Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 906147, 9 pages
http://dx.doi.org/10.1155/2014/906147

Research Article

A New Multiobjective Evolutionary Algorithm Based on Decomposition of the Objective Space for Multiobjective Optimization

Cai Dai and Yuping Wang

School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Correspondence should be addressed to Cai Dai; daicai8403@hotmail.com

Received 16 September 2013; Accepted 22 December 2013; Published 12 January 2014

Academic Editor: Mehmet Sezer

In order to well maintain the diversity of obtained solutions, a new multiobjective evolutionary algorithm based on decomposition of the objective space for multiobjective optimization problems (MOPs) is designed. In order to achieve the goal, the objective space of a MOP is decomposed into a set of subobjective spaces by a set of direction vectors. In the evolutionary process, each subobjective space has a solution, even if it is not a Pareto optimal solution. In such a way, the diversity of obtained solutions can be maintained, which is critical for solving some MOPs. In addition, if a solution is dominated by other solutions, the solution can generate more new solutions than those solutions, which makes the solution of each subobjective space converge to the optimal solutions as far as possible. Experimental studies have been conducted to compare this proposed algorithm with classic MOEA/D and NSGAII. Simulation results on six multiobjective benchmark functions show that the proposed algorithm is able to obtain better diversity and more evenly distributed Pareto front than the other two algorithms.

1. Introduction

Since there are many problems with several optimization problems or criteria in real world [1], multiobjective optimization has become a hot research topic. Unlike single-objective optimization problem, multiobjective optimization problem has a series of noninferior alternative solutions, also known as Pareto optimal solutions (the set of Pareto optimal solutions is called Pareto front [2]), which represent the possible trade-off among various conflicting objectives. Therefore, multiobjective optimization algorithms for MOP should be able to (1) discover solutions as close to the optimal solutions as possible; (2) find solutions as uniform as possible in the obtained nondominated front; (3) determine solutions to cover the true Pareto front (PF) as broad as possible. However, achieving these three goals simultaneously is still a challenge for multiobjective optimization algorithms.

Among various multiobjective optimization algorithms, multiobjective evolutionary algorithms (MOEA), which make use of the strategy of the population evolutionary to optimize the problems, are an effective method for solving MOPs. In recent years, many MOEAs have been proposed for solving the multiobjective optimization problems [3–18]. In the MOEA literatures, Goldberg's population categorization strategy [19] based on nondominance is important. Many algorithms use the strategy to assign a fitness value based on the nondominance rank of members. For example, the nondominated sorting genetic algorithm II [3] was proposed by Deb et al. in 2002, which uses the crowding measure and the elitism strategy; Zitzler et al. [20] proposed the strength Pareto evolutionary algorithm II (SPEA2) which was also based on the elitism strategy; Soylu and Köksalan proposed a favorable weight-based evolutionary algorithm (FWEA) [21]. These algorithms mainly rely on Pareto dominance to guide their search, particularly their selection operators. In contrast, MOEA/D (multiobjective evolutionary algorithm based on decomposition) [22] makes use of traditional aggregation methods to transform the task of approximating the Pareto front (PF) into a number of single-objective optimization subproblems. MOEA/D works well on a wide range of multiobjective problems with many objectives, discrete decision variables, and complicated Pareto sets [23, 24]. MOEA/D has also been used as a basic element in some hybrid algorithms; for example, MOEA/D with differential evolution and particle swarm optimization is proposed by Mashwani [25], Li and Landa-Silva [26] combine MOEA/D and simulated annealing to solve MOPs, and other hybrid algorithms also use MOEA/D to solve MOPs (e.g., [27–29]). Moreover, MOEA/D is used to solve various kinds of problems (e.g., [30, 31]). In MOEA/D, weight vectors and aggregate functions play a very important role. In the literature [32], the uniform design is used to generate the weight vectors; Qi et al. [33] design an adaptive weight vector adjustment to adaptively generate uniform distribution weight vectors. Ishibuchi et al. [34] propose an idea of automatically choosing between the weighted sum and the weighted Tchebycheff for each solution in each generation.

Any effective MOEAs must well maintain population diversity since their goal is to approximate a set instead of a single point. Pareto-dominance-based algorithms achieve the goal by maintaining the diversity of obtained Pareto optimal solutions. If these algorithms obtaining Pareto optimal solutions cannot cover the entire PF, this goal cannot be achieved. The current MOEA/D algorithms achieve their population diversity via the diversity of their subproblems. An elite diversity is used for their selection in these algorithms. The aggregation function values of a new solution and an old solution completely determined whether the new solution replaces the old solution or not. In some cases, such replacement can cause a severe loss of population diversity, which is because the aggregation function value of a solution only reflects the extent of the solution close to the ideal point which is determined by the aggregation function and the weight vector, but it does not reflect the space where the objective vector of the solution locates. Thus, the objective space where the objective vector of the solution locates should be considered to maintain the diversity of obtained Pareto optimal solutions. In this paper, the objective space of a MOP is decomposed into a set of subobjective spaces by a set of direction vectors. Each subobjective space has a solution: if a new solution will replace the solution, the new solution must dominate the solution and its objective vector locates in the subobjective space. In such a way, the diversity of obtained solutions can be maintained. In addition, the crowding distance [3] is used to calculate the fitness value of a solution for the selection operators. In this cause, if a solution is dominated by other solutions, the solution is more likely to be selected than other solutions, so it can generate more new solutions to quickly find the optimal solution of the subobjective space, which makes the solution of each subobjective space converge to the optimal solutions as far as possible. Based on these approaches, a new multiobjective evolutionary algorithm, EASS, is designed. We show that EASS can significantly outperform MOEA/D and NSGAII on a set of test instances used in our experimental studies.

