P systems based multi-objective optimization algorithm*

Huang Liang[1], He Xiongxiong[2], Wang Ning[1]** and Xie Yi[1]

(1. National Laboratory of Industrial Control Technology, Institute of Advanced Process Control, Zhejiang University, Hangzhou 310027, China; 2. Zhejiang University of Technology, Hangzhou 310033, China)

Abstract Based on P systems, this paper proposes a new multi-objective optimization algorithm (PMOA). Similar to P systems, PMOA has a cell-like structure. The structure is dynamic and its membranes merge and divide at different stages. The key rule of a membrane is the communication rule which is derived from P systems. Mutation rules are important for the algorithm, which has different ranges of mutation in different membranes. The cooperation of the two rules contributes to the diversity of the population, the conquest of the multimodality of objective function and the convergence of algorithm. Moreover, the unique structure divides the whole population into several subpopulations, which decreases the computational complexity. Almost a dozen popular algorithms are compared using several test problems. Simulation results illustrate that the PMOA has the best performance. Its solutions are closer to the true Pareto-optimal front and distributed well. Moreover, it converges fast.

Keywords: P systems, evolutionary algorithm, multi-objective optimization, Pareto optimality.

Many real-world problems must be optimized simultaneously on several incommensurable and conflicting objectives. In such a framework, there is no single optimal solution but rather a set of alternative solutions. The multi-objective optimization problems are too complex to be solved by exact methods, such as gradient search and linear programming, since the output of classical search and optimization methods is a single optimized solution. The ability of an evolutionary algorithm (EA) to find multiple optimal solutions in one single simulation run makes EAs unique in solving multi-objective optimization problems. The numerous applications and the rapidly growing interest in the area of EAs take this fact into account[1].

Since Schaffer suggested the first multi-objective evolutionary algorithm, the vector-evaluated genetic algorithm (VEGA)[2], a number of researchers across the globe have developed different implementations of multi-objective EAs. Zitzler et al. compared eight existing multi-objective evolutionary algorithms (MOEAs) on six test problems ZDT1 to ZDT6[3]. The results showed that the Strength Pareto Evolutionary Algorithm (SPA)[4] and the Non-dominated Sorting Genetic Algorithm (NSGA)[5] are superior to other algorithms. They have become popular MOEAs. However, they are criticized for their disadvantages. This paper attempts to suggest a new powerful algorithm based upon other researchers' current studies. Specifically, the new algorithm adopts the idea of P systems.

In 1998, Gheorghe Păun introduced P systems (Membrane computing)[6,7], which became a popular research area. As a powerful computing device, P systems have a significant potential to be applied to various problems of biology as well as to computer science[8,9]. However, the practical applications need more investigation. This paper investigates this kind of application of P systems, which considers the basic framework of P systems to implement the multi-objective optimization.

By using standard P systems it is difficult to directly solve multi-objective optimization problems (MOOPs). The new algorithm incorporates ideas from current seminal work on multi-objective evolutionary algorithms. It is called P systems based Multi-objective Optimization Algorithm (PMOA).

PMOA has two unique characteristics. Firstly, it has the structure of P systems[6—8]. Secondly, several subsystems exist in an algorithm. Some are single objective optimization subsystems, while others are multi-objective optimization subsystems. These features assure that the algorithm works well. The following experiments show that PMOA outperforms almost a dozen popular current MOEAs. Moreover, this paper reports experiments about test problems with a large number of parameters, which are absent in Zitzler's study.

* Supported by National Natural Science Foundation of China (Grant No. 60421002)

** To whom correspondence should be addressed. E-mail: nwang@iipc.zju.edu.cn

The remainder of this paper is organized as follows. In Section 1, the multiobjective optimization problem is briefly mentioned. Then, in Section 2, the PMOA is proposed in detail. Experimental designs and their results are the subject of Section 3 and Section 4. Section 5 indicates the conclusions.

