

# A Multi-objective Particle Swarm Optimization Based on P System Theory

Taowei Chen<sup>1,a</sup>, Yiming Yu<sup>2</sup> and Kun Zhao<sup>1</sup>

<sup>1</sup>Information School, Yunnan University of Finance and Economics, 650221, Kunming China

<sup>2</sup>Information Center, Yunnan University of Finance and Economics, 650221, Kunming China

**Abstract.** Particle swarm optimization(PSO) algorithm has been widely applied in solving multi-objective optimization problems(MOPs) since it was proposed. However, PSO algorithms updated the velocity of each particle using a single search strategy, which may be difficult to obtain approximate Pareto front for complex MOPs. In this paper, inspired by the theory of P system, a multi-objective particle swarm optimization (PSO) algorithm based on the framework of membrane system(PMOPSO) is proposed to solve MOPs. According to the hierarchical structure, objects and rules of P system, the PSO approach is used in elementary membranes to execute multiple search strategy. And non-dominated sorting and crowding distance is used in skin membrane for improving speed of convergence and maintaining population diversity by evolutionary rules. Compared with other multi-objective optimization algorithm including MOPSO, dMOPSO, SMPSO, MMOPSO, MOEA/D, SPEA2, PESA2, NSGAI on a benchmark series function, the experimental results indicate that the proposed algorithm is not only feasible and effective but also have a better convergence to true Pareto front.

## 1 INTRODUCTION

Nowadays, nature-inspired metaheuristic algorithms such as evolutionary algorithm, PSO and their variants have been widely developed to solve MOPs since they can obtain better Pareto-optimal sets than the weighted sum method does[1]. Among them, PSO as a class of swarm intelligence(SI) is a global optimization algorithm that mimics the social cooperative and competitive behavior of bird flocking and actions of ant colony[2]. In many real research and engineering application, PSO is mainly used in solving single objective problems(SOPs) because of its characteristics of the fast convergence and easy implementation. For example, Dang[3] proposed an effective PSO-based algorithm for particular vehicle routing problem. The Nayeri's study[4] demonstrates the effectiveness and robustness of the PSO approach for designing complicated multi-beam reflect array antennas where conventional design approaches fail. These promising result of SOPs provided a way to solve a large and complex problem such as an area of multiple criteria decision making, i.e. MOPs. Generally, the multi-objective PSO(MOPSO) algorithms can be classified two categories according to hybridization between evolutionary algorithms and other metaheuristics: The first class achieve Pareto optimal solutions by improving search strategies. The representatives of these algorithms include SMPSO[5], dMOPSO[6] and MMOPSO[7], etc. The second type incorporated the model of other evolutionary algorithms into PSO. Recently, a nature-inspired computing model, called membrane systems or

P systems[8], has drawn many researcher attention and was used as a framework to construct hybrid PSO because it provides an evolutionary procedure by hierarchical membrane structure, objects and rules. In real application, Dou[9] proposed a novel particle swarm optimizer on the membrane system for feature selection. The PSO is introduced to all elementary membrane as sub-algorithms using hierarchy and message mechanism of P system. ZHANG[10] put forward a cloud computing resource scheduling algorithm based on Particle Swarm Optimization algorithm and Membrane Computing theory. Moreover, previous researches in hybrid PSOs for MOPs such as non-dominated sorting PSO(NSPSO)[11] and PS-EA[12] is very competitive with existing multi-objective evolutionary algorithms(MOEAs). However, these algorithms based on membrane system mainly focused on SOPs though they can achieve better result and improve the performance of PSO. Motivated by the successful hybrid PSO and membrane system, this paper proposes a multi-objective particle swarm optimization (PSO) algorithm based on the framework of membrane system(PMOPSO). Comparing with hybrid strategies between PSO and EAs, the proposed PMOPSO has the following feature.