This paper is organized as follows. Section 2 introduces the main concepts of the multiobjective optimization; Section 3 describes the evolutionary algorithm based on decomposition of the objective space, while Section 4 shows the experiment results of proposed algorithm; finally, Section 5 gives the conclusions and future works.

2. Multiobjective Optimization

A multiobjective optimization problem can be formulated as follows [35]:

$$\begin{aligned} \min \quad & y = F(x) = \left(f_1(x), f_2(x), \ldots, f_m(x)\right) \\ \text{s.t.} \quad & g_i(x) \le 0, \quad i = 1, 2, \ldots, q \\ & h_j(x) = 0, \quad j = 1, 2, \ldots, p, \end{aligned} \tag{1}$$

where $x = (x_1, \ldots, x_n) \in X \subset R^n$ is called decision variable and X is n-dimensional decision space. $f_i(x)(i = 1, \ldots, m)$ is the ith objective to be minimized, $g_j(x)(j = 1, 2 \ldots, q)$ defines jth inequality constraint, and $h_j(x)(j = 1, 2, \ldots, p)$ defines jth equality constraint. Furthermore, all the constraints determine the set of feasible solutions which is denoted by Ω. To be specific, we try to find a feasible solution $x \in \Omega$ minimizing each objective function $f_i(x)(i = 1, \ldots, m)$ in F. In the following, four important definitions [36] for multiobjective problems are given.

Definition 1 (Pareto dominance). Pareto dominance between solutions $x, z \in \Omega$ is defined as follows. If

$$\begin{aligned} & \forall i \in \{1, 2, \ldots, m\}\ f_i(x) \le f_i(z), \\ & \wedge \exists i \in \{1, 2, \ldots, m\}\ f_i(x) < f_i(z) \end{aligned} \tag{2}$$

are satisfied, x dominates (Pareto dominate) z (denoted as $x \succ z$).

Definition 2 (Pareto optimal). A solution vector x is said to be Pareto optimal with respect to Ω if $\nexists z \in \Omega : z \succ x$.

Definition 3 (Pareto optimal set (PS)). The set of Pareto optimal solutions (PS) is defined as

$$\text{PS} = \left\{x \in \Omega \mid \nexists z \in \Omega : z \succ x\right\}. \tag{3}$$

Definition 4 (Pareto front). The Pareto optimal front (PF) is defined as

$$\text{PF} = \{F(x) \mid x \in \text{PS}\}. \tag{4}$$

3. A New Multiobjective Evolutionary Algorithm Based on Decomposition of the Objective Space

This new algorithm consists of three parts: solutions classification, update strategy, and selection strategy which will be introduced one by one in this section.

3.1. Solutions Classification. The objective space of an MOP is decomposed into a set of subobjective spaces by a set of direction vectors, and then obtained solutions are classified by these direction vectors to make each subobjective space have a solution. For a given set of direction vectors $(\gamma^1, \gamma^2, \ldots, \gamma^N)$, and the set of current obtained solutions being POP, these solutions will be classified by the following formula:

$$P^i = \left\{ x \mid x \in \text{POP}, \Delta\left(F(x), \gamma^i\right) = \max_{1 \le j \le N} \left\{ \Delta\left(F(x), \gamma^j\right) \right\} \right\},$$
$$\Delta\left(F(x), \gamma^i\right) = \frac{\gamma^i * (F(x) - Z)^T}{\left\|\gamma^i\right\| * \left\|F(x) - Z\right\|}, \quad i = 1, \ldots, m, \tag{5}$$

where $Z = (Z_1, \ldots, Z_m)$ is a reference point and $Z_i = \min\{f_i(x) \mid x \in \Omega\}$, $\Delta(F(x), \gamma^i)\}$ is the cosine of the angle between γ^i and $F(x) - Z$. These solutions are divided into N classes by the formula (5) and the objective space Ω is divided into N subobjective spaces $\Omega_1, \ldots, \Omega_N$, where $\Omega_k (k = 1, \ldots, N)$ is

$$\Omega_k = \left\{ F(x) \mid x \in \Omega, \Delta\left(F(x), \gamma^k\right) = \max_{1 \le j \le N} \left\{ \Delta\left(F(x), \gamma^j\right) \right\} \right\}. \tag{6}$$

If a $P^i (1 \le i \le N)$ is empty, a solution is randomly selected from POP and put to P^i. We would like to make the following comments on the classification (decomposition) method.