1 The multi-objective optimization problems

Multi-objective optimization problems involve multiple, conflicting objectives which are to be minimized or maximized. In the following, MOOPs are stated in their general form:

$$\left.\begin{array}{lll} \text{Minimize/Maximize:} & f_m(x), & m = 1,2,\cdots,M \\ \text{subject to:} & g_j(x) \geqslant 0, & j = 1,2,\cdots,J \\ & h_k(x) = 0, & k = 1,2,\cdots,K \\ & x_i^L \leqslant x_i \leqslant x_i^U, & i = 1,2,\cdots,n \end{array}\right\} \quad (1)$$

A solution x is a vector of n decision variables: $x = (x_1, x_2, \cdots, x_n)^T$. The last sets of constraints are called variable bounds, restricting each decision variable x_i to take a value between a lower x_i^L and an upper x_i^U bound. Theses bounds constitute a decision variable space D, or simply the decision space. There are M objective functions $f(x) = (f_1(x), f_2(x), \cdots, f_M(x))^T$ considered in the above formulation. Each objective function can be either minimized or maximized. Without loss of generality, only minimization problems are considered in what follows.

The multi-objective optimization problems have two goals:

1. to find a set of solutions as close as possible to the Pareto-optimal front;

2. to find a set of solutions as diverse as possible.

The first goal is mandatory in any optimization task. The second goal is entirely specific to multi-objective optimization, and it asks to find solutions which are sparsely spaced in the Pareto-optimal region.

2 The PMOA algorithm

PMOA incorporates the idea of P systems and the basic operator of EAs. First, inspired by the idea of P systems, PMOA has a special structure which consists of many *membranes*. Each membrane has its own subpopulation which evolves locally in the membrane. Therefore, each membrane works as an evolutionary algorithm. These subpopulations keep in touch by means of the communication rule and the dynamic structure. In addition, the communication rule, the mutation rule, and other rules are similar to their counterparts in P systems as well as to their corresponding operators in EAs. Secondly, the algorithm pays attention to the ends of the Pareto front. The ends constitute the skeleton of the Pareto front, so it has never drawn any more attention than necessary. In other words, PMOA emphasizes the optimization of each single objective. In fact, many real-world problems require that one special objective is optimized and other objectives do not turn for the worst. On the other hand, the emphasis on the single objective makes the set of solutions a faster and more effective approximation to the Pareto front.

2.1 The dynamic structure

As in standard P systems, the structure of PMOA consists of several membranes. The number of membranes is different according to the number of objectives and the difficulty of the optimization problem. As is shown in Fig. 1, membranes $m_{2,1}$, $m_{2,2}$, $\cdots$, $m_{2,N2}$ form the subsystem Sub_2. The outermost membrane, *skin*, contains several subsystems. Except the skin, all membranes merge into one m_{instar} at a certain time. Then, it divides into the same structure as it was previously reported. The dynamic course of structure evolution is described in Fig. 1. While the subsystems merge into the membrane m_{instar}, the objects (strings or solutions) of all subsystems will enter into the membrane m_{instar}. During the process of division, the copies of objects which have the best value on the objective f_1 will form the subpopulation of subsystem 1, and the copies of objects which have the best value on the objective f_2 will form the subpopulation of subsystem 2 and so on. The copies of objects which have the best Pareto value on all objectives will form another subpopulation. Some objects may be copied several times and some will be deleted without letting behind any copies.