1)According to the hierarchical structure of membrane system, the solutions are updated parallel in the elementary membranes by using the concepts of personal best and global best of PSO. This strategy can maintain the search diversity at the same time.

2)Using theory of cell-like P system, the particle objects consisting of real-valued strings, the rules

<sup>a</sup> Corresponding author: [cctw33@126.com](mailto:cctw33@126.com)

composed of the velocity update of particles, selection for particles and communication-like rules that transform the objects between elementary membrane and skin membrane.

3) In the skin membrane, non-dominated sorting and crowding distance mechanism is used to improve the non-dominated solutions, just like external archive found in many EAs. This can accelerate the convergence to the Pareto front.

The rest of this paper is organized as follows. Section 2 introduces the multiobjective optimization and membrane system(P system) theory. The details of proposed PMOPSO are described in Section 3. The experimental results are demonstrated in Section 4. Finally, Section 5 draw some conclusions of this paper.

## 2 MULTIOBJECTIVE OPTIMIZATION AND MEMBRANE SYSTEM

### 2.1 Multi-objective optimization problems

Most problems in nature have several objectives to be satisfied. These objective functions form a mathematical description of performance criteria which are usually in conflict with each other. The term optimization means finding such a solution which would give the values of all the objective functions acceptable to the decision maker. Hence, a multi-objective optimization problem(MOP) can be stated as follows.

$$\begin{aligned} \min \quad & \mathcal{F}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})] \\ \text{subject to} \quad & \mathcal{G}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \dots, g_J(\mathbf{x})] \geq 0 \\ & \mathcal{H}(\mathbf{x}) = [h_1(\mathbf{x}), h_2(\mathbf{x}), \dots, h_K(\mathbf{x})] = 0 \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, i = 1, \dots, N \end{aligned} \quad (1)$$

where

- $X = (x_1, x_2, \dots, x_N)^T$  is the vector of the  $N$  decision variables.
- $M$  is the number of objectives  $f_i$
- $J$  inequality and  $K$  equality constraints
- $x_i^{(L)}$  and  $x_i^{(U)}$  are respectively the lower and upper bound for each decision variables  $x_i$

In multiobjective optimization problems, there are a set of solutions called Pareto solution set, which partially described the solution as follows.

**Definition 1**, feasible solution:

Given a solution vector  $x \in \theta$ , if it satisfies the  $g(x), \mathcal{H}(x)$  constraints for all  $J$  and  $K$ , it is defined as a feasible solution, else it is a infeasible solution.

**Definition 2**, variable domination:

For  $a = [a_1, a_2, \dots, a_n]^T$  and  $b = [b_1, b_2, \dots, b_n]^T$ , if  $\forall i \in \{1, 2, \dots, M\}, f_i(a) \leq f_i(b), \exists j \in \{1, 2, \dots, M\}, f_j(a) < f_j(b)$ , we express as  $b < a$ , say that  $b$  is dominated by  $a$ .

**Definition 3**, Pareto optimal set:

if  $x^* = \{x \in R^n | \neg \exists x' \in R^n, x' < x\}$ ,  $x^*$  is called a global Pareto optimal set expressed as  $PS^*$ . Where  $R^n$  is the decision space of solution set.

**Definition 4**, Pareto optimal front:

The boundary defined by the set of all point mapped from the Pareto optimal set is called the Pareto- optimal front. It is defined as  $PF^* = \{f(x^*) | x^* \in PS^*\}$ .

