An Immune Algorithm Based on P System for Classification

Lian Ye[1,2](✉) and Ping Guo[1,2]

[1] College of Computer Science, Chongqing University, Chongqing 400044, China
{ylredleaf,guoping}@cqu.edu.cn
[2] Chongqing Key Laboratory of Software Theory and Technology,
Chongqing 400044, China

Abstract. The membrane system and artificial immune system are both a branch of natural computing, which has attracted much attention in various disciplines. Inspired from the structure and inherent mechanism of membrane computing and immune computing, a membrane system based on immune mechanism algorithm is proposed to deal with classification problems. The approach contains three important stages: firstly, the candidate cells are generated by selecting from the gene pool randomly; then, calculate the affinity of the candidate cell with each element in the self-set to construct the classifier; finally, input the un-label cells into the detectors to test the performance of the classifier.

Keywords: Membrane computing · Immune algorithm · Negative selection · Classifier

1 Introduction

Natural Computing is a discipline whose aim is the study and implementation of the dynamic processes that occur in the living nature and that are likely to be interpreted as calculation procedures. Membrane computing (P systems) was initiated by Păun, which is a class of powerful computing model abstracted from the way that the living cells process chemical compounds, energy and information in their compartmental structures [1].

As a model of computation with universality property, membrane computing is so computationally efficient that it can solve difficult computational NP-complete problems in a polynomial time by creating exponentially membranes [2]. SAT problem was solved by the splitting rule of membrane system [3]. [4] gives a family of P systems to solve All-SAT problem with simplified membrane structure and few evolution rules based on the character of membrane division and parallel processing in the P systems.

Artificial immune systems (AIS) can be defined as computational systems inspired by theoretical immunology, observed immune functions, principles, and mechanisms to solve problems [5]. Negative selection is one of the most discussed algorithms in artificial immune system [6].

M. Gong et al. (Eds.): BIC-TA 2016, Part I, CCIS 681, pp. 133–141, 2016.
DOI: 10.1007/978-981-10-3611-8_14

In this paper, based on a combination of membrane system and artificial immune systems, we proposed immune classification algorithm by a membrane system. In this system, a set of detectors are created by the given rules firstly, then this collection of detectors called classifier will be used to recognize the category of the unknown objects. It can be also used in some pattern identification and combinatorial optimization problems.

The paper is organized as follows. Section 2 describes the foundation of membrane system and negative selection algorithm. The details of Π_{NS}, including the membrane structure, evolving rules and analyses are proposed in Sect. 3. The conclusions and further researches are then discussed in Sect. 4.

2 Related Works

Membrane Computing devices are generically called P systems. They constitute a theoretical computing model of a distributed, parallel and non-deterministic type.

2.1 Cell-Like P System

The main syntactic ingredients of a cell-like P system are the membrane structure shown in Fig. 1, the multi-sets, and the evolution rules. The semantics of the systems are defined through a nondeterministic and synchronous model, by introducing the concepts of configuration, transition step, and computation.

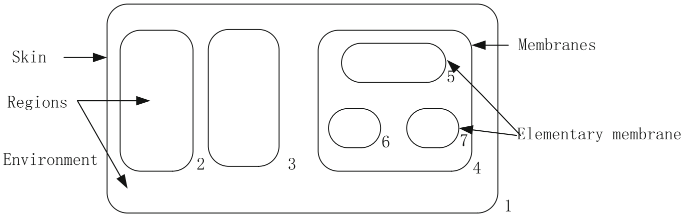

Fig. 1. The structure of cell-like P system

A basic transition P system of degree $m \geq 1$ is a tuple,

$$\Pi = (O, \mu, \omega_1, \cdots, \omega_m, (R_1, \rho_1), (R_2, \rho_2), \cdots, (R_m, \rho_m), i_o) \tag{1}$$

where,

(i) O is the alphabet of the system;
(ii) μ is a membrane structure consisting of m membranes, which are labeled by numbers in the set {1,$\cdots$, m};
(iii) $\omega_1 \cdots \omega_m$ are multi-sets, representing the objects initially presented in the regions (1,$\cdots$, m) of the system;

(iv) $\mathrm{R}_1, \cdots, \mathrm{R}_\mathrm{m}$ are finite sets of evolution rules associated with the regions (1,$\cdots$, m) of μ; $(\rho_1, \cdots, \rho_m)$ are strict partial order relations defined over $(\mathrm{R}_1, \cdots, \mathrm{R}_\mathrm{m})$ respectively, specifying a priority relation among the evolution rules; The rule can be described as the form $(u \rightarrow v, \rho_i)$, where, $u \rightarrow v$ is rewrite rule, and $\rho_i (1 \leq i \leq m)$ indicate the priority.

