

# A skin membrane-driven membrane algorithm for many-objective optimization

Zhangxiao Li<sup>1</sup> · Lei Zhang<sup>2</sup> · Yansen Su<sup>2</sup> · Jun Li<sup>2</sup> · Xun Wang<sup>3</sup>

Received: 8 July 2016 / Accepted: 25 October 2016 / Published online: 15 November 2016  
© The Natural Computing Applications Forum 2016

**Abstract** Many-objective optimization problems refer to problems that hold more than three conflicting objectives, which are more challenging than those with two or three objectives. Membrane computing models, usually termed P systems, are a class of living cell-inspired computing models, which provide a rich framework for solving a variety of challenging problems. In this paper, a membrane computing model-based algorithm is proposed for many-objective optimization. Specifically, the population in the skin membrane is divided into two subpopulations, one used for guiding the convergence of populations in the internal membrane, while the other taking charge of the diversity of populations. Experimental results on benchmark test problems for many-objective optimization indicate the superiority of the developed algorithm over existing evolutionary many-objective optimization algorithms and P systems based multi-objective optimization algorithms.

**Keywords** Membrane computing · Many-objective optimization · Membrane algorithm · Archive

## 1 Introduction

Multi-objective optimization problems (MOPs) are widely known as a challenging topic in the domain of optimization, since often a set of trade-off solutions instead of a single optimal solution are obtained due to the conflicting nature between the objectives. In the past years, a great many popular algorithms based on different ideas have been developed to solve MOPs. Two representatives belonging to this category are the fast and elitist multi-objective genetic algorithm (NSGA-II) [4] and the indicator-based evolutionary algorithm (IBEA) [66]. Recently, much attention has been paid to an even more challenging subset of MOPs, where the MOPs hold four or more objectives, also known as many-objective optimization problems (MaOPs) [7, 43, 62]. The MaOPs are widely present in various applications [9, 20]; however, most existing algorithms for MOPs with two or three objectives are hard to achieve a satisfied performance due to the phenomenon called dominance resistance [9]. For solving MaOPs, many bio-inspired computing models and ideas have been introduced to design algorithms, e.g., the multi-objective evolutionary algorithm based decomposition (MOEA/D) [53], the improved two-archive algorithm (Two\_Arch2) [22], grid-based evolutionary algorithm (GrEA) [50], hypervolume-based evolutionary algorithm (HypE) [1], and the knee point-driven evolutionary algorithm (KnEA) [55].

*Membrane computing* is a branch of natural computing whose aim is to abstract new algorithms or models from living cells. For simplicity, we also call the computing models obtained in membrane computing as P systems. Since the first P system was initiated by Păun [30], many novel models of P systems have been developed based on different biological mechanisms related to cells, e.g., cell-

---

✉ Xun Wang  
Wangsyun@upc.edu.cn

<sup>1</sup> Department of Accounting, Huishang Vocational College, Hefei 230032, China

<sup>2</sup> School of Computer Science and Technology, Anhui University, Hefei 230039, China

<sup>3</sup> College of Computer and Communication Engineering, China University of Petroleum, Qingdao 266580, China

like P systems [33, 63], spiking neural P systems (SN P systems, for short) [16, 38, 57], and tissue-like P systems [10, 26, 56]. It was shown that P systems have theoretically powerful capability in handling complex information [38, 39, 52, 59]. Most variants of P systems can achieve the same computing power as Turing machines or other equivalent machines [18, 58], and some P systems have also been shown to theoretically solve NP-hard problems [17, 19, 21]. Due to the promising features, P systems have been applied to deal with a variety of challenging problems in different research areas, such as modeling economic processes [32], segmenting images [8], and fault diagnosis [12]. Readers who are interested in membrane computing can refer to the books [31, 34] and for more information refer to the website <http://ppage.psystems.eu/>.

Among various applications of membrane computing, one of the most active areas is to apply P systems to design algorithms for solving computationally hard optimization problems. The obtained algorithms are usually called membrane computing model-based algorithms, also known as membrane algorithms for short. The first membrane algorithm was developed by Nishida [27], where the cell-like P systems were used and objects in membranes evolved by the tabu search. Empirical results demonstrated that this algorithm was superior over existing algorithms in solving traveling salesman problems. Since then, a large number of membrane algorithms have been suggested to handle other single-objective optimization problems on the basis of different kinds of P systems and evolution strategies [11, 23, 35, 37, 42, 46, 47, 54, 65]. For example, Zhang et al. [54] developed a membrane algorithm to handle knapsack problems, where cell-like P systems using only one level membrane structure and quantum-inspired evolution strategy were used. Xiao et al. suggested a cell-like optimization P system for designing DNA sequence in DNA computing [23] and an SN P optimization system was developed in [42] to approximately solve combinatorial optimization problems.

There are also a little work that focuses on designing effective P system-based algorithms to handle MOPs. The first multi-objective membrane algorithm was developed to solve MOPs with two objectives by Huang et al. [14], where three subsystems of P systems were used, one subsystem for optimizing the two objectives simultaneously and the other two for optimizing the two objectives independently. To solve multi-objective knapsack problems, Zhang et al. [61] suggested a multi-objective membrane algorithm MOMA by using the quantum-inspired evolutionary algorithm, which achieved a better performance than the existing algorithms designed for solving the problem. In the past four years, more complex information of P systems start to be considered for designing multi-objective membrane algorithms. Cheng et al. [6] developed

a multi-objective optimization algorithm based on tissue-like P systems with each pair of cells communicating at every generation. Liu et al. [24] proposed a multi-objective optimization algorithm by using P systems with division and dissolution rules. Recently, a multi-objective membrane algorithm, termed SMG-MOMA, was suggested in [60], where a skin membrane strategy was proposed to guide the evolution of populations in the internal membrane. It is worth noting that the multi-objective membrane algorithms mentioned above are all suggested to solve MOPs with two or three objectives, and they will not achieve a competitive performance when they are applied to solve MaOPs.

In this paper, we suggest a skin membrane-guiding membrane algorithm, termed SMG-MaOMA, specially tailored for solving MaOPs. In SMG-MaOMA, the population in the skin membrane is divided into two subpopulations, which are used to guide the convergence and diversity of populations in the internal membrane, respectively. The subpopulation related to convergence is used to accelerate the convergence of populations in the internal membrane, while the one related to diversity guarantees the diversity of these populations. In this way, the SMG-MaOMA can keep well the balance between diversity and convergence of populations in evolutions. Experimental results on 16 benchmark many-objective optimization problems from Deb, Thiele, Laumanns and Zitzler's DTLZ and the walking fish group WFG test suites, show that the SMG-MaOMA performs better than four well-known evolutionary many-objective optimization algorithms MOEA/D [53], HypE [1], GrEA [50] and Two\_Arch2 [22], and one P system-based multi-objective optimization algorithm SMG-MOMA on most test problems used in this paper.

The rest of the paper is organized as follows. We first present the details of the proposed algorithm in Sect. 2. Then, the proposed algorithm is compared with five popular many-objective optimization algorithms and empirical results are analyzed in Sect. 3. In Sect. 4, we conclude the paper and give some remarks.

## 2 The proposed membrane algorithm for many-objective optimization

In this section, we will describe the details of the proposed many-objective membrane algorithm, SMG-MaOMA. The proposed SMG-MaOMA adopts the one level membrane structure of cell-like P systems, which is the known simplest structure used in existing membrane algorithms. It is necessary to stress that there are some membrane algorithms which also used a skin membrane strategy; however, they hold quite different ways for guiding the skin

membrane. The skin membrane-guiding strategy proposed in SMG-MaOMA is specially tailored for solving MaOPs. In what follows, we first present the main framework of the proposed algorithm.