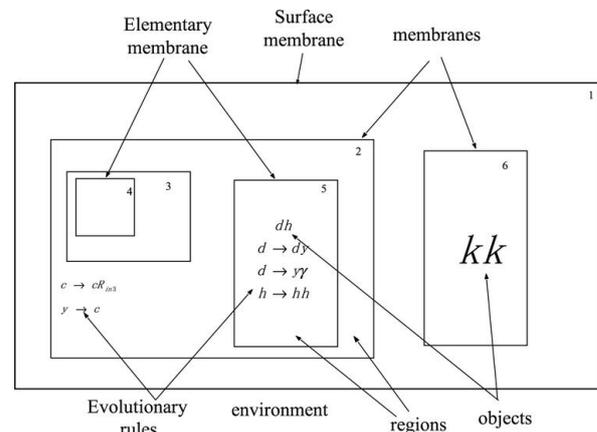
**Definition 5**, non-dominated solution:

In each generation of MOEAs, the optimal solution in evolution population is defined as non-dominated solution.

### 2.2 Membrane system

A new computational models from the study of biological cells, particularly of the cellular membranes, is first conceived by Gheorghe Păun. There are three types of membrane systems which are known as P systems, i.e. cell-like P systems[8], tissue-like P system[13,14] and neural-like P system[15,16]. A cell-like P system, which is applied in this paper, is featured as a hierarchical membrane structure as shown in Figure 1. P system can be described from Figure 1, where

- the membrane structure consists of 6 membranes;
- each membrane defines a compartment (region);
- the external compartment is called skin;
- the area outside skin is called environment (shown in the figure);
- compartments can contain other compartments, objects and rules;
- compartment (or membranes) which do not contain other compartments are called elementary;
- in the figure capital and small letters are defined as multiset of object.
- The most common types of rules are multiset rewriting rules(similar to chemical reaction, e.g  $d \rightarrow dy$ ,  $y \rightarrow c$ ) and transport rules, etc.



**Figure 1.** Structure of cell-like P system.

According to Figure 1, a cell-like P system can be formulated as the structure as follow.

$$\Pi = (V, T, \mu, \omega_1, \omega_2, \dots, \omega_m, R_1, R_2, \dots, R_m) \quad (2)$$

where:

- $V$  is the alphabet of objects and  $T \subseteq V$  is the set of catalysts.
- $\mu$  is a membrane structure, which consisting of  $m$  membranes, i.e. degree  $m$  of system. In Figure 1, the membrane structure can be equivalently represented as  $[[[[[4]_3]_5]_2]_6]_1$ .

- $\omega_1, \omega_2, \dots, \omega_m$  are strings over  $\Pi$  which denote multisets of object with the regions 1 to m of  $\mu$ .
- $R_1, R_2, \dots, R_m$  are the sets of evolution rules over  $\Pi$  with the regions of  $\mu$ .

A P system is a computing model which abstracts from the way the alive cells process chemical compounds in their compartmental structure. In short, in the regions defined by a membrane structure we have objects which evolve according to given rules. The objects can be described by symbols or by strings of symbols (in the former case their multiplicity matters, that is, we work with multisets of objects placed in the regions of the membrane structure. Hence, as a modeling framework, membrane systems is rather adequate for combining other EAs to construct a hybrid MOEAs applied to approximate optimization, having many attractive features: easy understandability, scalability and programmability, inherent compartmentalization and ability to handle discrete data, etc.

### 3 PMOPSO

For brevity we omit the introduction of PSO that can be referred to literature [17] and [3]. PMOPSO is developed with the hierarchical framework of cell-like P system, the objects consisting of the particles in the swarm which are moved around in the search-space according to a few simple formulae, the rules composed of PSO-inspired evolutionary rules and evolutionary rules of P system, and replacement operation.

The P system for MOPSO describes as follow:

(i) there exists a membrane structure  $[0[1]_1, [2]_2, \dots, [m]_m]_0$  where m elementary membranes contained in the skin membrane, denoted as 0.

(ii) an alphabet that consists of all N real-valued particles are encoded as strings in the swarm.

$$s_{i,j} = (s_{\max,j} - s_{\min,j}) * rand() + s_{\min,j} \quad (3)$$

where  $N \geq i \geq 1$  and  $D \geq j \geq 1$ , N is the number of object, i.e. particles.  $s_{i,j}$  represents j-th dimension of i-th object. The entry  $s_{\min,j}$  and  $s_{\max,j}$  denote respectively minimum and maximum value of j-column in matrix s. The function rand() generates uniformly distributed numbers between 0 and 1.