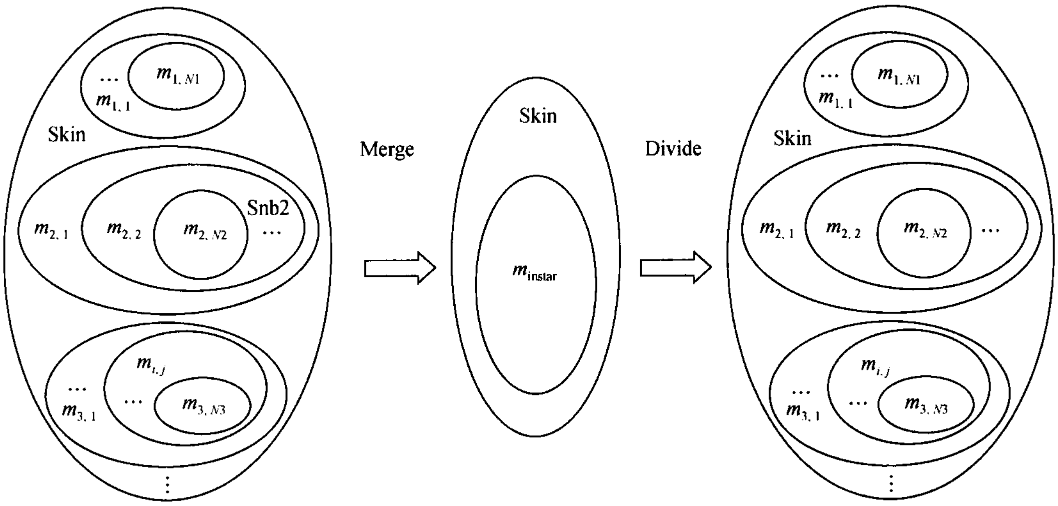

Fig. 1. Evolution of structure.

2.2 The function of subsystems

Each subsystem has its own preference order of the objectives. Often, the subsystems have a unique objective which they will optimize. Specifically, some subsystems optimize one of the objectives and do not care about other objectives. In fact, locally we have a single objective optimization system. For example, taking a problem with two objectives, the population of subsystems distribute as shown in Fig. 2. Around the ends of the Pareto front, there are more objects. Most of the objects come from single optimization subsystems. The subpopulation of subsystem 1 speeds up the decrease of objective f_1. The subsystem 2 contributes the convergence of solutions on the objective f_2.

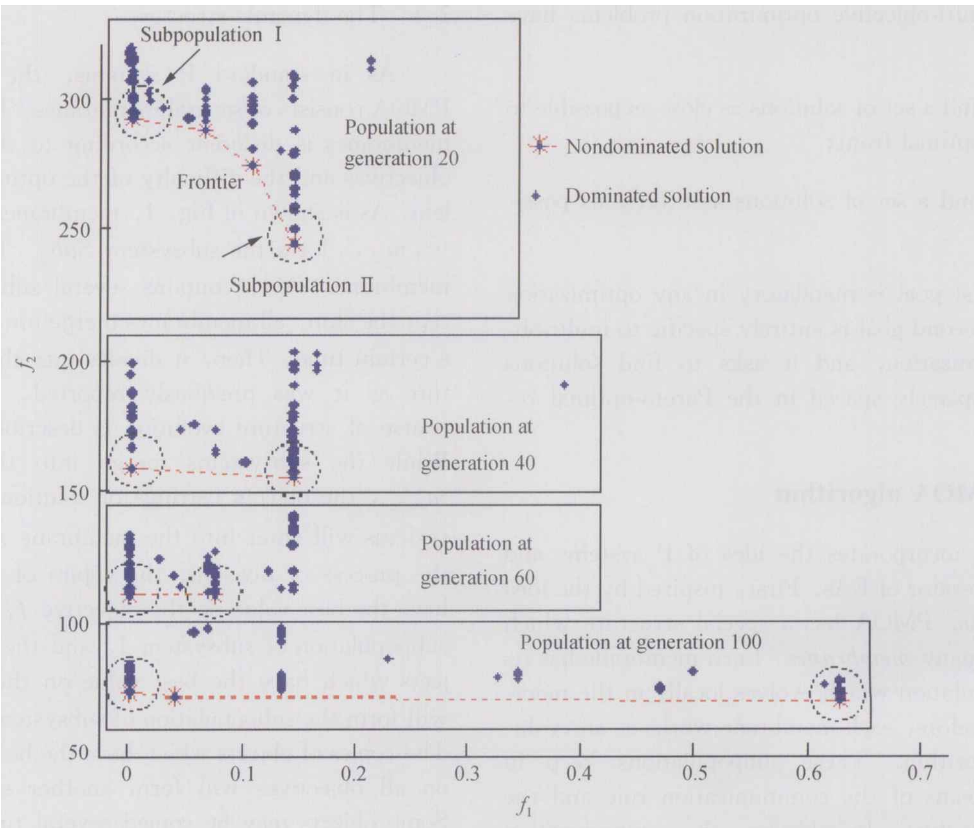

Fig. 2. Evolution of subsystems.

Therefore, a complete set of solutions will approach the Pareto front faster. On the other hand, there is another subsystem which keeps the entire population varied.

2.3 The evolution of subpopulations

As shown in Fig. 1, a subpopulation has one or more membranes. The following experiments study the subsystems with three membranes, the outer membrane, the middle membrane, and the inner membrane. Each membrane has its own subpopulations. The evolution of the subsystem consists of the evolution of its subpopulation.