**Algorithm 1** The Main Framework of SMG-MaOMA

**Input:**  $CA$  (sub-population in skin membrane for convergence),  $DA$  (sub-population in skin membrane for diversity),  $P_1, P_2$  (populations in internal membranes)  $N$  (population size of  $CA, DA$ ),  $N_1$  (population size of  $P_1, P_2$ )

**Output:**  $DA$

- 1:  $P_1, P_2 \leftarrow Initialize(P_1, P_2, N_1)$ ;
- 2:  $CA, DA \leftarrow Initialize(CA, DA, N)$ ;
- 3: **While** (termination criterion not satisfied) **do**
- 4:  $P'_1, P'_2 \leftarrow Genetic\_Operation(P_1, P_2)$ ;
- 5:  $P \leftarrow P'_1 \cup P'_2$ ;
- 6:  $CA \leftarrow CA\_Update(CA, P)$ ;
- 7:  $DA \leftarrow DA\_Update(DA, P)$ ;
- 8:  $P_1, P_2 \leftarrow Skin\_Guide\_CA(CA, P_1)$ ;
- 9:  $P_1, P_2 \leftarrow Skin\_Guide\_DA(DA, P_2)$ ;
- 10: **EndWhile**

**2.1 The main framework of the proposed algorithm**

Algorithm 1 presents the main framework of SMG-MaOMA, which contains the following steps. First, two populations  $P_1$  and  $P_2$  with a size of  $N_1$  are randomly generated in the internal membrane. Second, two populations  $CA$  and  $DA$  with a size of  $N$  are randomly generated in the skin membrane, which are used for convergence and diversity of populations in the internal membrane, respectively. Third, a genetic operation is applied to population  $P_1$  and to generate  $N_1$  offspring individuals and to  $P_2$  to generate  $N_1$  offspring individuals. Fourth, an indicator-based approach developed in [66] is used to update subpopulation  $CA$  by using the population  $P$  combined with offspring populations of  $P_1$  and  $P_2$  in the internal membrane. Fifth, an environmental selection based on  $L_p$  crowding distance suggested in [22] is used to update population  $DA$  by using the combined population  $P$  in the internal membrane. Finally, the skin membrane-guiding strategy proposed in this paper is used to select individuals for next population  $P_1$  from the combined populations of  $CA$  and  $P_1$ , and for  $P_2$  from  $DA$  and  $P_2$ . The population  $P_1$  selects the individuals with better convergence from the combined population of  $CA$  and  $P_1$ , whereas the  $P_2$  enjoys those with better diversity from  $DA$  and  $P_2$ . We repeat this procedure until a termination condition is satisfied.

**2.2 The proposed skin membrane-guiding strategy**

The main idea of the proposed skin membrane-guiding strategy is as follows. It is firstly to divide the population in the skin membrane into two subpopulations  $CA$  and  $DA$ , and then,  $CA$  is used for guiding the convergence of population  $P_1$  in the internal membrane;  $DA$  is used to take charge of the diversity of population  $P_2$  in the internal membrane.

Specifically, the next population of  $P_1$  can be obtained from the population combined  $CA$  and the current population, with a convergence-related environmental selection criterion from [66]. The next population of  $P_2$  is obtained from the combined population of  $DA$  and  $P_2$  by a diversity-related selection criterion. It is shown in Algorithms 2 and 3 the detailed procedures of subpopulations  $CA$  and  $DA$  guiding strategies in SMG-MaOMA, respectively.

**Algorithm 2** *Skin\_Guide\_CA(CA, P1)*

**Input:**  $CA$  (sub-population in skin membrane for convergence optimization),  $P_1$  (populations in internal membranes)

**Output:**  $P_1$

- 1:  $Q \leftarrow CA \cup P_1$ ;
- 2: **While**  $|Q| > |P_1|$  **do**
- 3: Calculate the fitness value of each solution in  $Q$  by the formula
- 4:  $F(x) = \sum_{y \in Q \setminus \{x\}} -e^{-I(\{y\}, \{x\})/\kappa}$ ;
- 5: Delete the solution with the smallest fitness value from  $Q$ ;
- 6: **EndWhile**
- 7:  $P_1 \leftarrow Q$ ;

**Algorithm 3** *Skin\_Guide\_DA(DA, P2)*

**Input:**  $DA$  (sub-population in skin membrane for diversity optimization),  $P_2$  (populations in internal membranes)

**Output:**  $P_2$

- 1:  $Q \leftarrow$  extreme solutions in  $DA \cup P_2$ ;
- 2: **While**  $|Q| < |P_2|$  **do**
- 3:  $x \leftarrow \operatorname{argmax}_{x \in DA \cup P_2 \setminus Q} \min_{y \in Q} dis_p(x, y)$ ;
- 4:  $Q \leftarrow Q \cup \{x\}$ ;
- 5: **EndWhile**
- 6:  $P_2 \leftarrow Q$ ;

As described in Algorithm 2, the subpopulation  $CA$  guiding population  $P_1$  is performed as follows. First, we calculate the fitness value of each solution in the combined population of  $CA$  and  $P_1$  by formula  $F(x) = \sum_{y \in Q \setminus \{x\}} -e^{-I(\{y\}, \{x\})/\kappa}$  proposed in [66], where  $Q = CA \cup P_1, I(\{y\}, \{x\}) = \max\{f_i(y) - f_i(x) | 1 \leq i \leq M\}, f_i(x)$  is the  $i$ th objective value of solution  $x, M$  is the number of objectives and  $\kappa$  is a scaling parameter with the restriction  $\kappa > 0$ . The  $I(\{y\}, \{x\})$  defines the maximum cost that solution  $y$  dominates solution  $x$ , and the larger fitness value of a solution, the better convergence the solution. With the fitness values of the solutions, solutions with the smallest fitness value in the combined population of  $CA$  and  $P_1$  are removed one by one until the size of population  $P_1$  is reached.

The subpopulation  $DA$  guides the population  $P_2$  in the following way as shown in Algorithm 3. First, the extreme solutions with at least one objective having the worst value are selected from the population combined  $DA$  with  $P_2$  for next population of  $P_2$ . If the selected solutions cannot fill up the population  $P_2$ , then the solutions with the maximum  $L_p$  crowding distance are added to  $P_2$  one by one. This operation repeats until the size of population  $P_2$  is reached.

### 3 Experimental results

In this section, we compare the proposed SMG-MaOMA with five algorithms MOEA/D [53], HypE [1], GrEA [50], Two\_Arch2 [22], and SMG-MOMA [60], on benchmark test problems for MaOPs. The former four algorithms are popular many-objective optimization algorithms, and the last one SMG-MOMA is a membrane computing-based multi-objective optimization algorithm recently proposed in [60]. For fair comparisons, all compared algorithms are set to the recommended values. Specifically, the experimental settings are as follows.

#### 3.1 Experimental settings

##### 1. Test problems

In the experiments, a total of 16 test problems from DTLZ [51] and WFG [15] test suits are used. For the DTLZ test suit, the DTLZ1 to DTLZ7 are used, whose parameters are set as shown in Table 1. As for the WFG test suit, the WFG1 to WFG9 are adopted, whose parameters are set as those suggested in [15].

##### 2. Genetic operation

The simulated binary crossover [2] and polynomial mutation [5] are adopted to generate offspring for all considered algorithms. The distribution indexes of crossover and mutation are set to  $n_c = 20$  and  $n_m = 20$ , respectively. The crossover probability  $p_c = 1.0$  and the mutation probability  $p_m = 1/n$  are used, where  $n$  denotes the number of decision variables.

##### 3. Population size

For all algorithms, we adopt the recommended versions in the comparisons of this paper. The two algorithms Two\_Arch2 and SMG-MOMA use an archive, whose size is set to 100 as suggested in the experiments. For fair comparisons, the maximum number of solutions in subpopulations  $CA$  and  $DA$  in skin membrane is also fixed to 100. The population size of all considered algorithms is set to 100, with the only exception of MOEA/D. For MOEA/D, the population size can not be randomly set, which is determined by a set of evenly distributed weight vectors as

described in [53]. For this reason, the population size of MOEA/D was set to 120, 132, 120, 55 for 4, 6, 8, 10-objective test problems, respectively.

