Algorithm of molecular computing on the base of membranes

D S Nikiforov and I Yu Popov
St. Petersburg National Research University of Information Technologies, Mechanics and Optics, Kronverkskiy, 49, St. Petersburg, 197101, Russia
E-mail: dmitri_nikiforov@mail.ru

Abstract. Parallelism of membrane computing can be used for efficient solving NP-complete computational problems. A problem of such type known as questionnaire problem is considered. One should choose the sequence of yes-no questions which allows to reveal the system configuration. Algorithm based on membrane computing is suggested.

1. Introduction
Membrane computing is rapidly developing area of Natural Computing [1, 2] introduced by Paun in [3]. It is inspired by the structure and the vital functions of a living cell. The perspectives in this field are related with the progress of bio- and nanotechnologies which allows one to construct structures and materials needed for these procedures [4]. This model is based on a set of membranes bounding regions where some chemical reactions are proceed. Membranes can contain other membranes and chemical reagents. We consider the cell-like membrane structure where there is a top-level membrane called skin membrane. It is surrounded by environment. Membrane is called elementary if it doesn’t contain other membranes. In mathematical definition reagents are called objects and reactions are called rules. At each step, rules are applied to objects in maximal parallel manner. It means that the multiset of rules is applied to multiset of objects and there are no more rules that can be applied to the remaining objects. The computation halts when no rule can be applied to any object in any membrane. The multiset of objects caught in the membrane marked as output region is considered as a result of computation. Membrane computing gives one a possibility to trade time for space and solve computationally hard problems in feasible time [5]. Algorithms for SAT [6,7], Subset Sum [8], Knapsak [9], Bin Packing [10] and Partition [11] were suggested. It is known that any NP-complete problem can be reduced to another one but it can be non-trivial so construction of specific algorithms for various problems is interesting and important. Algorithm for questionnaire problem is suggested in this paper.

2. Questionnaire problem
To make the procedure clear we give a simple example (the toy task) of the questionnaire problem, which is related with known logical task. There are two cities: A and C. Inhabitants of A always tell the truth, inhabitants of C in turn tell the truth and a lie. Observer E is in one of these cities. He meets one person and wants to know in what city is he, and from what city is
this person. How many questions and in what sequence should be asked if the person answers
only yes or no?

For our example, let the initial set of questions contain the following ones:

(1) Am I in A?
(2) Am I in C?
(3) Are you from A?

There are 6 possible configurations of system because there are two types of C inhabitant’s
behavior: he can tell the truth answering the first question or he can tell a lie. We denote these
options respectively C+ and C−. Shannon entropy of the system is \(\log_2 6\). It means that two
questions are not sufficient to obtain the answer and there exists an appropriate sequence of
three correct questions because \(2 < \log_2 6 < 3\). There are three ones in our set (we assume that
we have correct set of questions) so the only thing to do is to find the right order of them. Table
1 shows the answers to the questions asked in the following order 1, 2, 3: + means yes, − means
no.

| E in A | E in C |
|--------|--------|
| Person from A | + − + − + + |
| Person from C+ | + + − − − − |
| Person from C− | − − + + + + |

Table 1. Answers for valid sequence of questions \(\{1, 2, 3\}\).

One can see that plus-minus sequences are unique for each cell, so all possible configurations
are distinguishable. In contrast, table 2 shows the answers to the sequence of questions that
leaves some configurations undistinguishable.

| E in A | E in C |
|--------|--------|
| Person from A | − + + + − − |
| Person from C+ | − + + + + − |
| Person from C− | + − − − − + |

Table 2. Answers for "bad" sequence of questions \(\{2, 3, 1\}\).

In our model all possible sequences of questions are processed in the same time. Each of
them corresponds to membrane substructure of the following type:

\[
[0,1\_n,2\_n,3\_n,3\_y,3\_y,2\_y,2\_y,3\_n,3\_n,3\_y,3\_y,2\_y,2\_y,3\_n,3\_n,3\_y,3\_y,2\_y,1\_y,0]
\]

Common notation for membrane structures is used: all stuff placed between opening and
closing brackets with same label is considered as content of membrane with this label. This
example refers to the sequence \(\{1, 2, 3\}\). 1\_n, 1\_y, 2\_n, 2\_y, 3\_n, 3\_y are the labels of membranes
corresponding to negative (with suffix \(n\)) and positive (\(y\)) answers for the first, the second and
the third questions from the list. Tree representation of this structure is shown in figure 1.
0-membrane is a container. In our example the skin membrane, denoted as 00, contains 3! = 6 0-membranes for every possible order of questions. In the initial configuration each 0-membrane contains objects $a_1, a_2, \ldots, a_6$ corresponding to table cells listed from left to right and from top to bottom (see table 3) each as a single copy. $a_1$ corresponds to situation when E is in A and person from A is asked and so on. 0-membrane also contain four copies of object $t$ corresponding to the turn of the truthful answer for C+ and the turn of the false answer for C−. Object $f$ coming later is the opposite to object $t$. There are objects $finish$ located in every elementary membrane in a single copy.