The evolution of the subpopulation can incorporate ideas from most of the popular EAs. However, one key rule must be adopted from P systems, i.e., the communication rule. In fact, there are several communication rules that can be selected. The following experiments adopt the antiport and symport rules. In Fig. 1, subsystem 3 is taken as an example. If the whole system is used for an optimization problem with two objectives, the third subsystem is a *multi-objective optimization one*. Its best objects of the outer membrane are its current Pareto front. When the communication rule is used, the outer membrane $m_{3,1}$ sends its best objects into the skin membrane. However, the copies of the solutions of Pareto front remain in the membrane $m_{3,1}$. On the other hand, when the outer membrane sends the same worst objects into the middle membrane $m_{3,2}$, the same number of best solutions of the latter enter into the outer membrane. At the same time, the middle membrane uses several of its worst solutions to exchange for the same number of the best solutions of the inner membrane $m_{3,3}$.

The mutation rule of PMOA is also unique. It might be described as follows:

$$S \rightarrow S' \tag{2}$$

where

$$\begin{cases} S = x_1 x_2 \cdots x_l \\ S' = y_1 y_2 \cdots y_l \end{cases} \tag{3}$$

$$\begin{cases} y_i = x_i \\ \text{or } y_i = x_i + \eta_i \end{cases} \tag{4}$$

$i = 1, 2, \cdots, l$; l is the length of the strings or the number of variables of a problem and x_i, y_i are the variables of the problem; these l variables compose a string S, which is an approximate solution of the problem. η_i is a random number with normal distribution. Its range of distribution is different in different membranes. According to the rule, several genes x_i in a string S turn into $y_i = x_i + \eta_i$ at random. The selection of their ranges is important for the convergence of the PMOA.

With the above two main rules, the algorithm can efficiently implement the multi-objective optimization. In order to improve it, some operators or methods of evolutionary algorithms should be utilized according to different problems. Similar to the $(\mu + \lambda) - ES$ in evolutionary strategy, the offspring population C_t is first created in a membrane by using its parent population P_t. Then the two populations are combined together to form R_t of size $(\mu + \lambda)$. The non-dominated front of the entire population R_t is taken out as a part of the population of the next generation P_{t+1}. If the number of solutions of the offspring population is less than μ, another front is adopted from the remaining population $R_t - P_{t+1}$. While the last front is considered, there may exist more solutions in the last front than the remaining slots in the offspring population. Then, we should use a niching strategy to choose the objects of the last front, which reside in the least crowded region. The PMOA adopts the crowding distance assignment procedure like the elitist non-dominated sorting genetic algorithm (NSGA-2)[1,10]. When the entire population converges to the Pareto-optimal front, the continuation of this algorithm will ensure a better spread among the solution. Therefore, the second goal of the MOOPs is implemented.

Suppose the initial subpopulation is denoted by P_0. The evolution of the subpopulation is outlined as follows:

1. The communication rule is applied. The membrane sends out some objects and adopts the objects which come from other membrane to form the population P_t'.

2. Create the offspring population C_t from P_t' by using the mutation rule. Other operators of evolutionary algorithm can be considered. In the following experiments, the cross and crowded tournament are explored.

3. Combine the parent and offspring population $R_t = P_t' \cup C_t$.

4. According to the Pareto selection and the

crowding distance, choose the best μ objects from R_t to form the offspring population P_{t+1}.

2.4 Elitism in skin

At each generation, each subsystem sends their front into the skin. Therefore, there are a mass of solutions, some of which are dominated by the other. In the skin, the dominated solutions are deleted and only the non-dominated solutions are stored at each generation. Many seminal works report that the incorporation of elitism improves the performance of EAs. However, the elitism in the skin never takes into evolution. The function of such strategy is not obvious in the early stages of evolution. When non-dominated solutions exceed the size of the offspring population, some of them will be deleted. If the Pareto front of the problem is wide and its shape is complex, the front with little solutions coming from the subsystems makes it difficult to represent the real front. The store of the elitism can increase the number of the Pareto solutions and improve their distribution.