(iii) real-valued strings as the set of terminal symbols T.

(iv) initial multisets  $w_0 = \lambda$

$$\begin{aligned} w_1 &= x_1 x_2, \dots, x_{n_1} \\ w_2 &= x_{n_1+1} x_{n_1+2}, \dots, x_{n_2} \\ &\dots \\ w_m &= x_{n_{(m-1)+1}} x_{n_{(m-1)+2}}, \dots, x_{n_m} \end{aligned}$$

where  $x_i \in [1, n]$ ,  $n = \sum_{j=1}^m n_j$ , n is the total number of individuals.

(v) rules are defined as

(a) evolution rules: In each of the membrane 1 to m, the evolutionary process of object is based on the PSO algorithm.

(b) communication rules: the forms of rule (w, in) and (w, out) move objects of multiset w inside (from skin membrane to each of elementary membrane) or outside (from each of elementary membrane to skin membrane) a membrane.

(c) rewriting rules: the type  $u \rightarrow v$  similar chemical reaction is the replacement operation which is to replace some objects(local Pareto solutions) in multiset u by mutation formula as follows:

$$S_{(t+1,j)} = S_{(t,j)} + 0.1 * (S_{(\max,j)} - S_{(\min,j)}) * rand() \quad (4)$$

where the  $S_{(t+1,j)}$  denotes the j-dimensional value of object S at the t+1 generation. The other parameters are the same as formula (3).

This pseudocode algorithm of PMOPSO can be summarized in Figure 2.

```

Begin
    t ← 1
(i) Initialize the membrane structure and string objects
(ii) While (not termination condition) do
(iii) assign string objects into each membrane
(iv) Perform MOPSO in elementary membranes
(v) Execute the communication rule (b)
(vi) Calculate crowding distance
(vii) Implement non-dominated sorting
(viii) Perform rewriting rule (c) in skin membrane
(ix) Execute the communication rule (b)
    t ← t + 1
END
END
    
```

**Figure 2.** Pseudocode algorithm for PMOPSO

In Figure 2, each step of algorithm is described in detail as follows:

(i) Initialization: the operation  $[ ]_0 \rightarrow [[ ]_1, [ ]_2, \dots, [ ]_m]_0$  makes a membrane to be divided m+1 membranes, i.e. a skin membrane and m elementary membrane. Moreover, N real-valued particles are formed as string objects by formula (3).

(ii) Setting the maximal iteration as termination condition.

(iii) Using non-dominated sorting mechanism in NSGA-II, the string objects are assigned to the m elementary membranes by formula (5)

$$\begin{cases} \omega = sort(\omega) \\ \omega = \{\omega_1, \omega_2, \dots, \omega_m\} \\ n = \frac{sizeof(\omega)}{m} \end{cases} \quad (5)$$

where  $\omega$  is the multisets of string objects. The function sort() returns a sequence of solutions  $\omega$  using non-dominated sorting method and then the rule (w, in) is implemented on  $\omega$ . The value n is the number of string object in each membrane.

(iv) The pseudocode algorithm of MOPSO implemented in this step is briefly outlined in Figure 3.

```

initializeSwarm()
intializeArchive()
determinQuality()
generation = 0
while generation < maxGerrations do
    for each particle do
        select()
        updatePosition() // flight (Formulas. 6 and
7)
        evaluation()
        updatePbest()
    end for
    updateArchive()
    determineQuality()
    generation ++
end while
return Archive()
    
```