(v) i_o is indicating the output region.

The cell division in P systems plays in a crucial role for generating exponential work space in linear time. Here we consider the following cell membrane interaction operation to formalize in P systems area.

2.2 Negative Selection

The immune algorithm called negative selection was first developed by Forrest [6] for real-time detection of computer virus. In this algorithm, T cells are randomly generated to detect foreign antigens without reacting to self-cells in the thymus. Hence, the mature T cells leaving the thymus will not match the self-cells and will therefore only match the non-self cells. The basic concept of a negative selection algorithm is shown in Fig. 2.

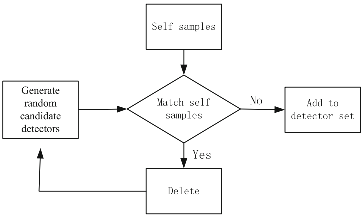

Fig. 2. Negative selection algorithm

3 Π_{NS} for Classification

The P system named Π_{NS} is proposed to create the classifier which will classify the un-label objects. The details of the system will be given in this section.

3.1 Definition

Π_{NS} for classification can defined as:

$$\Pi_{NS} = (O, \mu, \omega_1, \cdots, \omega_m, R, i_o) \tag{2}$$

I. O is a finite and non-empty alphabet of objects.
$\mathrm{O} = \{\Psi_1, \Psi_2, \beta, \gamma, \gamma', \lambda, \delta, \phi, \phi_s, \phi_{ss}, \phi_c, \phi_{cc}, \eta, \eta_0, \eta', \eta_i, \eta_{ii}\}$
$\cup \{\xi_i, 0 \leq i \leq 26\} \cup \{a_j, b_j, \cdots, z_j, 1 \leq j \leq k\}$

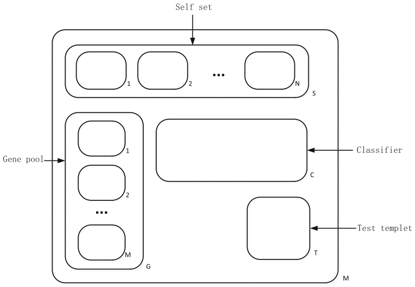

Fig. 3. Initial membrane structure

II. μ is the membrane structure composed of four main membranes in the skin membrane shown in Fig. 3. The membrane G represents gene pool to store the different gene segments as the initial multi-sets which should be placed. The membrane S represents the self-set to store the autologous cells which should also be placed at first, the number of these cells is define as $(2^k - 1)$. The membrane C denotes as classifier which is used to retain the detectors, it is empty at the beginning. The membrane T is the templte of a membrane to generate new membrane u to store the un-label cells.

III. ω_i is multi-sets, representing the objects for i record of the data set in the regions i.

IV. The rules in R should have priority, and the explanations of some rules are given here, k indicates the priority:

- $[_h u\delta]_h \rightarrow \mathrm{v}, k$
  The membrane h is dissolved, and the object u is replaced with object v.
- $[_h u \rightarrow v]_h, k$
  In the region named h, remove the multi-set of objects specified by u, and to introduce the objects specified by v.
- $[_h u \rightarrow (v, in_j)]_h, k$
  The object v should be moved into the membrane j which is the upper membrane that has not been dissolved, and the object u will be removed from membrane h.
- $[_h u \rightarrow (v, out)]_h, k$
  The object v will be moved to the region immediately outside membrane h.
- $[_h u \rightarrow [_h v]_h]_h, k$
  The object u will be replaced with a new membrane containing object v.
- $[_{h1} u]_{h1} \rightarrow [_{h2} u]_{h2} [_{h2} u]_{h2}$
  All objects and membranes in the original membrane will be duplicated and appear in the two new membranes.
- $u_j [_h]_h [_j]_j \rightarrow [_j v [_h]_h] j$
  The membrane will be filtered into j_{th} membrane by u_j, and u_j be replaced with symbol v.

V. i_o is the skin membrane to output the result.

The value of k is smaller, and the priority of the corresponding rule is higher. When k = 1, the corresponding rule will have the highest priority.

All of the rules that can be applied must be applied simultaneously. The marked output membranes are never dissolved.