3 Experiments

3.1 Design for the experiment

With the availability of many multi-objective evolutionary algorithms, it is natural to ask which of them performs better when compared to other algorithms on various test problems. Zitzler et al. constructed six difficult test problems, and investigated the performance of various popular multi-objective EAs. They are Fonseca and Fleming's multi-objective EA (FFGA), the Niched Pareto Genetic Algorithm (NPGA), Hajela's weighted-sum based approach (HLGA), the Vector Evaluated Genetic Algorithm (VEGA), the Non-dominated Sorting Genetic Algorithm (NSGA), a single-objective evolutionary algorithm using weighted-sum aggregation (SOEA), and the Strength Pareto Evolutionary Algorithm (SPEA). The PMOA is compared with these algorithms on ZDT1, ZDT2, ZDT4, and ZDT6. The same parameter settings are adopted. The population size is 100 and the stopping criterion is 250 generations. In the following experiments, the PMOA has 3 subsystems and each subsystem has 3 membranes. The 9 membranes share the population with 100 objects as shown in Table 1. The ranges of mutation which belong to different membranes are different according to the different decision spaces. As is shown in the following table, the ranges on ZDT6 are different from those of other test problems.

Table 1. The parameters of each membrane

Membranes	$m_{1,1}$	$m_{1,2}$	$m_{1,3}$	$m_{2,1}$	$m_{2,2}$	$m_{2,3}$	$m_{3,1}$	$m_{3,2}$	$m_{3,3}$
Subpopulation size	2	3	5	5	10	5	5	30	35
Mutation range	0.2	0.02	10^{-3}	0.05	0.01	10^{-3}	0.1	0.05	0.005
Mutation range (ZDT6)	0.1	0.005	10^{-4}	10^{-4}	10^{-5}	10^{-7}	0.05	5×10^{-4}	10^{-6}

3.2 Test problems

In our study, only ZDT1, ZDT2 and the most difficult problems ZDT4, ZDT6 are used for the comparisons[3]. These problems have two objectives which are to be minimized:

$$\begin{cases} \text{Minimize:} & f_1(x) \\ \text{Minimize:} & f_2(x) = g(x)h(f_1(x), g(x)) \end{cases} \quad (5)$$

These problems are different according to the different definitions of the three functions $f(x)$, $g(x)$, and $h(x)$ as follows:

$$\text{ZDT1}\begin{cases} f_1(x_1) = x_1 \\ g(x_1, \cdots, x_m) = 1 + 9\sum_{i=2}^{m} x_i/(m-1) \\ h(f_1, g) = 1 - \sqrt{(f_1/g)} \end{cases} \quad (6)$$

where $m = 30$ and $x_i \in [0, 1]$. This is the easiest problem with a convex Pareto-optimal front.

$$\text{ZDT2}\begin{cases} f_1(x_1) = x_1 \\ g(x_1, \cdots, x_m) = 1 + 9\sum_{i=2}^{m} x_i/(m-1) \\ h(f_1, g) = 1 - (f_1/g)^2 \end{cases} \quad (7)$$

where $m = 30$ and $x_i \in [0, 1]$. The difficulty with this problem is that the Pareto-optimal region is non-convex.

$$\text{ZDT4}\begin{cases} f_1(x_1) = x_1 \\ g(x_1, \cdots, x_m) = 1 + 10(m-1) \\ \qquad + 9\sum_{i=2}^{m}(x_i^2 - 10\cos(4\pi x_i)) \\ h(f_1, g) = 1 - \sqrt{(f_1/g)} \end{cases} \quad (8)$$

where $m = 10$, $x_1 \in [0, 1]$, and $x_1, \cdots, x_m \in [-5, 5]$. This contains 21^9 local Pareto-optimal fronts. The sheer number of multiple local Pareto-optimal fronts produces a large number of hurdles for an algorithm to converge to the global Pareto-optimal front.

$$\text{ZDT6}\begin{cases} f_1(x_1) = 1 - e^{-4x_1}\sin^6(6\pi x_1) \\ g(x_1, \cdots, x_m) \\ \quad = 1 + 9\left(\left(\sum_{i=2}^{m} x_i\right)\Big/(m-1)\right)^{0.25} \\ h(f_1, g) = 1 - (f_1/g)^2 \end{cases} \tag{9}$$

where $m = 10$ and $x_i \in [0, 1]$.

