no consumption of object b in membrane 1, and in membrane 2, the number of object a are unchanged and the object b is added one more.

For the storage of the structure of P chain system, due to the special chain structure, we choose linear structure to save. Specifically, it is a kind of linear list, that is to say, two elements which are adjacent logically are adjacent physically. Here we use the vector representation, and there are three marks in each storage unit, the node mark (I), precursor node mark (Pre) and subsequent node mark (Sub). Take the P chain system $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow O$ for example, its storage form showed as Table 6.

Following is the code of object and rule for P chain system for addition operation to illustrate the specific storage mode.

(1) int[]$Ob21$ = new int []{num1};

(2) int[]$Ob22$ = new int []{num2};

(3) String[,]$R21$ = new String[,]{{1, 0}, {0, 0}, {1}};

(4) String[,]$R22$ = new String[,]{{0, 0}, {0, 1}, {2}}.

Here line (1) and (2) are the storage way of objects, num1 and num2 are any two nature numbers to add; line (3)-(4) are the storage of rule $a \rightarrow (b, \mathrm{out}_2)$, using two-dimension string array, where the first dimension shows the rule is divided to three parts, and the second dimension displays the number of object a, object b and the label of membrane, for line (3) as an example, the last {1} means this is in the membrane 1, the {1, 0} denotes before rule execution there are 1 object a and 0 object b, and {0, 0} explains after rule execution there are 0 object a and b left the membrane 1.

4.2. Description of Rules Selection Algorithm. The strong computing power of membrane computing is largely due to its maximum parallelism, in the theoretical aspect, the rules in the membrane are allowed to perform freely until there are no rules can be selected. But in the specific process to fulfill the simulation of P system, we need to consider the order

Table 4: Comparison of time complexity of two methods.

Time complexity	Addition	Subtraction	Multiplication	Division
Literature [11]	$O(n+m+1)$	$O(n+m+1)$	$O(m)$	$O(n+m+q+1+1)$
This paper	$O(\max\{m,n\})$	$O(n)$	$O(m+2)$	$O(m/n+2)$

Table 5

1	Start	leading flag
2	$1 \rightarrow 2 \rightarrow O$	membrane
3	O	output membrane label
4	$a\ b$	multisets
5	a	output multisets
6	Init m_1	initialization membrane *1*
7	Init M	initialization the multisets in membrane *1*
8	*8 0*	8 a, 0 b in multisets
9	Init R	initialization the rule sets in membrane *1*
10	$a \rightarrow (b, \text{out}_2)$	the *0*th rule in membrane *1*
11	Prio R	the priority of the rules in membrane *1*
12	—	the null priority set
13	Init m_2	initialization membrane *2*
14	Init M	initialization the multisets in membrane *2*
15	0 6	0 a, 6 b in multisets
16	Init R	initialization the rule sets in membrane 2
17	$b \rightarrow (a, \text{out}_0)$	the *0*th rule in membrane *2*
18	Prio R	the priority of the rules in membrane *2*
19	—	the null priority set
20	Init m_O	initialization output membrane O
21	Init M	initialization the multisets in membrane O
22	0 0	0 a, 0 b in multisets
23	Init R	initialization the rule sets in membrane O
24	—	the null rule in membrane O
25	Prio R	the priority of the rules in membrane O
26	—	the null priority set
27	End	ending flag

Table 6: Storage of P chain system.

Storage position	0	1	2	3	4	5
Node mark (I)	*1*	*2*	*3*	*4*	*5*	*O*
Precursor node mark (pre)	*#*	*1*	*2*	*3*	*4*	*5*
Subsequent node mark (sub)	*2*	*3*	*4*	*5*	*O*	*#*

of the rule executed and the phenomenon "competition for resource" among the implement process of rules.