3.2 Rule Set

The rules of P system Π_{NS} in each membrane are as follows:

(1) The rule set in the skin membrane called R_M. In this set, $r_1 \sim r_7$ are used in generation phase, $r_8 \sim r_{37}$ are used in testing phase. In generation phase, The number of new cell is determined by the number of symbol ϕ denoted as N. The symbol ϕ_{ss} is used to send detectors into self set. The key point of this process is that leaving only one cell in skin membrane, the others all sending into membrane S. In testing phase, The symbol L represents number of detectors to be generated. The symbol K represents the count of each attribute. The priority of r_9 is lower than that of $r_12 \sim r_{37}$, in other words, sending the objects into membrane u before duplicating membrane u.

r_1: $\Psi_1 \rightarrow \phi^N(\xi_0\xi_1 \cdots \xi_M, in_G)$
r_2: $\phi[_0]_0 \rightarrow \phi_s[_0]_0[_0]_0$
r_3: $\phi_s^N \rightarrow \phi_{ss}^N$
r_4: $\phi_{ss}^N[_0]_0 \rightarrow (\eta[_0]_0, in_s)$
r_5: $\gamma \rightarrow \gamma'\Psi_1([_0]_0, in_C), 2$
r_6: $\gamma'^L\Psi_1 \rightarrow \sharp, 1$
r_7: $\beta \rightarrow \Psi_1(\eta'', in_0)$
r_8: $\Psi_2[_T]_T \rightarrow \phi^{L-1}[_T]_T[_u]_u$
r_9: $\phi[_u]_u \rightarrow \phi_c[_u]_u[_u]_u, 2$
r_{10}: $\phi_c^{L-1} \rightarrow \phi_{cc}^L$
r_{11}: $\phi_{cc}[_u]_u \rightarrow (\eta_0[_u]_u, in_C)$
r_{12}: $a_j \rightarrow (a_j, in_u), 1; (1 \leq j \leq K)$
r_{13}: $b_j \rightarrow (b_j, in_u), 1; (1 \leq j \leq K)$
...
r_{37}: $z_j \rightarrow (z_j, in_u), 1; (1 \leq j \leq K)$

(2) The rule set in membrane G called R_G. The symbol M represents the number of attributes in the data set, its max value is 26. The symbol ξ_0 was generated a new candidate cell. The membrane *0* will be out of membrane G after that all the values have been sent into the membrane by $r_4 \sim r_{29}$.

r_1: $\xi_i \rightarrow (\xi_i, in_i), 1; (1 \leq i \leq M)$
r_2: $\xi_0 \rightarrow [_0]_0, 1$
r_3: $\phi^M[_0]_0 \rightarrow ([_0]_0, out_G)$
r_4: $a_j \rightarrow \phi(a_j, in_0), 1; (1 \leq j \leq K)$
r_5: $b_j \rightarrow \phi(b_j, in_0), 1; (1 \leq j \leq K)$
...
r_{29}: $z_j \rightarrow \phi(z_j, in_0), 1; (1 \leq j \leq K)$

(3) The rule set in gene membrane of membrane G called R_{gi}, $1 \leq i \leq M$. The symbol ξ_i releases an attribute value randomly out of corresponding membrane *i*.
$r_1\colon \xi_1 a_j \rightarrow a_j(a_j, out_1); (1 \leq j \leq K)$
$r_2\colon \xi_2 b_j \rightarrow b_j(b_j, out_2); (1 \leq j \leq K)$
...
$r_{26}\colon \xi_{26} z_j \rightarrow z_j(z_j, out_{26}); (1 \leq j \leq K)$

(4) The rule set in membrane S called R_S. The symbol N represents the number of self cells. The symbol β represents that the corresponding attribute from the two cells are matched. The symbol γ represents that the two cells are not matched.
$r_1\colon \eta\eta_i \rightarrow \eta_{ii}; (1 \leq i \leq N)$
$r_2\colon \eta_{ii}[_i]_i \rightarrow [_i]_i\eta_i([_i\eta']_i, in_0); (1 \leq i \leq N)$
$r_3\colon \beta^2 \rightarrow \beta\gamma, 1$
$r_4\colon \beta\gamma^{M-1} \rightarrow (\beta, out_S), 2$
$r_5\colon \gamma^M \rightarrow (\gamma, out_S)$

(5) The rule set in self membrane of membrane S called R_{si}, $1 \leq i \leq N$. Dissolve the membrane to release the object into outer membrane.
$r_1\colon \eta' \rightarrow \gamma\delta$