#### 4. Stopping condition and number of runs

For all considered algorithms, we adopt the maximum number of evaluations suggested in GrEA [50] as the termination criterion. For DTLZ2, DTLZ4, DTLZ5 and DTLZ7, the maximum number of evaluations is set to 30,000, and to 100,000 for DTLZ1, DTLZ3 and DTLZ6. As for the WFG test suite, we set the maximum number of evaluations to 100,000 for WFG1 and 75,000 for WFG2, and 25,000 for WFG3 to WFG9. For each algorithm, 20 independent runs are performed on each test instance and the median value is reported.

#### 5. Performance metrics

We evaluate the performance of algorithms by the inverted generational distance (IGD) [28] and Hypervolume (HV) [1], which are two widely used performance indicators in many-objective optimization. It is widely accepted that IGD and HV account for both convergence and diversity of the solution set found by many-objective optimization algorithms. The smaller values of IGD, the better performance of the algorithms. On the contrary, the larger values of HV indicates a better performance of the algorithms. The calculation of IGD needs a set of reference points, which are uniformly selected from the Pareto optimal fronts of test problems and 5000 reference points are used. In the calculation of HV, the Monte Carlo method is used for estimating the HV and 1,000,000 sampling points are used due to the high complexity of calculating the exact HV.

#### 3.2 Experimental results and analysis

It is shown in Table 2 that the inverted generational distance (IGD) values of the six considered algorithms on DTLZ1 to DTLZ7 with 4, 6, 8, 10 objectives averaging over 20 independent runs, where the best median value on each test instance is highlighted in boldface. The rank of the algorithms on each test instance, as well as the mean rank of each algorithm averaging over all considered DTLZ test instances are also shown in the table. From the table, we can find that the proposed SMG-MaOMA is

**Table 1** Settings of DTLZ1 to DTLZ7 test problems

Problem	Number of objectives ( $M$ )	Number of variables ( $n$ )	Parameter ( $k$ )
DTLZ1	4, 6, 8, 10	$M - 1 + k$	5
DTLZ2	4, 6, 8, 10	$M - 1 + k$	10
DTLZ3	4, 6, 8, 10	$M - 1 + k$	10
DTLZ4	4, 6, 8, 10	$M - 1 + k$	10
DTLZ5	4, 6, 8, 10	$M - 1 + k$	10
DTLZ6	4, 6, 8, 10	$M - 1 + k$	10
DTLZ7	4, 6, 8, 10	$M - 1 + k$	20

**Table 2** Inverted generational distance (IGD) values of the six compared algorithms on DTLZ1 to DTLZ7 averaging over 20 independent runs, where the best mean on each instance is highlighted in boldface

Problems	Obj.	MOEA/D	HypE	GrEA	Two_Arch2	SMG-MOMA	SMG-MaOMA
DTLZ1	4	1.03E−1[4] (5.24E−3)	1.60E−1[5] (2.52E−2)	1.71E−1[6] (1.02E−1)	4.54E−2[2] (2.05E−3)	<b>4.46E−2[1]</b> <b>(1.17E−3)</b>	6.57E−2[3] (1.28E−2)
	6	3.96E−1[5] (1.43E−1)	5.55E−1[6] (5.91E−1)	3.09E−1[4] (1.08E−1)	<b>8.80E−2[1]</b> <b>(3.36E−3)</b>	1.73E−1[3] (1.17E−1)	1.25E−1[2] (1.45E−2)
	8	2.79E−1[3] (2.27E−2)	6.43E−1[5] (4.15E−2)	3.93E−1[4] (5.86E−2)	<b>1.33E−1[1]</b> <b>(4.59E−3)</b>	2.24E+1[6] (1.46E+1)	1.65E−1[2] (5.17E−3)
	10	3.32E−1[3] (2.61E−2)	8.28E−1[5] (1.05E−1)	4.54E−1[4] (1.16E−1)	1.85E−1[2] (8.16E−3)	1.77E+2[6] (5.35E+0)	<b>1.82E−1[1]</b> <b>(4.51E−3)</b>
DTLZ2	4	2.34E−1[6] (1.89E−3)	2.55E−1[5] (4.09E−3)	<b>1.23E−1[1]</b> <b>(1.06E−3)</b>	1.39E−1[4] (2.48E−3)	1.29E−1[2] (2.68E−3)	1.35E−1[3] (3.30E−3)
	6	4.12E−1[4] (2.66E−1)	4.73E−1[5] (5.78E−2)	<b>2.54E−1[1]</b> <b>(1.52E−3)</b>	2.97E−1[3] (1.09E−2)	4.74E−1[6] (4.91E−2)	2.72E−1[2] (7.87E−4)
	8	8.08E−1[5] (3.46E−2)	6.49E−1[4] (3.60E−2)	3.95E−1[2] (1.33E−2)	4.63E−1[3] (6.46E−3)	2.21E+0[6] (2.70E−1)	<b>3.88E−1[1]</b> <b>(9.30E−3)</b>
	10	9.91E−1[5] (1.80E−2)	7.88E−1[4] (3.57E−2)	5.01E−1[2] (2.46E−3)	5.69E−1[3] (6.61E−3)	2.43E+0[6] (4.71E−2)	<b>4.50E−1[1]</b> <b>(3.60E−3)</b>
DTLZ3	4	3.55E−1[4] (6.08E−2)	4.79E−1[5] (1.78E−2)	5.61E−1[6] (3.34E−1)	<b>1.54E−1[1]</b> <b>(7.82E−2)</b>	1.61E−1[2] (1.68E−1)	3.93E−1[3] (8.22E−2)
	6	1.24E+0[3] (1.21E+0)	2.68E+0[5] (1.79E+0)	1.08E+0[2] (1.29E−1)	1.63E+0[4] (7.33E−1)	1.34E+2[6] (3.34E+1)	<b>7.35E−1[1]</b> <b>(2.11E−1)</b>
	8	1.15E+0[5] (8.49E−2)	9.08E−1[2] (3.03E−3)	4.02E+0[4] (3.00E+0)	1.89E+0[3] (6.81E−1)	1.02E+3[6] (1.54E+2)	<b>8.95E−1[1]</b> <b>(6.51E−2)</b>
	10	1.35E+0[2] (5.32E−2)	4.00E+0[4] (4.12E−3)	6.59E+0[5] (2.56E+0)	1.38E+0[3] (4.79E−2)	1.62E+3[6] (1.80E+2)	<b>9.80E−1[1]</b> <b>(1.04E−1)</b>
DTLZ4	4	4.33E−1[6] (1.96E−1)	3.78E−1[5] (1.52E−1)	1.59E−1[3] (9.85E−2)	1.69E−1[4] (6.22E−3)	1.36E−1[2] (3.39E−3)	<b>1.34E−1[1]</b> <b>(7.50E−4)</b>
	6	7.68E−1[6] (2.63E−1)	6.67E−1[4] (1.02E−1)	<b>2.55E−1[1]</b> <b>(1.51E−3)</b>	3.46E−1[3] (1.02E−2)	7.49E−1[5] (1.07E−1)	2.71E−1[2] (2.27E−3)
	8	8.12E−1[5] (7.43E−2)	6.59E−1[4] (2.63E−2)	3.82E−1[2] (9.96E−3)	5.32E−1[3] (1.39E−2)	2.09E+0[6] (4.10E−1)	<b>3.80E−1[1]</b> <b>(9.62E−3)</b>
	10	9.47E−1[5] (5.80E−2)	7.81E−1[4] (6.05E−2)	5.04E−1[2] (1.31E−2)	6.49E−1[3] (2.62E−2)	2.43E+0[6] (7.01E−2)	<b>4.47E−1[1]</b> <b>(1.32E−3)</b>
DTLZ5	4	2.72E−2[4] (1.71E−5)	2.58E−2[3] (6.76E−3)	<b>1.49E−2[1]</b> <b>(1.90E−3)</b>	3.50E−2[5] (3.71E−3)	1.25E−1[6] (2.32E−2)	2.17E−2[2] (4.08E−3)
	6	7.44E−1[5] (5.48E−3)	1.73E−1[4] (5.94E−2)	1.20E−1[3] (2.28E−2)	9.30E−2[2] (2.06E−2)	8.46E−1[6] (1.22E−1)	<b>8.83E−2[1]</b> <b>(1.22E−2)</b>
	8	<b>1.24E−1[1]</b> <b>(2.94E−3)</b>	3.10E−2[5] (4.81E−3)	2.43E−1[4] (4.06E−2)	1.60E−1[3] (4.48E−2)	1.23E+0[6] (2.00E−1)	1.49E−1[2] (1.70E−2)
	10	7.45E−1[5] (2.06E−2)	<b>3.57E−2[1]</b> <b>(8.02E−3)</b>	5.78E−1[4] (1.05E−1)	1.74E−1[2] (2.48E−2)	1.20E+0[6] (3.40E−1)	2.49E−1[3] (2.26E−2)
DTLZ6	4	1.43E−1[4] (2.90E−2)	2.45E−1[6] (1.58E−2)	3.16E−2[2] (6.19E−3)	3.84E−2[3] (4.14E−3)	1.84E−1[5] (3.48E−2)	<b>2.44E−2[1]</b> <b>(4.43E−3)</b>
	6	7.45E−1[3] (5.87E−4)	5.67E+0[6] (5.79E−1)	1.38E+0[4] (5.40E−1)	1.01E−1[2] (1.49E−2)	4.40E+0[5] (5.55E−1)	<b>9.77E−2[1]</b> <b>(2.93E−2)</b>
	8	1.93E−1[2] (3.13E−2)	6.70E+0[5] (5.83E−1)	3.05E+0[4] (1.58E+0)	<b>1.59E−1[1]</b> <b>(2.73E−2)</b>	9.77E+0[6] (2.74E−1)	2.36E−1[3] (2.32E−2)
	10	7.26E−1[5] (6.37E−5)	6.91E+0[4] (8.17E−1)	4.93E+0[3] (2.05E+0)	<b>1.66E−1[1]</b> <b>(1.40E−2)</b>	9.75E+0[6] (6.80E−1)	4.84E−1[2] (1.19E−1)