For the order of the rules execution, there are many exploratory researches, for example, in the literature [12] chained-rule was proposed, which is a new way to design rules in the P system, where rules are arranged as a form of a chain, meaning they are used by a sequence in the chain. Supposing that there is a chained rule R, which contains several vectors of the rule $r_1, r_2, \ldots, r_n$, in the membrane if the first rule r_1 of chained rule R is applied, then in the next step the rest of the rules from R will be applied in order in consecutive steps. Only when the condition of certain vector rule r_i is not satisfied or there is no more vector rules can be used, the chained rule R will halt.

In fact, all the ways to explore the order of rule execution are to solve the priority problem, here are two main rule priorities, the strong relationship and the weak one. The strong relationship is referred to that a rule can be used when and only when there is no prior rule can be used, while the weak relationship is defined as for a rule R, if there are prior rules than it, meaning the condition to perform the prior is also met, no matter the prior rules are used or not, the rule R cannot be used. For example, for the objects $a\ b\ c$ and rules 1: $ab \rightarrow ac$; 2: $ab \rightarrow bc$; 3: $a \rightarrow b$; 4: $b \rightarrow c$ with priority $2 > 3$ and $2 > 4$, referring to the definition of strong relationship, there are three possible rule choices $\{1,3\}$, $\{1,4\}$ and $\{2\}$, but considering the weak relationship, there are only two choices $\{1\}$ and $\{2\}$, and this is because the condition that is satisfied to rule 1 is also met to rule 2, even if rule 2 is not used, when rule 1 is executed, rule 3 and 4 cannot be used. In this paper, we choose the weak one as the standard in the simulation of P chain system. And also some tentative algorithms are given to solve the problem of "competition of resource". The algorithm of rule selection for simulating P chain system is shown in Algorithm 2.

Now we have set the executive priority of rules, which has reduced the problem of "competition for resource" in some extend, but for the rules without priority, the problem is also existed, here we make full use of another significant characteristic of P system—"Randomness", meaning that if there are two rules which compete for resource, randomly select one to use. The specific algorithm to handle "competition for resource" further is shown in Algorithm 3.

4.3. Demonstration of Simulation of P Chain System. The interface of simulation software of P chain system for implementing arithmetic operation is shown in Figure 6(a). Here the first number and the second number represent the two positive integers used to operate, and there are four kinds of operators $+, -, *, /$ to be chosen. When you have chosen a kind of operator and clicked on the "calculate" button, the calculating result will be shown in the resulting box, and when you have selected to use clear button, the data will be eliminated from the textbox and the next calculation is ready. Figures 6(b), 6(c), 6(d), 6(e), and 6(f) show the realization to calculate $2+9$, $17-9$, $9*22$, $98/14$, $10/3$ is possible.

5. Summary and Prospect

In this paper, a novel P system with chain structure is introduced, which combines membrane computing and discrete Morse theory and takes use of the advantages of discrete

INPUT: object multisets of membrane M, rule set R, priority set of rule execution P (P is represented by the ordered pairs of rule labels, such as $1 > 2$, denoted as $\langle 1, 2\rangle$; for the chained rules, P is showed as $\langle ic, i(c+1)\rangle$, $i, c = 1, 2, \ldots$, meaning to be performed by order);
OUTPUT: rule set chosen to be used A
(1) For every rule in the set R, if each value in the INT vector corresponded to the former of the rule is no more than corresponding value in M, rule r is put into the candidate rule set L.
(2) For each ordered pairs $\langle p, q\rangle$ in priority set P, setting $A = L$, if rule r_p and r_q are both in L, then if $p = ic$, $q = i(c+1)$, then r_p, r_q are left in A, and the sequence is r_p, r_q, otherwise delete the less prior rule r_q.

Algorithm 2: Rule selection algorithm for simulation of P chain system.

INPUT: rule set A (the output of Algorithm 2)
OUTPUT: rule set L_p without competition for resource
(1) Select one rule r form A randomly, adding r into L_p, and delete r from A at the same time.
(2) For every rule r' in A except r, if there is competition for resource between r' and r, delete r' from A. If A is not null, return (1), else output L_p.

Algorithm 3: Rule selection algorithm with "competition for resource".