The difficulties are that the problem has a non-convex Pareto-optimal set, the density of solutions across the Pareto-optimal region is non-uniform, and the density towards the Pareto-optimal front is also thin.

4 Results and discussions

Figs. 3—7 illustrate the comparison between the simulation result of PMOA and the results of other algorithms[3]. Elitism is an important factor in evolutionary multi-objective optimization. The performance of the algorithms improved significantly when SPEAs elitist strategy was adopted. The algorithms with an asterisk adopted the elitism. As the figures show, the solutions of PMOA reside at the bottom of the figures. Therefore, the performance of PMOA is better than that of other algorithms on ZDT1, ZDT2, and ZDT4.

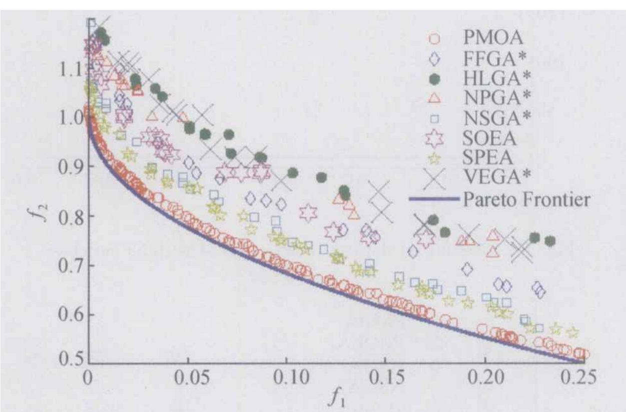

Fig. 3. Results on the test function ZDT1.

ZDT4 seems to be the most difficult test problem, since even the NSGA with a population size of 10000 was not enough to converge to the optimal trade-off front after 250 generations. The PMOA with a population size of 100 after 250 generations obtains the result which is better than the solutions found by the elitism variants of NSGA with a population size of 500 after 1000 generations.

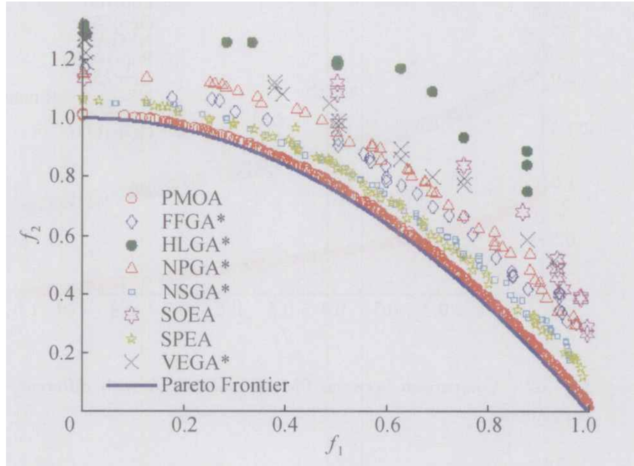

Fig. 4. Results on the test function ZDT2.