(6) The rule set in membrane *0* called R_0. $r_1 \sim r_{26}$ are matching rules for the affinity measure. The symbol m is a threshold of the affinity. If the number of matched attributes is greater than the threshold, it will release a symbol β out of membrane, otherwise it will release a symbol γ.
$r_1\colon a_j^2 \rightarrow \beta; (1 \leq j \leq K)$
$r_2\colon b_j^2 \rightarrow \beta; (1 \leq j \leq K)$
...
$r_{26}\colon z_j^2 \rightarrow \beta; (1 \leq j \leq K)$
$r_{27}\colon \gamma \rightarrow \gamma^2, 3$
$r_{28}\colon \beta^m\gamma^2 \rightarrow \eta''(\beta, out_0), 1$
$r_{29}\colon \gamma^2 \rightarrow \eta''(\gamma, out_0), 1$
$r_{30}\colon \beta \rightarrow \lambda|_{\eta''}, 1$
$r_{31}\colon a_j \rightarrow \lambda|_{\eta''}, 1; (1 \leq j \leq K)$
$r_{32}\colon b_j \rightarrow \lambda|_{\eta''}, 1; (1 \leq j \leq K)$
...
$r_{56}\colon z_j \rightarrow \lambda|_{\eta''}, 1; (1 \leq j \leq K)$
$r_{57}\colon \eta'' \rightarrow \delta, 2$
$r_{58}\colon \eta' \rightarrow \gamma\delta$

(7) The rule set in membrane C called R_C. The copies of unlabel cell are sent into each detector in this membrane. The symbol Z represents the self cell and the symbol Y represents the non-self cell. The result will be sent out of this membrane into skin membrane.
$r_1\colon \eta_0[_0]_0 \rightarrow [_0]_0([_0\eta']_0, in_u)$
$r_2\colon \beta^2 \rightarrow \beta\gamma, 1$
$r_3\colon \beta\gamma^{N-1} \rightarrow (Y, out_C), 2$
$r_4\colon \gamma^N \rightarrow (Z, out_C)$

(8) The rule set in membrane T and u called R_T. Each unknown cell is sent into the copy of membrane T saved as membrane u. The match procedure is the same with R_0. If the affinity of the two cell is greater than the threshold, it will release a symbol β out of membrane, otherwise it will release a symbol γ.

r_1: $a_j^2 \rightarrow \beta; (1 \leq j \leq K)$
r_2: $b_j^2 \rightarrow \beta; (1 \leq j \leq K)$
$\cdots$
r_{26}: $z_j^2 \rightarrow \beta; (1 \leq j \leq K)$
r_{27}: $\gamma \rightarrow \gamma^2, 3$
r_{28}: $\beta^m \gamma^2 \rightarrow \eta''(\beta, out_u), 1$
r_{29}: $\gamma^2 \rightarrow \eta''(\gamma, out_u), 1$
r_{30}: $\beta \rightarrow \lambda|_{\eta''}, 1$
r_{31}: $a_j \rightarrow \lambda|_{\eta''}, 1; (1 \leq j \leq K)$
r_{32}: $b_j \rightarrow \lambda|_{\eta''}, 1; (1 \leq j \leq K)$
$\cdots$
r_{56}: $z_j \rightarrow \lambda|_{\eta''}, 1; (1 \leq j \leq K)$
r_{57}: $\eta'' \rightarrow \delta, 2$

3.3 Algorithm Implementation

Π_{NS} contains three important phases: generate the candidate detectors randomly; calculate the affinity of the detector to decide whether deleted or converted into immune cell in classifier; test the performance of the classifier.

In skin membrane M, when it receives the starting symbol Ψ_1, the rule $(r_1 \in R_M)$ will send symbols into membrane G to produce a new elementary membrane called the candidate cell. In skin membrane G, the new membrane marked zero for a candidate detector by rule $(r_2 \in R_G)$. The symbol $\xi_i(1 \leq i \leq M)$ is sent into membrane $(1 \leq i \leq M)$ in order to release a random attribute value by the rules $(r_1 \sim r_{26} \in R_{gi})$, the detector will be out of the gene pool by rule $(r_3 \in R_G)$.

The candidate cell will be replicated by membrane separation. The affinity of each self cell in self set with the candidate cell must be calculated. Suppose the number of self cells is $N(\mathrm{N} = 2^k - 1)$, the symbol ϕ controls the times of replication to produce N copies which will be sent into membrane S by the rules $(r_2 \sim r_4 \in R_M)$. Only one membrane will be left in the skin membrane waiting for the next operation.