**Table 2** continued

Problems	Obj.	MOEA/D	HypE	GrEA	Two_Arch2	SMG-MOMA	SMG-MaOMA
DTLZ7	4	1.01E+0[5] (1.56E−1)	1.26E+0[6] (1.14E−2)	1.80E−1[3] (7.64E−3)	<b>1.44E−1[1]</b> <b>(4.57E−3)</b>	1.51E−1[2] (4.47E−3)	2.35E−1[4] (1.01E−1)
	6	1.04E+0[5] (1.27E−1)	3.89E+0[6] (1.94E−1)	7.28E−1[4] (2.34E−2)	<b>5.68E−1[1]</b> <b>(1.12E−2)</b>	5.34E−1[2] (1.17E−2)	6.87E−1[3] (2.31E−2)
	8	1.91E+0[5] (1.94E−1)	4.17E+0[6] (1.77E−1)	1.19E+0[4] (6.94E−2)	<b>9.73E−1[1]</b> <b>(2.74E−2)</b>	1.15E+0[3] (1.07E−1)	1.14E+0[2] (2.41E−1)
	10	3.36E+0[5] (4.42E−1)	8.24E+0[6] (9.76E−2)	1.34E+0[2] (5.03E−2)	<b>1.21E+0[1]</b> <b>(1.43E−2)</b>	2.28E+0[4] (8.10E−1)	1.39E+0[3] (3.75E−2)
Average rank		4.29	4.64	3.11	2.36	4.71	1.89

superior over the four popular many-objective optimization algorithms (namely, MOEA/D, HypE, GrEA and Two\_Arch2) and a recently developed membrane computing model-based multi-objective optimization algorithm, SMG-MOMA. The proposed SMG-MaOMA achieves an average rank of 1.89, which is much better than the second best algorithm Two\_Arch2 who holds an average rank of 2.36 on all considered test instances.

From the table, we can find that the Two\_Arch2 method also holds a competitive performance on the test instances, whose superiority seems to be enhanced as the number of objectives increases on DTLZ test problems. The promising performance of the Two\_Arch2 method may be partly attributed to the fact that the algorithm also adopts two archives (similar to subpopulations *CA* and *DA* in the skin membrane). It is necessary to stress that, although the SMG-MOMA used a skin membrane-guiding strategy, it still can not achieve a competitive performance on the MaOP test instances. This is mainly attributed to the skin membrane-guiding strategy developed in SMG-MOMA, which is only suited for solving MOPs with two or three objectives. The performance of skin membrane-guiding strategy in SMG-MOMA will considerably deteriorate to solve MaOPs. The above empirical results confirm the effectiveness of the skin membrane-guiding strategy suggested in SMG-MaOMA. Therefore, we can conclude that the proposed SMG-MaOMA is a promising approach for solving MaOPs.

To take a closer look at the results of algorithms on DTLZ test problems, Figs. 1 and 2 depict the objective values of the solution set with the best IGD value over 20 runs obtained by different algorithms on DTLZ2 and DTLZ4 with 10 objectives. From the figures, it can clearly be seen that the Two\_Arch2 method and SMG-MOMA obtain some solutions which do not converge to the Pareto front for 10-objective DTLZ2, while the MOEA/D, HypE and GrEA achieve a solution set which has a worse distribution than that of SMG-MaOMA. For 10-objective DTLZ4, a similar result can be observed from Fig. 2. The Two\_Arch2 method, SMG-MOMA and MOEA/D cannot

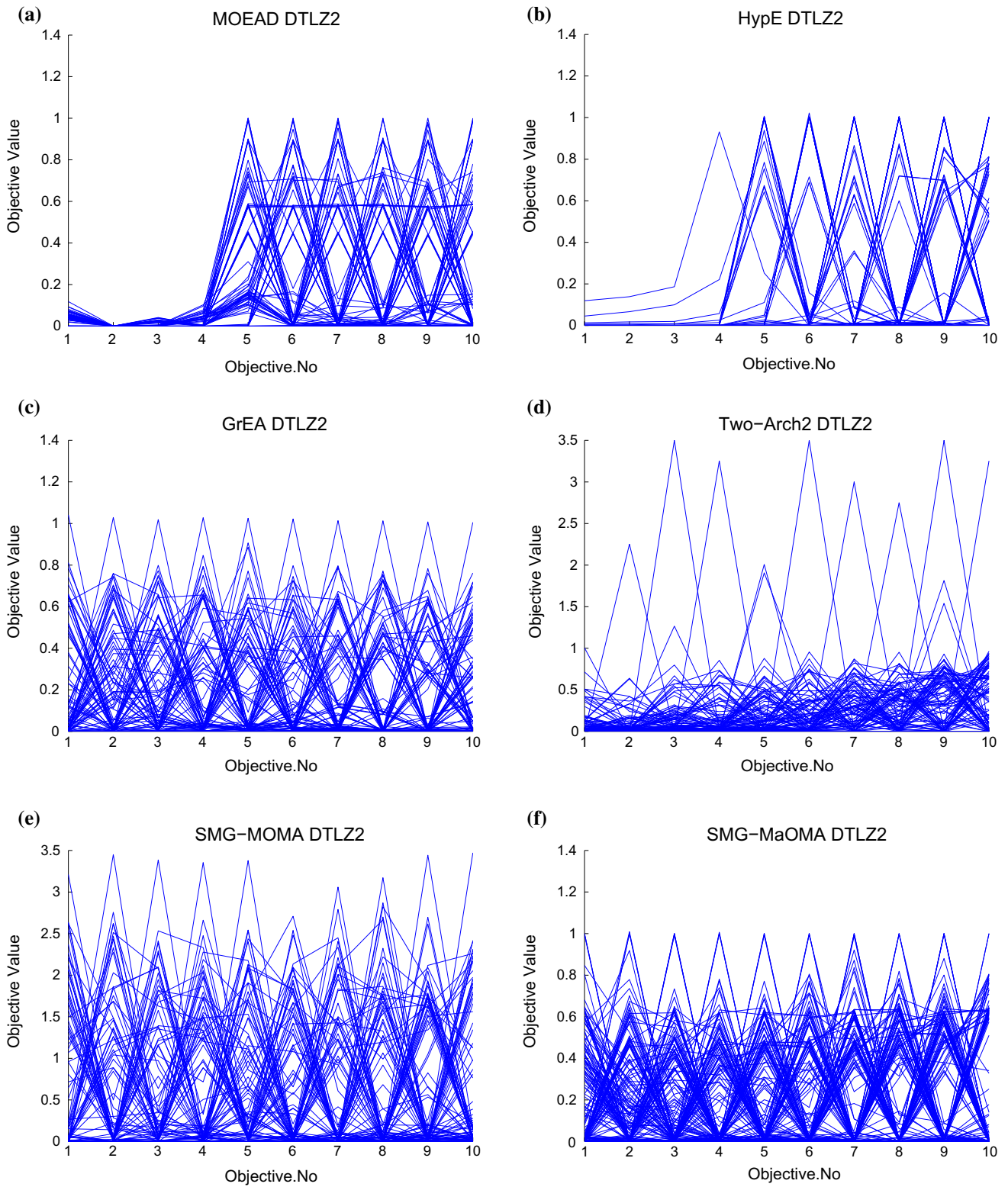
guarantee that the found solutions are all located on the Pareto front, but the distribution of solution set obtained by HypE and GrEA needs to be further improved. Compared to the five considered algorithms, the proposed SMG-MaOMA achieves a better balance between convergence and distribution of the obtained solution set.