(a) Simulation interface (b) "+"

(c) "−" (d) "∗"

(e) "/" (exact division) (f) "/" (aliquant division)

Figure 6: Simulation interface of P chain system for arithmetic operation.

gradient vector path. As one of new P system, it has some theoretical and practical contributions.

First, the definition of structure, object and rule of P system with chain structure is proposed, especially its differences with the former P system, mainly referred to its chain structure, such as its oriented property, additive property, and precursory and subsequent relationships. It contributes to enrich the theoretical framework of P system research. Second, the application on arithmetic operation by P chain system is demonstrated, and the specific model and rules of P chain system used are shown in detail. Compared with the former methods, the new thinking is more effective both in time and space complexity. Last, the simulation system of P chain system for arithmetic operation is given, and the feasibility of the system is proved, which demonstrates the computational feasibility and effectiveness of P system with chain structure. This is a good start to simulate P system, and is also an exploratory way to show and take full use of the advantages of P system to solve some practical problems.

While we should admit that there are some insufficiencies in this paper; for example, the application on arithmetic operation is not connected with the rules proposed so closely and does not display the computing power of P chain system obviously, and the function of the simulation system is a little easy. All of these should be taken into consideration for further study. Based on the work in this paper, there are many other studies can be furthered on. For example, there are some special properties of P chain system, and how to use these particularities to enhance the computational power of P system is a question deserved to think. Also, whether there are some special applications which may be solved by P chain system more efficiently or not is a meaningful question too. Moreover, the simulation of P system with chain structure to fulfill more functions is a big challenge and needed to study more.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work has been supported by the Natural Science Foundation of China (Projects nos. 71271012, 71071006, and 71332003).

References

[1] G. Păun, "Membrane computing," in *Fundamentals of Computation Theory*, vol. 2751, pp. 284–295, Springer, Berlin, Germany, 2003.

[2] G. Păun, "Computing with membranes," *Journal of Computer and System Sciences*, vol. 61, no. 1, pp. 108–143, 2000.

[3] G. Păun, G. Rozenberg, and A. Salomaa, *Handbook of Membrane Computing*, Oxford University Press, Oxford, UK, 2009.

[4] G. Păun, "An introduction to and an overview of membrane computing," *The Journal of Logic and Algebraic Programming*, vol. 79, no. 6, pp. 291–294, 2010.

[5] G.-X. Zhang and L.-Q. Pan, "A survey of membrane computing as a new branch of natural computing," *Chinese Journal of Computers*, vol. 33, no. 2, pp. 208–214, 2010.

[6] J. Milnor, *Morse Theory*, Princeton University Press, Englewood Cliffs, NJ, USA, 1963.

[7] R. Forman, "A user's guide to discrete Morse theory," *Seminare Lotharinen de Combinatore*, vol. 48, pp. 1–35, 2002.

[8] M. K. Chari and M. Joswig, "Complexes of discrete Morse functions," *Discrete Mathematics*, vol. 302, no. 1–3, pp. 39–51, 2005.

[9] X. Liu and A. Xue, "Communication P systems on simplicial complexes with applications in cluster analysis," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 415242, 17 pages, 2012.

[10] H. King, K. Knudson, and N. Mramor, "Generating discrete Morse functions from point data," *Experimental Mathematics*, vol. 14, no. 4, pp. 435–444, 2005.

[11] P. Guo and J. Chen, "Arithmetic operation in membrane system," in *International Conference on BioMedical Engineering and Informatics (BMEI '08)*, vol. 1, pp. 231–234, 2008.

[12] D. Sburlan, "P systems with chained rules," in *Membrane Computing: 12th International Conference, CMC 2011, Fontainebleau, France, August 23-26, 2011, Revised Selected Papers*, vol. 7184 of *Lecture Notes in Computer Science*, pp. 359–370, Springer, Berlin, Germany, 2012.

[13] The P systems Web Page, http://ppage.psystems.eu/.

[14] M. Ai, *Research on some arithmetic operation and simulation of membrane computing model [Ph.D. dissertation]*, Harbin Institute of Technology, 2010.