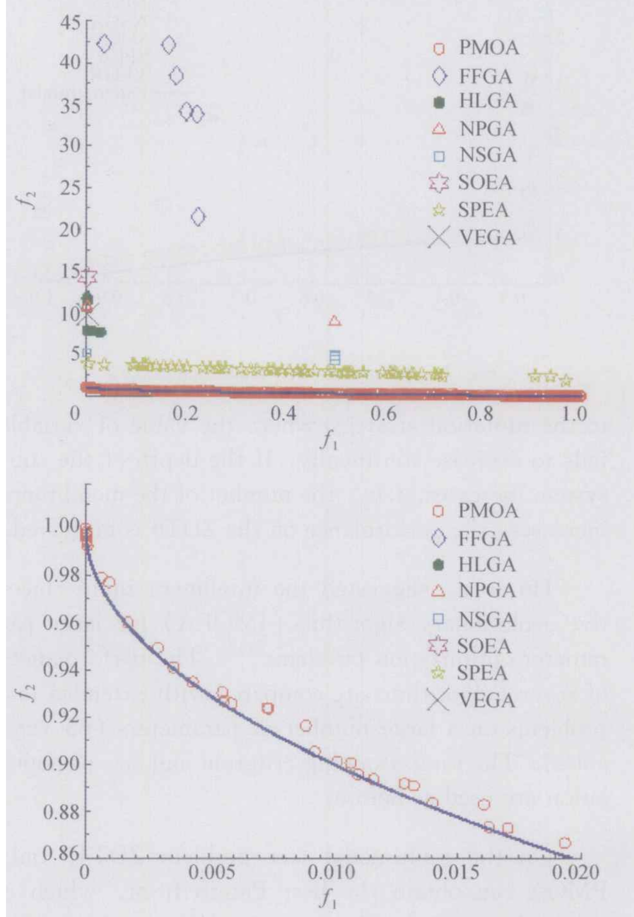

Fig. 5. Results on the test function ZDT4.

The performance of PMOA on ZDT6 surpasses that of other algorithms except SPEA. However, the distribution of solutions is better with PMOA. Moreover, the maximum of variables except x_1 is optimized to less than 5.0×10^{-8}. This is probably due

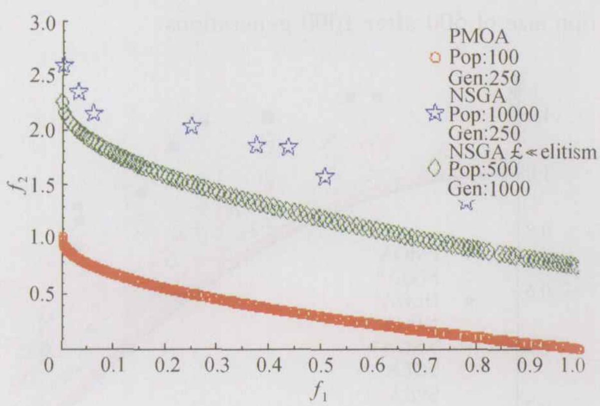

Fig. 6. Comparison between PMOA and NSGA with different populations.

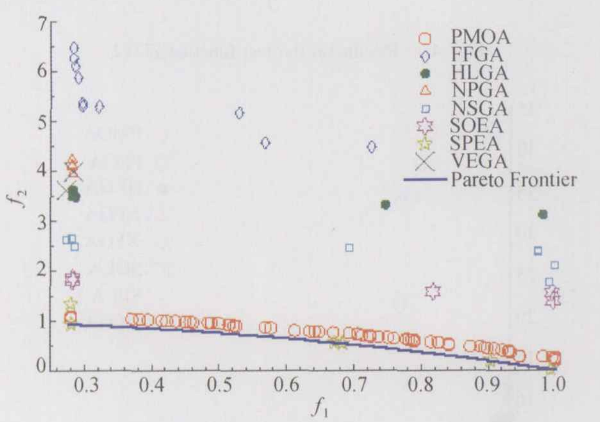

Fig. 7. Results on the test function ZDT6.

to the mutation strategy where the value of variable fails to decrease continually. If the depth of the subsystem increases, i.e., the number of the membranes increases, the performance on the ZDT6 is improved.

Ho et al. suggested the intelligent multi-objective evolutionary algorithm (IMOEA) for large parameter optimization problems[11]. The performances of several algorithms are compared with extended test problems on a large number of parameters (63 variables). The same stopping criterion and size of population are used as before.

On the multimodal test problem ZDT4, only PMOA can obtain the best Pareto front, which is much closer to the Pareto-optimal front when compared with the other algorithms in Fig. 8. Its performance is worse than that of IMOEA on the ZDT6 with 63 variables. As is shown in Fig. 9, however, it performs better than NSGA-2, SPEA2, and other algorithms. As shown by the figures above, The PMOA performs well according to the two goals of multi-objective optimization problems. In addition, it converges faster in terms of generations. In fact, it runs faster due to less computational complexity.

NSGA has been criticized for its computational complexity $O(MN^3)$, where M represents the objectives and N represents the population size. The NSGA-2 and SPEA needs at most $O(MN^2)$ computations in each generation. The PMOA only required at most $O(M_{sub}N_{sub}^2)$, where the M_{sub} is the number of objectives of the maximum subsystem and N_{sub} is the size of the maximum subpopulation. Because the whole population is divided into several small subpopulations and there are some single objective optimization subsystems, the computational complexity decreases and therefore the algorithm runs faster. The population size strongly influences the algorithms capability to converge towards the true Pareto-optimal front. Obviously, the small subpopulations of PMOA are not enough to keep diversity among the individuals. By increasing the population size properly, it can solve more difficult MOOPs. However, the range of mutation and size of subpopulations should be adjusted according to different problems in order to obtain best performance on difficult problems.