In the following, we further verify the performance of the proposed SMG-MaOMA on the more challenging WFG test problems for MaOPs. Table 3 presents the Hypervolume (HV) values of the six considered algorithms on WFG1 to WFG9 with 4, 6, 8, 10 objectives averaging over 20 independent runs, where the best mean on each test instance is highlighted in boldface. The rank of the algorithms on each instance and the mean rank of each algorithm averaging over all considered WFG test instances are also shown in the table. From the table, the following two observations can be obtained. On the one hand, the performance of GrEA seems to be enhanced for solving WFG test problems. The GrEA achieves an average rank of 2.69 on all WFG test instances, which is much better than the ranks of the MOEA/D, HypE, Two\_Arch2 method and SMG-MOMA. These results demonstrate that GrEA is well suited to solve the more challenging WFG test problems. On the other hand, the proposed SMG-MaOMA is also more suited to solve WFG test problems whose superiority is enhanced on these test problems. On all 36 test instances of WFG test problems, the proposed SMG-MaOMA obtains the best HV value on 23 test instances.

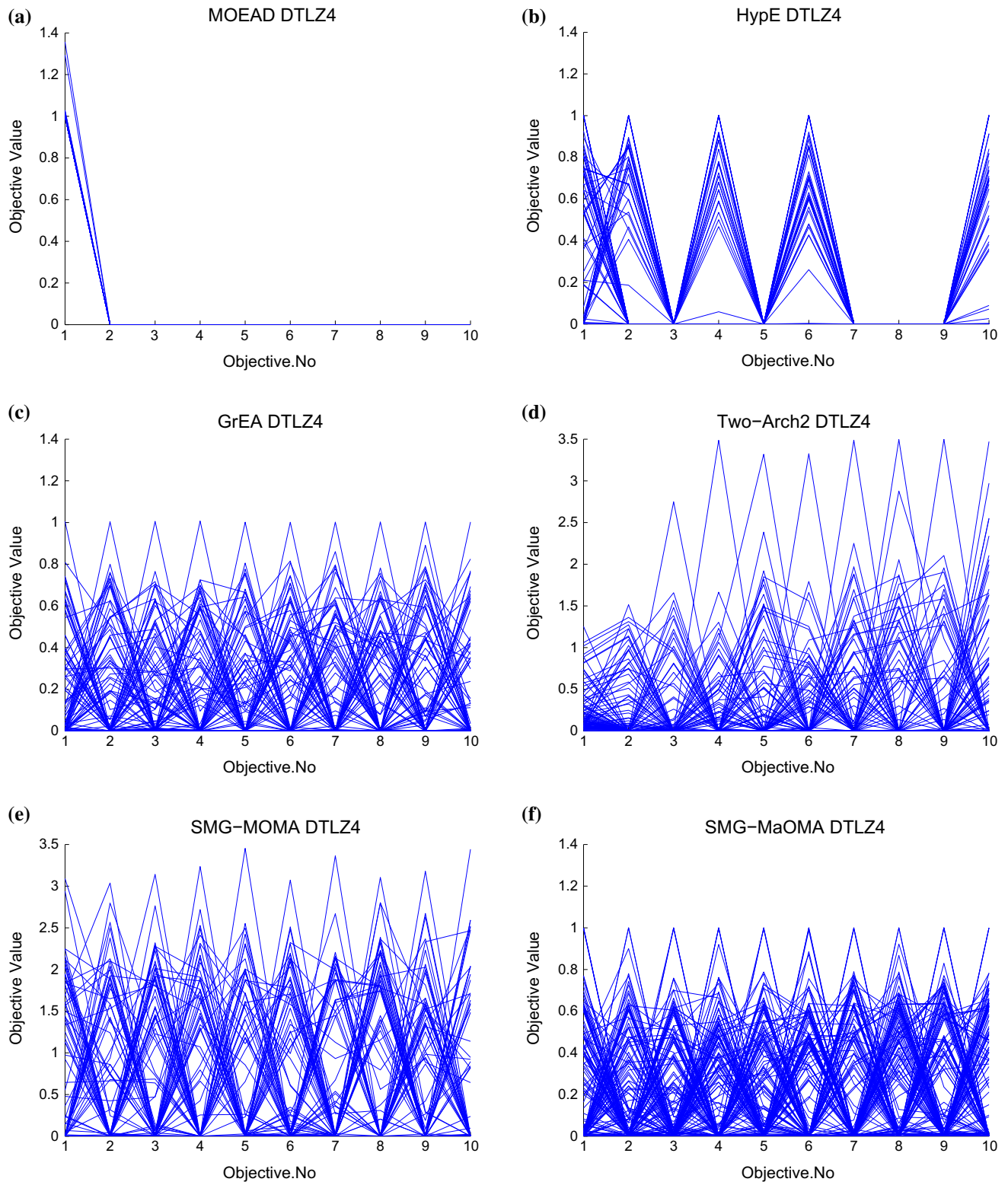
In summary, the proposed SMG-MaOMA is a competitive many-objective optimization algorithm, no matter that it is compared to existing popular many-objective optimization algorithms or membrane computing model-based multi-objective optimization algorithms.

## 4 Conclusions

In this paper, based on membrane computing models, we proposed a many-objective algorithm named SMG-MaOMA. In SMG-MaOMA, a skin membrane-guiding



**Fig. 1** Objective values of the solution set with the best IGD value over 20 runs obtained by the six compared algorithms on 10-objective DTLZ2. **a** MOEA/D, **b** HypE, **c** GrEA, **d** Two-Arch2, **e** SMG-MOMA, **f** SMG-MaOMA



**Fig. 2** Objective values of the solution set with the best IGD value over 20 runs obtained by the six compared algorithms on 10-objective DTLZ4. **a** MOEA/D, **b** HypE, **c** GrEA, **d** Two-Arch2, **e** SMG-MOMA, **f** SMG-MaOMA



**Table 3** Hypervolume (HV) values of the six compared algorithms on WFG1 to WFG9, where the best mean on each instance is highlighted in boldface