In membrane S, once the symbol η sent in, the rule $(r_1, r_2 \in R_S)$ will send the copy of each self cell into the candidate cell. Then the membranes i will dissolve and release the objects into membrane zero. Next, the rules $(r_1 \sim r_{26} \in R_0)$ may be reacted or not. There are $(\mathrm{M} + 1)$ kinds of possibility. If one attribution is equivalent, it will produce one β. The symbol γ is as a time slice, its count will be two by the rule r_{27}.m is the affinity to determine the similarity of two cells according to the specific problem $(1 \leq m \leq M)$. The final output is β or γ representing matched or not matched respectively.

In the skin membrane, the symbol γ means that the new cell can not be recognized by the self set and can join to the classifier by rule ($r_5 \in R_M$). The symbol β means that the new cell can be recognized and must be deleted and start the next round search of by rules ($r_7 \in R_M$ and $r_{57} \in R_0$). L indicates the number of detectors to be generated, the search process will stop until the termination symbol $\sharp$.

When the classifier is generated, the classification performance of that must be tested. In the testing process, input the symbol Ψ_2 from the environment to start the detection, then apply the rule ($r_8 \in R_M$) to generate membrane u to store the unlabel cell by rules ($r_{12} \sim r_{37} \in R_M$). Then replicate the membrane u and send them into membrane C by rule ($r_9 \sim r_{11} \in R_M$). The replication process and matching process are similar to the process of affinity testing in generation of detector.

In membrane C, the rules ($r_3 \sim r_4 \in R_C$) are applied to release the symbol Y which means that the cell is allogeneic or Z which means that the cell is autologous.

3.4 Analyses

Π_{NS} is a P system of immune algorithm for classification. The key issues of the classification problem are implemented in three membranes. Therefore, the data of practical problems in real life can be transferred into objects in the specific membrane. The membrane G represents gene pool to store the different gene segments as the initial multi-sets, in other words, which contains the range of values for each attribute and the values must be the discrete numerical. The membrane S represents the self-set to store the autologous cells. Typically, all the data is divided into two sets: training set and testing set. The records of training set are transferred and placed in the membrane S in advance. If the new candidate cell is recognized by the cells of training set, that means they are the same kind, otherwise, they belong to different kinds, and the candidate cell will be converted to detector and sent to the membrane C which denotes as classifier. When the classifier is generated, it can be used to differentiate the objects without class label. The performance of the classifier can be tested by the data of testing set.

The simulation of Π_{NS} based on negative selection was validated using iris data set from UCI Machine Learning Repository [7]. The initial data of iris data set was preprocessed. In the experimentations, the affinity of two different cell is 75%. According to the thought of membrane calculation, both in the detector generation phase and in the testing phase, the matching process between self cell with detector and the testing cell with detector are asynchronous execution. It means that this system converts serial execution of the basic algorithm to parallel execution, decreases the algorithm execution time. The classification accuracy is improved with the increase of the number of detectors and the accuracy of classification is sufficiently comparable to other top classification algorithm.

4 Conclusion

In this paper, we have proposed Π_{NS} for classification based on immune algorithm of immune system. The important advantage of the system is the parallelism which is inherent character of membrane system. In standard negative selection algorithm, the operate to calculate the affinity of two cells both in searching the immune cell and testing the unlabel cell is serial process. However, the candidate cell will be duplicated to match the self cells or detectors at the same time. It could be increase the searching speed. There are still some problem to solve, from which the most important is ignoring noise in data and increasing efficiency of algorithm when using data sets containing patterns not uniformly distributed within different classes.

References

1. Păun, G.: Computing with membranes. J. Comput. Syst. Sci. **61**(1), 108–143 (2000)
2. Calude, C.S., Paun, G.: Bio-steps beyond turing. Biosystems **77**(1C3), 175–194 (2004)
3. Obtulowicz A.: Deterministic P system for solving SAT problem. J. Inf. Sci. Technol. Roman. **4**(1-2), 195–202 (2001)
4. Ping, G., Jinfang, J., Haizhu, Z.: Solving All-SAT problems by P systems. Chin. J. Electron. **24**(4), 744–749 (2015)
5. de Castro, L.N., Timmis, J.: Artificial immune systems as a novel soft computing paradigm. Soft Comput. **7**, 526–544 (2003)
6. Forrest, S., Perelson, A.S., Allen, L.: Self-nonself discrimination in a computer. In: IEEE Computer Society Symposium on Research in Security and Privacy, Los Alamitos, pp. 202–212 (1994)
7. Frank, A., Asuncion, A.: UCI machine learning repository. School of Information and Computer Sciences, University of California, Irvine, CA (2010). http://archive.ics.uci.edu/ml