**Figure 3.** Pseudocode algorithm for MOPSO

At the start of MOPSO in each elementary membrane, the  $n$  string objects (i.e. particles in PSO) is used as initial swarm (i.e. the population of solutions in PSO). An initialized external archive is used to store the non-dominated string objects in elementary membrane. After initialization of external archive, the function `determinQuality()` has to calculated to select one best object in entire string objects. In the loop of the MOPSO, the flight of each particle is performed after a string object has been selected by the function `select()`. The way in which MOPSO updates particle  $x_i$  at the generation  $t$  is through the formula:

$$x_i(t) = x_i(t-1) + v_i(t) \quad (6)$$

where the factor  $v_i(t)$  is known as velocity and it is given by

$$v_i(t) = \omega * v_i(t-1) + C1*r1(x_{pbest_i} - x_i) \quad (7)$$

$$+ C2*r2(x_{gbest_i} - x_i)$$

where  $w$  is the inertia weight of the particle and controls the trade-off between global and local experience,  $r1$  and  $r2$  are two uniformly distributed random numbers in the range  $[0,1]$ , and  $C1$  and  $C2$  are the learning factors which control the effect of the personal and global best particles. And then each particle is evaluated and updated by two "best" values  $x_{pbest_i}$  and  $x_{gbest_i}$ . After each iteration, the set of solutions is updated and the quality measure is calculated. Finally, the MOPSO returned the set of results in the archive.

(v) for the elementary membrane  $i$ , the best  $n$  individuals from archive  $i$  is a sub-population denoted as  $\omega_i(t)$ ,  $i = 1, 2, \dots, m$ . And then the communication rule ( $a$ , out),

$a \in \omega_i(t)$ , is executed in a metaphorical way that each object  $a$  in  $\omega_i(t)$  is sent to the skin membrane. Moreover, the step (ix) is the communication rule ( $a$ , in) contrary to step (v) that move individual from skin membrane to elementary membrane.

(vi) according to the mechanism of diversity-preserving, the crowding distance is calculated as the

sum of individual distance values corresponding to each objective.

(vii) After normalizing the crowding distance for each non-dominated solution, the fast non-dominated sorting procedure which returns a list of the non-dominated fronts.

(viii) Rewriting rules are a replacement operation using the formula (4) to obtain better string objects in skin membrane.

## 4 Experiment

### 4.1. Parameters of bench algorithms

In the experiment, the bench algorithm to be compared with proposed algorithm consist of MOPSO, dMOPSO, SMPSO, MMOPSO, MOEA/D[18], SPEA2[19], PESA2[20] and NSGAI[21]. For PSO-based hybrid algorithm, the inertia weight  $\omega = 0.5$ , the learning factors (or acceleration coefficients)  $C1 = C2 = 2$ . The size of archive is 100 for bi-objective test functions and 200 for DTLZ test suit.

(1) Population size: 100 for bi-objective test function and 200 for three-objective DTLZ test series.

(2) Crossover probability for PESA2 and SPEA2 is 0.7 and 0.5 respectively.

(3) Mutation probability for PESA2 and SPEA2 is  $1/D$ , where  $D$  is the dimension of the decision variable.

(4) Evaluations for different MOPs is 10000 times for bi-objective test function and 100000 times for DTLZ test function.

(5) Independent runs for calculating the mean and variant of IGD of each algorithm is 30 times.

(6) The number of elementary membrane is 10 ( $m=10$ ).

(7) The parameters in this paper for solving the MOPs can be obtained from the literatures in reference.

### 4.2 Experiment result for bi-objective problems

In order to test the PMOPSO performance, we use the two-objective functions such as KUR, ZDT1, ZDT2 and ZDT3 to conduct the experiments. These test instances have been widely used in testing performance of MOPs. In the experiment, we investigate the performance proposed PMOPSO by comparisons of three bench algorithms, i.e. MOPSO, PESA2 and SPEA2 test on four instances. The experimental results of the three bench algorithms for solving the ZDT1 and KUR test function are shown in figure 4 and 5. And then the inverted generational distance(IGD), which is a metric for assessing the quality of approximations to the Pareto front, is used to evaluate the performance of the algorithms for solving all instances. After 30 independent tests, the statistical results are shown in Table 1