Problems	Obj.	MOEA/D	HypE	GrEA	Two_Arch2	SMG-MOMA	SMG-MaOMA
WFG1	4	3.56E+2[2] (2.20E+1)	2.96E+2[4] (2.47E+1)	2.82E+2[5] (1.81E+1)	1.20E+2[6] (1.25E+0)	3.22E+2[3] (2.36E+1)	<b>3.65E+2[1]</b> ( <b>1.14E+0</b> )
	6	3.89E+4[4] (6.75E+3)	<b>4.51E+4[1]</b> ( <b>5.38E+1</b> )	4.48E+4[3] (2.31E+2)	1.29E+4[6] (1.15E+3)	2.73E+4[5] (2.70E+3)	4.50E+4[2] (1.41E+2)
	8	6.85E+6[2] (1.23E+6)	<b>6.91E+6[1]</b> ( <b>5.10E+5</b> )	6.32E+6[4] (1.24E+6)	2.48E+6[6] (2.61E+4)	6.04E+6[5] (8.01E+5)	6.36E+6[2] (1.35E+5)
	10	1.69E+9[5] (4.30E+8)	2.62E+9[4] (4.14E+8)	3.29E+9[2] (8.72E+7)	8.66E+8[6] (6.04E+7)	3.02E+9[3] (2.00E+8)	<b>3.61E+9[1]</b> ( <b>3.00E+7</b> )
WFG2	4	2.58E+2[5] (4.02E+1)	2.32E+2[6] (2.99E+0)	2.72E+2[4] (4.18E+1)	3.51E+2[2] (2.93E+1)	<b>3.72E+2[1]</b> ( <b>9.52E-1</b> )	3.59E+2[3] (3.36E+2)
	6	4.14E+4[5] (3.78E+3)	4.22E+4[3] (4.93E+3)	4.37E+4[2] (2.58E+3)	<b>4.40E+4[1]</b> ( <b>2.78E+2</b> )	4.01E+4[6] (2.22E+2)	4.29E+4[4] (3.89E+3)
	8	1.03E+7[2] (9.28E+4)	1.02E+7[3] (2.39E+3)	9.95E+6[6] (4.39E+4)	1.00E+7[5] (6.33E+4)	1.01E+7[4] (3.84E+4)	<b>1.13E+7[1]</b> ( <b>3.77E+4</b> )
	10	3.31E+9[6] (4.93E+8)	<b>3.70E+9[1]</b> ( <b>6.06E+5</b> )	3.56E+9[5] (1.89E+7)	3.64E+9[4] (1.19E+7)	3.67E+9[3] (5.94E+6)	3.68E+9[2] (1.82E+7)
WFG3	4	1.89E+2[3] (5.26E-1)	1.98E+2[2] (3.19E+0)	<b>2.15E+2[1]</b> ( <b>2.19E+0</b> )	1.76E+2[4] (7.16E+2)	1.32E+2[6] (3.72E-1)	1.60E+2[5] (3.96E+0)
	6	1.71E+4[6] (1.93E-1)	4.30E+4[2] (3.23E+3)	3.19E+4[4] (2.56E+3)	<b>5.26E+4[1]</b> ( <b>5.02E+2</b> )	2.45E+4[5] (2.56E+2)	3.35E+4[3] (2.60E+2)
	8	3.57E+6[3] (1.73E+5)	4.82E+6[2] (2.96E+4)	<b>5.93E+6[1]</b> ( <b>4.12E+4</b> )	2.82E+6[6] (3.36E+4)	3.25E+6[5] (5.97E+5)	4.27E+6[4] (2.91E+5)
	10	1.25E+9[6] (1.34E+8)	1.43E+9[2] (3.99E+7)	<b>1.53E+9[1]</b> ( <b>4.33E+8</b> )	1.36E+9[4] (6.30E+8)	1.27E+9[5] (5.89E+8)	1.38E+9[3] (2.02E+8)
WFG4	4	1.87E+2[4] (6.28E+0)	1.97E+2[2] (3.45E+0)	1.91E+2[3] (9.92E-1)	1.84E+2[5] (2.92E+0)	1.65E+2[6] (2.04E+0)	<b>2.03E+2[1]</b> ( <b>2.53E+0</b> )
	6	1.50E+4[6] (8.52E+3)	2.41E+4[3] (1.94E+3)	2.86E+4[2] (4.30E+2)	2.25E+4[4] (6.64E+2)	1.90E+4[5] (1.02E+3)	<b>3.05E+4[1]</b> ( <b>7.38E+2</b> )
	8	4.91E+6[5] (8.29E+5)	6.43E+6[2] (4.99E+5)	5.72E+6[3] (9.43E+4)	5.25E+6[4] (8.86E+4)	4.70E+6[6] (1.68E+5)	<b>7.16E+6[1]</b> ( <b>1.42E+5</b> )
	10	9.05E+8[6] (4.07E+8)	2.02E+9[4] (3.65E+8)	2.20E+9[2] (3.22E+7)	2.07E+9[3] (1.53E+8)	1.63E+9[5] (9.01E+7)	<b>2.61E+9[1]</b> ( <b>4.23E+7</b> )
WFG5	4	1.81E+2[3] (3.28E+0)	1.12E+2[6] (9.25E+0)	1.85E+2[2] (7.23E-1)	1.71E+2[4] (1.80E+0)	1.66E+2[5] (2.09E+0)	<b>1.88E+2[1]</b> ( <b>8.34E-1</b> )
	6	1.67E+4[6] (1.35E+4)	<b>3.03E+4[1]</b> ( <b>3.72E+2</b> )	2.80E+4[3] (2.14E+2)	2.09E+4[4] (2.30E+3)	2.00E+4[5] (3.22E+2)	2.94E+4[2] (2.09E+2)
	8	5.00E+6[4] (2.08E+6)	4.58E+6[6] (2.87E+5)	6.23E+6[2] (9.87E+4)	5.38E+6[3] (8.44E+4)	4.90E+6[5] (2.29E+5)	<b>7.03E+6[1]</b> ( <b>9.34E+4</b> )
	10	8.94E+8[6] (5.95E+8)	1.61E+9[5] (3.48E+7)	2.45E+9[2] (2.85E+7)	2.05E+9[3] (2.27E+7)	1.67E+9[4] (1.01E+8)	<b>2.51E+9[1]</b> ( <b>2.97E+7</b> )
WFG6	4	1.66E+2[2] (7.36E+0)	4.96E+1[6] (1.48E+0)	1.47E+2[5] (1.04E+0)	1.40E+2[4] (1.80E+0)	1.60E+2[3] (9.28E+0)	<b>1.88E+2[1]</b> ( <b>5.35E+0</b> )
	6	2.27E+4[6] (9.70E+3)	<b>3.08E+4[1]</b> ( <b>5.34E+2</b> )	2.75E+4[3] (9.51E+2)	1.83E+4[4] (4.33E+2)	1.99E+4[5] (7.23E+2)	2.85E+4[2] (8.72E+2)
	8	2.70E+6[4] (1.78E+6)	1.36E+6[6] (3.96E+4)	4.89E+6[2] (6.53E+4)	4.39E+6[3] (1.08E+5)	4.59E+6[5] (4.14E+5)	<b>6.80E+6[1]</b> ( <b>1.79E+5</b> )
	10	4.08E+8[6] (3.30E+8)	4.93E+8[5] (6.50E+6)	1.87E+9[2] (2.70E+7)	1.70E+9[3] (6.88E+7)	1.38E+9[4] (1.53E+8)	<b>2.54E+9[1]</b> ( <b>7.25E+7</b> )