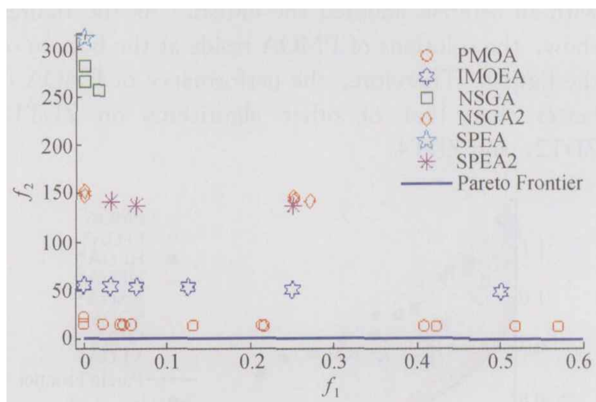

Fig. 8. Results on the test function ZDT4 with 63 variables.

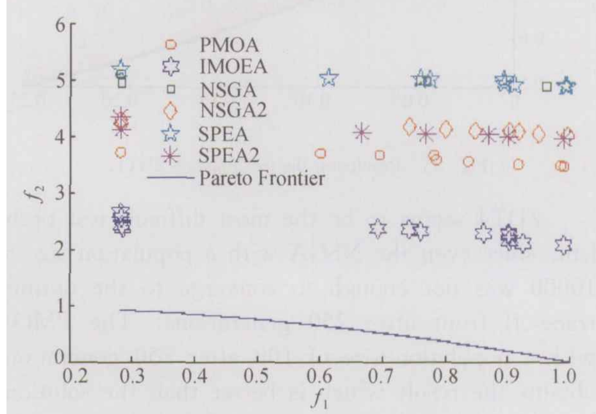

Fig. 9. Results on the test function ZDT6 with 63 variables.

5 Conclusion

Inspired by the theory of P systems, a well-performing multi-objective optimizer is proposed in this paper. First, this paper extends the applications of P systems. Second, it proposes a new strategy for the multi-objective optimization problems. As shown by the experiments above, the quality of non-dominated solutions obtained by PMOA is superior to those of other EAs: the distance between the obtained Pareto front and the true Pareto-optimal front is small, and the distribution of non-dominated solutions is nearly uniform. The relative performance of PMOA demonstrated that PMOA may be well qualified to join the current set of "best performers" in the multi-objective evolutionary algorithm community.

References

1 Deb K. Multi-Objective Optimization Using Evolutionary Algorithms. New York: Wiley, 2001

2 Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of an International Conference on Genetic Algorithms and Their Applications sponsored by Texas Instruments and U. S. Navy Center for Applied Research in Artificial Intelligence (NCARAI), 1985, 93—100

3 Zitzler E, Deb K and Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results. Evol Comput, 2000, 8 (2): 173—195

4 Zitzler E and Thiele L. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation, 1999, 3(4): 257—272

5 Serinivas N and Deb K. Multiobjectie optimization using nondominated sorting in genetic algorithms. Evolutionary Computation, 1994, 2(3): 221—248

6 Păun Gh. Computing with membranes. Journal of Computer and System Sciences, 2000, 61(1): 108—143

7 Păun Gh. From cells to computers: Computing with membranes (P systems), BioSystems, 2001, 59(3): 139—158

8 Alhazov A and Sburlan D. Static sorting P systems. In: Applications of Membrane Computing. Berlin: Springer-Verlag, 2005, 215—252

9 Nishida TY. Membrane algorithms: Approximate algorithms for NP-complete optimization problems. In: Applications of Membrane Computing. Berlin: Springer-Verlag, 2005, 301—312

10 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multi-objective genetic algorithm: NSGA-2. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182—197

11 Ho SY, Shu LS and Chen JH. Intelligent evolutionary algorithm for large parameter optimization problems. IEEE Transactions on Evolutionary Computation, 2004, 8(6): 522—541