As shown in figure 4, KUR is a classic MOPs that has three discontinuous Pareto optimal front regions, which are an isolated point, a concave region and a convex region. Compared with MOPSO, the PESA2, SPEA2 and PMOPSO algorithm have the better uniform

distribution which overlap with true Pareto front. Figure 5 shows that the PMOPSO and MOPSO have the better performance than the SPEA2 and PESA2 on the test problems ZDT1 that have 30 decision variables each and

the former has a convex Pareto optimal front and the latter has a concave Pareto optimal front.

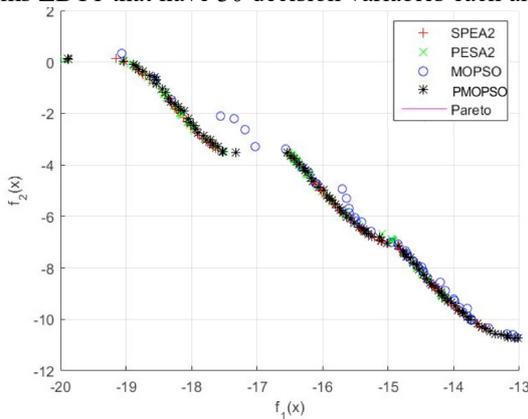


Figure 4. the results of four different algorithms on KUR

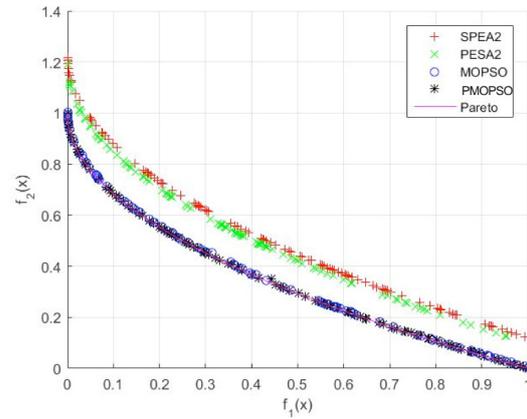


Figure 5. the results of four different algorithms on ZDT1

Table 1. IGD performance statistics of 4 algorithms on 5 test functions

Algorithms \ MOPs	PMOPSO		MOPSO		SPEA2		PESA2	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
KUR	5.96E-02	2.25E-01	2.25E+00	4.50E+00	5.71E-02	3.46E-02	6.12E-02	2.75E-03
ZDT1	4.84E-02	1.34E-01	5.63E-02	2.10E+00	1.31E-01	1.03E-01	1.05E-01	1.47E-01
ZDT2	4.96E-01	9.75E+02	6.23E-02	4.10E-01	1.52E-01	5.12E-01	1.41E-01	3.51E-01
ZDT3	5.96E-02	2.50E-01	3.79E-01	9.50E-02	4.50E+00	3.54E-01	4.50E+00	2.51E-01
ZDT6	4.36E-02	2.15E-01	5.29E-02	6.50E-01	5.68E-02	5.51E-01	5.41E-02	2.51E-01

According to the IGD performance data shown in the Table 1, the PMOPSO obtain better result as same as it was shown in Figure 4 and 5. Moreover, for ZDT2, ZDT3 and ZDT6, the proposed algorithm has achieved lower IGD value and better performance than other algorithm does.

### 4.3. Experiment result for three-objective problems

In order to further observe the experimental result of three-objective problems, we consider DTLZ test suit which was conceived for multiobjective problems with scalable fitness dimensions to investigate the

performance of proposed PMOPSO algorithm. Compared with other PSO-based multi-objective optimization algorithm including dMOPSO, SMPSO, NMPSO, MOEA/D, NSGAI on a DTLZ series function, the values of IGD are shown in Table 2.