**Table 3** continued

Problems	Obj.	MOEA/D	HypE	GrEA	Two_Arch2	SMG-MOMA	SMG-MaOMA
WFG7	4	1.95E+2[3]	1.92E+2[4]	2.04E+2[2]	1.75E+2[6]	1.84E+2[5]	<b>2.14E+2[1]</b>
		(3.68E+0)	(1.74E+0)	(1.00E+0)	(1.56E+0)	(3.80E+0)	<b>(5.31E−1)</b>
	6	1.02E+4[6]	3.05E+4[3]	3.17E+4[2]	1.96E+4[5]	2.01E+4[4]	<b>3.26E+4[1]</b>
		(6.70E+3)	(5.94E+2)	(2.41E+2)	(1.16E+3)	(5.57E+2)	<b>(7.29E+2)</b>
8	2.94E+6[6]	7.35E+6[2]	6.21E+6[3]	4.95E+6[4]	4.31E+6[5]	<b>7.48E+6[1]</b>	
	(1.62E+6)	(6.07E+5)	(1.46E+5)	(1.36E+5)	(4.51E+5)	<b>(8.40E+4)</b>	
10	4.09E+8[6]	2.36E+9[3]	2.45E+9[2]	1.92E+9[4]	1.43E+9[5]	<b>2.86E+9[1]</b>	
	(3.50E+8)	(1.78E+8)	(2.45E+7)	(2.98E+7)	(1.15E+8)	<b>(3.56E+7)</b>	
WFG8	4	1.30E+2[2]	4.83E+1[6]	1.08E+2[4]	1.29E+2[3]	1.08E+2[5]	<b>1.47E+2[1]</b>
		(2.17E+0)	(4.33E−1)	(3.78E+1)	(6.40E+0)	(5.27E+0)	<b>(1.40E+0)</b>
	6	3.37E+3[6]	<b>2.49E+4[1]</b>	2.29E+4[2]	1.49E+4[4]	1.42E+4[5]	2.24E+4[3]
		(6.58E+3)	<b>(4.80E+2)</b>	(1.54E+3)	(1.05E+3)	(1.23E+3)	(3.39E+2)
8	2.67E+6[5]	1.84E+6[6]	4.18E+6[2]	3.64E+6[4]	3.98E+6[3]	<b>4.79E+6[1]</b>	
	(1.10E+4)	(3.02E+4)	(9.30E+4)	(1.03E+5)	(1.25E+5)	<b>(1.72E+5)</b>	
10	6.92E+8[5]	6.76E+8[6]	1.74E+9[2]	1.35E+9[4]	1.38E+9[3]	<b>1.90E+9[1]</b>	
	(7.79E+7)	(2.62E+6)	(3.76E+7)	(2.74E+7)	(1.00E+8)	<b>(1.71E+8)</b>	
WFG9	4	1.60E+2[4]	1.39E+2[6]	1.81E+2[2]	1.66E+2[3]	1.53E+2[5]	<b>1.89E+2[1]</b>
		(1.60E+1)	(1.74E+1)	(3.51E+0)	(2.26E+1)	(1.95E+1)	<b>(4.14E+0)</b>
	6	1.91E+4[5]	2.25E+4[3]	2.79E+4[2]	1.97E+4[4]	1.76E+4[6]	<b>2.60E+4[1]</b>
		(8.17E+3)	(4.35E+2)	(3.84E+2)	(1.29E+3)	(4.81E+2)	<b>(2.20E+3)</b>
8	2.36E+6[6]	2.59E+6[5]	6.00E+6[2]	4.80E+6[3]	4.56E+6[4]	<b>6.28E+6[1]</b>	
	(1.91E+6)	(3.44E+5)	(9.30E+4)	(1.12E+5)	(3.75E+5)	<b>(4.60E+5)</b>	
10	3.42E+8[6]	9.20E+8[5]	<b>2.26E+9[1]</b>	1.98E+9[3]	1.52E+9[4]	2.29E+9[2]	
	(2.99E+8)	(4.48E+7)	<b>(5.78E+7)</b>	(2.10E+7)	(1.04E+8)	(1.02E+8)	
Average rank		4.64	3.56	2.64	3.58	4.53	1.67

strategy specially tailored for solving MaOPs is developed, where the population in the skin membrane is divided into two subpopulations. One subpopulation focuses on improving the convergence of populations in the internal membrane, while the other subpopulation aims to maintain the diversity of populations in the internal membrane. Experimental results show that the proposed SMG-MaOMA is very competitive compared with P system-based multi-objective algorithms and popular many-objective optimization algorithms.

The SMG-MaOMA has shown that the skin membrane-guiding strategy is an effective idea for designing membrane computing model-based many-objective algorithms; however, its performance can still be further improved. Hence, an interesting topic along this research line is to investigate the effect of convergence and diversity of populations for MaOPs and design a more effective the skin membrane-guiding strategy. It also deserves to use the proposed SMG-MaOMA to solve the many-objective optimization problems in real-world, since there are a large number of challenging many-objective optimization problems widely present in engineering

applications and scientific research that still need to be investigated.

For further research, it would be of interests to consider the most heavily investigated models, such as neural networks [3, 25, 40, 41] and novel approaches, such as matching strategies [29, 36, 45, 48], learning algorithms [13, 44, 49, 64] for multi/many-objective optimization.

**Acknowledgements** This work was supported by National Natural Science Foundation of China (61402187, 61502001, 61502004, 61502535, 61502532 and 61502012), Beijing Natural Science Foundation (4164096) and the Fundamental Research Funds for the Central Universities (2652015340), China Postdoctoral Science Foundation funded Project (2016M592267).

## References

- Bader J, Zitzler E (2011) Hype: an algorithm for fast hypervolume-based many-objective optimization. *Evolut Comput* 19(1):45–76
- Bin G, Sheng VS (2016) A robust regularization path algorithm for-support vector classification. *IEEE Trans Neural Netw Learn Syst*. doi:10.1109/TNNLS.2016.2527796