Table 3. Each $a$-object corresponds to one cell.

| E in A | E in C |
|--------|--------|
| Person from A | $a_1$ | $a_2$ |
| Person from C+ | $a_3$ | $a_4$ |
| Person from C− | $a_5$ | $a_6$ |

We use in and out communication rules in our model. In communication rules introduce an object into the specified membrane and possibly object can be modified. Out rules send object out of the membrane and possibly object can be modified. Following in rules send $a$-objects to $1_y$ or $1_n$ membranes according to person’s answer on the first question:

$$
\begin{align*}
& a_1 \ [1_y]_{1_y} \rightarrow \ [1_y a_1]_{1_y} \\
& a_2 \ [1_n]_{1_n} \rightarrow \ [1_n a_2]_{1_n} \\
& a_3 t \ [1_y]_{1_y} \rightarrow \ [1_y a_3 f]_{1_y} \\
& a_3 f \ [1_n]_{1_n} \rightarrow \ [1_n a_3 t]_{1_n} \\
& a_4 t \ [1_n]_{1_n} \rightarrow \ [1_n a_4 f]_{1_n} \\
& a_4 f \ [1_y]_{1_y} \rightarrow \ [1_y a_4 t]_{1_y} \\
& a_5 t \ [1_n]_{1_n} \rightarrow \ [1_n a_5 f]_{1_n} \\
& a_5 f \ [1_y]_{1_y} \rightarrow \ [1_y a_5 t]_{1_y} \\
& a_6 t \ [1_y]_{1_y} \rightarrow \ [1_y a_6 f]_{1_y} \\
& a_6 f \ [1_n]_{1_n} \rightarrow \ [1_n a_6 t]_{1_n}
\end{align*}
$$

There are two rules for two cells corresponding to person from A. Object $a_1$ goes to $1_y$ membrane because positive answer on the first question is always given if E is in A. In the opposite $a_2$ goes to $1_n$ because person from A answers ”no” on the first question if E is in C. The same logic applies to objects $a_3, \ldots, a_6$ but their routes depend on the turn of truthful or
false answer given by C+ or C−. The presence of t or f objects is taken into consideration to represent this fact. Analogous rules involve 2a, 2g, 3a, and 3g membranes.

Introduced rules send a-objects from 0-membrane deeper into membrane structure. They go from root to leaves if tree representation is considered. Consider the configuration after three steps made. All a-objects came to elementary membranes. If the order of questions was appropriate each elementary membrane contains at most one a-object otherwise some of them can contain several a-objects. Then the next out rules come into play:

$$[i a_n \text{ finish } ]_i \rightarrow [i]q_i$$

$$[j q_i ]_j \rightarrow [j]q_{ji}$$

$$[k q_{ji}]_k \rightarrow [k]k q_{kji}$$

where a_n is arbitrary a-object; i, j, k are in (1, 2, 3); q-objects are special objects tracking the order of questions (q1, q2, q3, q12, q13, q23, q21, q31, q32, q31, q21, q12, q32, q213, q132). Three steps later q-objects with three indexes will arrive to 0 membranes. There will be 6 q-objects in 0 membranes corresponding to valid sequence of questions but there will be less than 6 q-objects in 0 membranes corresponding to “bad” sequence of questions. All q-objects can be popped to 00 membrane if it is considered as an output region and one can count their multiplicities.

Consider the questionnaire problem in general. There are n cities of type A (A1, A2, …, An) and m cities of type C (C1, C2, …, Cm). The number of possible system configurations is (n + m)(n + 2m). The lower bound for necessary and sufficient minimal number of questions is \( \log_2(n + m)(n + 2m) \). The initial set of questions contains not less than mentioned number of questions. It is possible that no fitting sequence can be found in this set in general. In this case when computation halts there will be insufficient number of q-objects in 00 membrane. The structure, initial configuration and rules are constructed in the same way as in example. The difference lies in a greater number of a and q objects needed and deeper membrane structure.

Further generalization leads to functional rules. We used two objects t and f one by one to alternate a-objects’ behavior according to the turn of truthful or false answer: t → f → t and so on. Rules with arbitrary alternation can be suggested, e.g. involving t1 → … → tn → f1 → … → fm and so on. Such rules can handle arbitrary determinate answering strategy. Another point of interest is usage of nondeterminism in rules’ definition and application.

The work was partially financially supported by the Government of the Russian Federation (grant 074-U01), by the Ministry of Science and Education of the Russian Federation (GOSZADANIE 2014/190, Project 14.Z50.31.0031), by grant of Russian Foundation for Basic Researches and grants of the President of Russia (state contracts 14.24.13.2045-MK and 14.24.13.1493-MK).

References

1. Ciobanu G, Perez-Jimenez M J and Paun G 2008 Applications of Membrane Computing (Berlin: Springer)
2. Popov I Yu, Vorobyova A V and Blinova I V 2014 Int. J. Bioinform. Res. Appl. 10 145-56
3. Paun G 2000 J. Comput. System Sci. 61 108-43
4. Rambidi N G 2007 Nanotechnologies and Molecular Computers (Moscow: Fizmatlit)
5. Gutierrez-Naranjo M A, Perez-Jimenez M J and Romero-Campero F J 2007 Theoret. Comput. Sci. 371 54-61
6. Perez-Jimenez M J, Romero-Jimenez A and Sancho-Caparrini F 2003 Proc. 5th Workshop on Descriptional Complexity of Formal Systems (Budapest) 284-94
7. Paun G 1999 Centre for Discrete Mathematics and Theoretical Computer Science Research Report 102
8. Perez-Jimenez M J and Riscos-Nunez A 2005 New Generat. Comput. 23 367-84
9. Perez-Jimenez M J and Riscos-Nunez A 2004 Lecture Notes in Comput. Sci. 2933 250-68
10. Perez-Jimenez M J and Romero-Campero F J 2004 Proc. 2nd Brainstorming Week on Membrane Computing (Seville) 414-30
11. Gutierrez-Naranjo M A, Perez-Jimenez M J and Riscos-Nunez A 2005 Soft Comput. 9 673-8