From Table 2, the MOEA/D had the smallest values and obtained best result on DTLZ1, and DTLZ2 and DTLZ3. Overall, PSO-based hybrid algorithms including proposed PMOPSO algorithm have better performance than MOEA/D and NSGAI. Moreover, the PMOPSO obtained best convergence and diversity because the values of IGD are smaller on DTLZ4, DTLZ5, DTLZ6 and DTLZ7. Thus, PMOPSO can balance the capability of global and local searching in algorithm.

Table 2. IGD performance statistics of 6 algorithms on 7 DTLZ test functions

Test instance		PMOPSO	dMOPSO	SMPSO	MMOPSO	MOEA/D	NSGAI
DTLZ1	Mean	1.44e-04	5.79e00	2.08e00	1.66e-03	1.37e-04	1.85e-02
	Std.	2.35e-05	7.67e-01	2.57e-01	2.78e-03	3.78e-06	2.67e-03
DTLZ2	Mean	6.23e-04	1.98e-02	1.06e-03	7.22e-04	1.63e-04	4.88e-02
	Std.	4.56e-06	2.45e-04	5.42e-04	4.24e-05	5.76e-05	1.35e-04
DTLZ3	Mean	1.43e-03	3.03e+01	3.75e00	5.86e-04	1.78e-04	4.92e-02
	Std.	2.98e-05	4.16e-01	6.29e-01	2.38e-06	4.55e-05	3.27e-03
DTLZ4	Mean	4.11e-04	7.61e-03	3.02e-04	7.33e-04	9.45e-01	4.83e-02
	Std.	3.56e-05	4.65e-03	7.24e-04	1.78e-03	3.02e-01	3.32e-05
DTLZ5	Mean	3.15e-05	2.58e-03	5.18e-05	6.54e-05	2.16e-02	2.85e-03
	Std.	7.98e-06	1.98e-01	3.87e-05	3.57e-06	3.45e-04	1.27e-03

DTLZ6	Mean	<b>2.35e-06</b>	4.25e-03	3.17e-06	3.58e-06	2.26e-02	2.97e-03
	Std.	3.24e-06	3.78e-04	2.54e-06	3.28e-06	3.05e-04	1.05e-05
DTLZ7	Mean	<b>4.23e-04</b>	1.33e-03	1.76e-03	1.28e-03	1.03e-01	5.63e-02
	Std.	1.98e-06	2.45e-04	9.24e-04	2.56e-02	3.91e-02	1.27e-02

## 5 Conclusion

In order to solve the complex MOPS in reality, a PMOPSO algorithm based on membrane system is proposed. According to the hierarchical structure of membrane framework, the objects and rules of P system, the PSO approach is used in elementary membranes to execute multiple search strategy. And non-dominated sorting and crowding distance is used in skin membrane for improving speed of convergence and maintaining population diversity by evolutionary rules. Compared with other multi-objective optimization algorithm including MOPSO, dMOPSO, SMPSO, MMOPSO, MOEA/D, SPEA2, PESA2, NSGAI on a benchmark series function, the experimental results indicate that the proposed algorithm is not only feasible and effective but also have a better convergence to true Pareto front.

Because of the complexity of MOPs, the further researches will be conducted in:

- (1) Comparing the time complexity of PMOPSO with other algorithms.
- (2) Parallelizing PMOPSO based on membrane algorithm according to the features of parallel and distributed computing.
- (3) Investigating effects of different parameters chosen in PMOPSO.

## Acknowledgment

This work of was partially supported by the Nation Natural Science Foundation of China (No. 61461051,71462036).