3. Bin G, Sheng VS, Wang Z, Ho D, Osman S, Li S (2015) Incremental learning for-support vector regression. *Neural Netw* 67:140–150
4. Bin G, Sheng VS, Tay KY, Romano W, Li S (2015) Incremental support vector learning for ordinal regression. *IEEE Trans Neural Netw Learn Syst* 26(7):1403–1416
5. Bin G, Sun X, Sheng VS (2016) Machine structural minimax probability. *IEEE Trans Neural Netw Learn Syst*. doi:[10.1109/TNNLS.2016.2544779](https://doi.org/10.1109/TNNLS.2016.2544779)
6. Cheng J, Zhang G, Wang T (2015) A membrane-inspired evolutionary algorithm based on population P systems and differential evolution for multi-objective optimization. *J Comput Theor Nanosci* 12(7):1150–1160
7. Cheng R, Jin Y, Olhofer M, Sendhoff B (2016) A reference vector guided evolutionary algorithm for many-objective optimization. *IEEE Trans Evolut Comput* 20(5):773–791
8. DíAz-Pernil D, Berciano A, PeñA-Cantillana F, Gutiérrez-Naranjo MA (2013) Segmenting images with gradient-based edge detection using membrane computing. *Pattern Recogn Lett* 34(8):846–855
9. Fleming PJ, Purshouse RC, Lygoe RJ (2005) Many-objective optimization: an engineering design perspective. In: Proceedings of the third international conference on evolutionary multi-criterion optimization, pp 14–32
10. Freund R, Păun G, Pérez-Jiménez MJ (2005) Tissue P systems with channel states. *Theor Comput Sci* 330(1):101–116
11. Fu Z, Sun X, Liu Q, Zhou L, Shu J (2015) Achieving efficient cloud search services: multi-keyword ranked search over encrypted cloud data supporting parallel computing. *IEICE Trans Commun E98–B(1)*:190–200
12. Fu Z, Wu X, Guan C, Sun X, Ren K (2016) Towards efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. *IEEE Trans Inform Forensics Secur*. doi:[10.1109/TIFS.2016.2596138](https://doi.org/10.1109/TIFS.2016.2596138)
13. Guo P, Wang J, Li B, Lee S (2014) A variable threshold-value authentication architecture for wireless mesh networks. *J Internet Technol* 15(6):929–936
14. Huang L, He X, Wang N, Xie Y (2007) P systems based multi-objective optimization algorithm. *Prog Nat Sci* 17(4):458–465
15. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans Evolut Comput* 10(5):477–506
16. Ionescu M, Paun G, Yokomori T (2006) Spiking neural P systems. *Fundam Inform* 71(2):279–308
17. Ishdorj T-O, Leporati A, Pan L, Zeng X, Zhang X (2010) Deterministic solutions to QSAT and Q3SAT by spiking neural P systems with pre-computed resources. *Theor Comput Sci* 411(25):2345–2358
18. Jaya Prakash V (2003) On the power of tissue P systems working in the maximal-one mode. In: Preproceedings of the workshop on membrane computing, vol 8, pp 356–364
19. Jin X (2016) Probe machine. *IEEE Trans Neural Netw Learn Syst* 27(7):1405–1416
20. Krüsselbrink JW, Emmerich MTM, Bäck T, Bender A, IJzerman AP, van der Horst E (2009) Combining aggregation with Pareto optimization: a case study in evolutionary molecular design. In: Proceedings of the fifth international conference on evolutionary multi-criterion optimization, pp 453–467
21. Leporati A, Mauri G, Zandron C, Păun G, Pérez-Jiménez MJ (2009) Uniform solutions to SAT and subset sum by spiking neural P systems. *Nat Comput* 8(4):681–702
22. Li B, Li J, Tang K, Yao X (2014) An improved two archive algorithm for many-objective optimization. In: Proceedings of the 2014 IEEE congress on evolutionary computation, pp 2869–2876
23. Li J, Li X, Yang B, Sun X (2015) Segmentation-based image copy-move forgery detection scheme. *IEEE Trans Inf Forensics Secur* 10(3):507–518
24. Liu C, Han M, Wang X (2011) A multi-objective evolutionary algorithm based on membrane systems. In: Proceedings of the fourth international workshop on advanced computational intelligence, pp 103–109
25. Ma T, Zhou J, Tang M, Tian Y, Al-Dhelaan A, Al-Rodhaan M, Lee S (2015) Social network and tag sources based augmenting collaborative recommender system. *IEICE Trans Inform Syst* E98–D(4):902–910
26. Martín-Vide C, Păun G, Pazos J, Rodríguez-Patón A (2003) Tissue P systems. *Theor Comput Sci* 296(2):295–326
27. Nishida TY (2004) An application of P system: a new algorithm for NP-complete optimization problems. In: Proceedings of the 8th world multi-conference on systems, cybernetics and informatics, pp 109–112
28. Pan Z, Zhang Y, Kwong S (2015) Efficient motion and disparity estimation optimization for low complexity multiview video coding. *IEEE Trans Broadcast* 61(2):166–176
29. Pan Z, Lei J, Zhang Y, Sun X, Kwong S (2016) Fast motion estimation based on content property for low-complexity H.265/HEVC encoder. *IEEE Trans Broadcast*. doi:[10.1109/TBC.2016.2580920](https://doi.org/10.1109/TBC.2016.2580920)
30. Păun G (2000) Computing with membranes. *J Comput Syst Sci* 61(1):108–143
31. Păun G (2012) Membrane computing: an introduction. Springer, Berlin
32. Păun G, Păun RA (2005) Membrane computing as a framework for modeling economic processes. In: Proceedings of the seventh international symposium on symbolic and numeric algorithms for scientific computing, pp 8–31
33. Păun G, Suzuki Y, Tanaka H, Yokomori T (2004) On the power of membrane division in P systems. *Theor Comput Sci* 324(1):61–85
34. Păun G, Rozenberg G, Salomaa A (2010) The Oxford handbook of membrane computing. Oxford University Press Inc, Oxford
35. Ren Y, Shen J, Wang J, Han J, Lee S (2015) Mutual verifiable provable data auditing in public cloud storage. *J Internet Technol* 16(2):317–323
36. Shen J, Tan H, Wang J, Wang J, Lee S (2015) A novel routing protocol providing good transmission reliability in underwater sensor networks. *J Internet Technol* 16(1):171–178
37. Song T, Pan L (2015) On the universality and non-universality of spiking neural P systems with rules on synapses. *IEEE Trans Nanobiosci* 14(8):960–966
38. Song T, Pan L (2015) Spiking neural P systems with rules on synapses working in maximum spikes consumption strategy. *IEEE Trans Nanobiosci* 14(1):38–44
39. Song T, Pan L (2016) Spiking neural P systems with request rules. *Neurocomputing* 193(12):193–200
40. Song T, Pan L (2016) Spiking neural P systems with request rules. *Neurocomputing* 193(12):193C200
41. Song T, Liu X, Zhao Y, Zhang X (2016) Spiking neural P systems with white hole neurons. *IEEE Trans Nanobiosci*. doi:[10.1109/TNB.2016.2598879](https://doi.org/10.1109/TNB.2016.2598879)
42. Wang X, Song T, Gong F, Zheng P (2016) On the computational power of spiking neural P systems with self-organization. *Sci Rep*. doi:[10.1038/srep27624](https://doi.org/10.1038/srep27624)
43. Wang H, Jin Y, Yao X (2016) Diversity assessment in many-objective optimization. *IEEE Trans Cybern*. doi:[10.1109/TCYB.2016.2550502](https://doi.org/10.1109/TCYB.2016.2550502)
44. Wen X, Ling Shao Y, Xue Y, Fang W (2015) A rapid learning algorithm for vehicle classification. *Inf Sci* 295(1):395–406

45. Xia Z, Wang X, Sun X, Wang B (2014) Steganalysis of least significant bit matching using multi-order differences. *Secur Commun Netw* 7(8):1283–1291
46. Xia Z, Wang X, Sun X, Wang Q (2015) A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data. *IEEE Trans Parallel Distrib Syst* 27(2):340–352
47. Xia Z, Wang X, Zhang L, Qin Z, Sun X, Ren K (2016) A Privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. *IEEE Trans Inform Forensics Secur.* doi:[10.1109/TIFS.2016.2590944](https://doi.org/10.1109/TIFS.2016.2590944)
48. Xia Z, Wang X, Sun X, Liu Q, Xiong N (2016) Steganalysis of LSB matching using differences between nonadjacent pixels. *Multimed Tools Appl* 75(4):1947–1962
49. Xie S, Wang Y (2014) Construction of tree network with limited delivery latency in homogeneous wireless sensor networks. *Wirel Pers Commun* 78(1):231–246
50. Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 17(5):721–736
51. Yuan C, Sun X, Lv R (2016) Fingerprint liveness detection based on multi-scale LPQ and PCA. *China Commun* 13(7):60–65
52. Zeng X, Lei X, Liu X, Pan L (2014) On languages generated by spiking neural p systems with weights. *Inf Sci* 278:423–433
53. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evolut Comput* 11(6):712–731
54. Zhang G, Gheorghe M, Chao-Zhong W (2008) A quantum-inspired evolutionary algorithm based on P systems for knapsack problem. *Fundam Inform* 87(1):93
55. Zhang X, Tian Y, Jin Y (2014) A knee point driven evolutionary algorithm for many-objective optimization. *IEEE Trans Evolut Comput* 19(6):761–776
56. Zhang X, Liu Y, Luo B, Pan L (2014) Computational power of tissue P systems for generating control languages. *Inf Sci* 278:285–297
57. Zhang X, Wang B, Pan L (2014) Spiking neural P systems with a generalized use of rules. *Neural Comput* 26:2925–2943
58. Zhang X, Zeng X, Luo B, Pan L (2014) On some classes of sequential spiking neural P systems. *Neural Comput* 26(5):974–997
59. Zhang X, Pan L, Paun A (2015) On the universality of axon P systems. *IEEE Trans Neural Netw Learn Syst* 26:2816–2829
60. Zhang X, Li J, Zhang L (2016) A multi-objective membrane algorithm guided by the skin membrane. *Nat Comput* 15(4):597–610
61. Zhang Y, Sun X, Baowei W (2016) Efficient algorithm for K-barrier coverage based on integer linear programming. *China Commun* 13(7):16–23
62. Zhang X, Tian Y, Jin Y (2016) Approximate non-dominated sorting for evolutionary many-objective optimization. *Inform Sci* 369:14–33
63. Zhangjie F, Ren K, Shu J, Sun X, Huang F (2015) Enabling personalized search over encrypted outsourced data with efficiency improvement. *IEEE Trans Parallel Distrib Syst.* doi:[10.1109/TPDS.2015.2506573](https://doi.org/10.1109/TPDS.2015.2506573)
64. Zheng Y, Jeon B, Danhua X, Wu QMJ, Zhang H (2015) Image segmentation by generalized hierarchical fuzzy C-means algorithm. *J Intell Fuzzy Syst* 28(2):961–973
65. Zhou Z, Wang Y, Jonathan Wu QM, Yang C-N, Sun X (2016) Effective and efficient global context verification for image copy detection. *IEEE Trans Inform Forensics Secur.* doi:[10.1109/TIFS.2016.2601065](https://doi.org/10.1109/TIFS.2016.2601065)
66. Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In *Proceedings of 2004 international conference on parallel problem solving from nature-PPSN VIII*, pp 832–842