## References

1. Jones, D. F., Mirrazavi, S. K., & Tamiz, M. Multi-objective meta-heuristics: An overview of the current state-of-the-art. *European Journal of Operational Research*, 137, 1-9. (2002).
2. Kennedy J., Eberhart R. "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks. IV.* pp. 1942–1948. (1995).
3. Dang, D. C., Guibadj, R. N., & Moukrim, A. An effective PSO-inspired algorithm for the team orienteering problem. *European Journal of Operational Research*, 229, 332–344. (2013).
4. Nayeri, P., Yang, F., & Elsherbeni, A. Z. Design of single-feed reflectarray antennas with asymmetric multiple beams using the particle swarm optimization method. *IEEE Transactions on Antennas and Propagation*, 61, 4598–4605. (2013).
5. Nebro, A. J., Durillo, J. J., Garcia-Nieto, J., Coello Coello, C. A., Luna, F., & Alba, E. SMPSO: A new PSO-based metaheuristic for multi-objective optimization. In *Proceedings of IEEE symposium on computational intelligence in multi-criteria decision-making*, pp. 66–73. (2009).
6. Martinez, S. Z., & Coello Coello, C. A. A multi-objective particle swarm optimizer based on decomposition. In *Proceedings of the 13th annual genetic and evolutionary computation conference*, pp. 69–76. (2011).
7. Q. Lin, J. Li, Z. Du, J. Chen, Z. Ming. "A novel multi-objective particle swarm optimization with multiple search strategies," *Eur. J. Oper. Res.*, vol.247, no. 3, pp732-744 (2015).
8. Gheorghe Păun, Grzegorz Rozenberg. *A guide to membrane computing*[J].*Theoretical Computer Science*. 287:73-100,(2002)
9. Dou Zengfa, Gao Lin. Feature selection in conditional random fields using a membrane particle swarm optimizer. *Journal of XIDIAN university*, vol.39 no.5 107-112, (2012).
10. ZHANG Qunhui, LI Renfa. Cloud resource scheduling based On improved particle swarm optimization algorithm by membranecomputing. *Computer Engineeringand Applications*, 49(20):40-44. (2013)
11. X. D. Li, "A non-dominated sorting particle swarm optimizer for mul- tiobjective optimization," in *Proc. Genet. Evol. Comput.*, vol. LNCS 2723, pp. 37-48. (2003).
12. D. Srinivasan and T. H. Seow, "Particle swarm inspired evolutionary algorithm (PS-EA) for multiobjective optimization problem," in *Proc. Congr. Evol. Comput.*, pp. 2292–2297. (2003)
13. Martín-Vide, C., Păun, G., Pazos, J. and Rodríguez-Patón, Alfonso. *Tissue P systems*. *Theor. Comput. Sci.* 296 (2003), 295-326.
14. Freund,R., Păun, G. and Pérez-Jiménez, M.J. *Tissue P systems with channel states*. *Theor. Comput. Sci.* 330 (2005), 101-116.
15. Leporati,A.,Zandron,C.,Ferretti,C. and Mauri,G. On the computational power of spiking neural P systems. *Int. J. Unconv. Comput* 5 (2009), 459–473.
16. García-Arnau,M.,Pérez,D.,Rodríguez-Patón,A. and Sosik,P. Spiking neural P systems: stronger normal forms. *Int. J. Unconv. Comput* 5 (2009), 411–425.
17. Timothy Ganesan,Pandian Vasant,Irraivan Elamvazuthy. A hybrid PSO approach for solving non-convex optimization problems[J]. *Archives of Control Sciences*,(2012),22(1).
18. Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algo- rithm based on decomposition," *IEEE Trans. Evol. Comput.*, (2007).vol. 11, pp. 712–731, Dec.
19. E. Zitzler, M. Laumanns, and L. Thiele, *SPEA2: Improving the strength Pareto evolutionary algorithm* *Comput. Eng. Networks Lab., Swiss Fed.*

- Inst. Technol., Zurich, Switzerland, (2001), Tech. Rep. 103.
- 20 J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, (2000), vol. 8, no. 2, pp. 149-172.
  - 21 K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.*, (2002), vol. 6, pp. 182-197, Apr.