(1) This method is equivalent in a sense that the PFs of all these subobjective spaces constitute the PF of (1).

(2) Even when the PS of (1) has a nonlinear geometric shape, the PS of each subobjective space could be close to linear because it is just a small part of the PS of (1). Therefore, formulas (5) and (6) make (1) simpler than before, at least in terms of PS shapes.

(3) This classification (decomposition) method does not require any aggregation methods. A user only needs to choose a set of direction vectors. To some extent, it requires little human labor.

3.2. Update Strategy. The elitist strategy is used to update solutions. When it meets one of the following conditions, a new solution will replace the current solution of a subobjective space:

(1) if the objective vector of the current solution does not locate in this subobjective space, the objective vector of the new solution locates in this space or the current solution is dominated by the new solution;

(2) if the objective vector of the current solution locates in this subobjective space, the objective vector of the new solution also locates in this space and the current solution is dominated by the new solution.

The first update condition makes each subobjective space have a solution whose objective vector locates in this subobjective space, which can well maintain the diversity of obtained solutions. The second update condition makes nondominated solution be kept, which can make solutions converge to the PF.

3.3. Selection Strategy. In this work, we hope that those solutions which are kept after the above update strategy and dominated by other solutions have a stronger ability to survive than other solutions. In such a way, these solutions are more likely to be selected to generate new solutions, and then their subobjective spaces can quickly find their optimal solutions. Therefore, the finally obtained solution of each subobjective space is as close as possible to the PF. In order to achieve the goal, the crowding distance [3] is used to calculate the fitness value of a solution for the selection operators. Because these solutions are dominated by other solutions and the objective vectors of those solutions do not locate in these subobjective spaces of these solutions, so in the term of the objective vector, these solutions have fewer solutions in their surround than other solutions. Thus, by using the crowding distance to calculate the fitness value of a solution, the fitness values of these solutions are better than those solutions and these solutions are more likely to be selected to generate new solutions.

3.4. Steps of the Proposed Algorithm. Based on the above methods, a new multiobjective evolutionary algorithm (EASS) is proposed and the steps of the algorithm EASS are as follows.

Step 1 (initialization). Given N direction vectors $(\gamma^1, \gamma^2, \ldots, \gamma^N)$, randomly generate an initial population POP(k), and its size is N; let $k = 0$. Set $Z_i = \min\{f_i(x) \mid x \in \text{POP}(k)\}$, $1 \le i \le m$.

Step 2 (fitness). Solutions of POP(k) are firstly divided into N classes by the formula (5) and the fitness value of each solution in POP(k) is calculated by the crowding distance. Then, some better solutions are selected from the population POP(k) and put into the population POP. In this work, binary tournament selection is used.

Step 3 (new solutions). Apply genetic operators to the parent population to generate offspring. The set of all these offspring is denoted as O.

Step 4 (update). Z is firstly updated. For each $j = 1, \ldots, m$, if $Z_j > \min\{f_j(x) \mid x \in O\}$, then set $Z_j = \min\{f_j(x) \mid x \in O\}$. The solutions of $\text{POP}(k) \cup O$ are firstly classed by the formula (5); then N best solutions are selected by the update strategy of Section 3.2 and put into POP($k + 1$). Let $k = k + 1$.

Step 5 (termination). If stop condition is satisfied, stop; otherwise, go to Step 2.

4. Experimental Study

In this section, EASS has been fully compared with a promising multiobjective evolutionary algorithm based on decomposition (MOEA/D) [22], which was ranked first in the unconstrained MOEA competition [37], and nondominated sorting genetic algorithm II (NSGAII) [3] on seven continuous test problems.

Table 1: Multiobjective benchmark functions.

Instance	Objective function
F1	$f_1(x) = (1+g(x))\,x_1$ $f_2(x) = (1+g(x))(1-\sqrt{x_1})$ $g(x) = 2\sin(\pi x_1)\left(n-1+\sum_{i=2}^{n}\left(t_i^2-\cos(2\pi t_i)\right)\right)$ $t_i = x_i - \sin(0.5\pi x_i)$
F2	$f_1(x) = (1+g(x))\,x_1$ $f_2(x) = (1+g(x))(1-x_1^2)$ $g(x) = 2\sin(\pi x_1)\left(n-1+\sum_{i=2}^{n}\left(t_i^2-\cos(2\pi t_i)\right)\right)$ $t_i = x_i - \sin(0.5\pi x_i)$
F3	$f_1(x) = (1+g(x))\cos\left(\frac{\pi x_1}{2}\right)$ $f_2(x) = (1+g(x))\sin\left(\frac{\pi x_1}{2}\right)$ $g(x) = 2\sin(\pi x_1)\left(n-1+\sum_{i=2}^{n}\left(t_i^2-\cos(2\pi t_i)\right)\right)$ $t_i = x_i - \sin(0.5\pi x_i)$
F4	$f_1(x) = (1+g(x))\,x_1$ $f_2(x) = (1+g(x))(1-x_1^{0.5}\cos^2(2\pi x_1))$ $g(x) = 2\sin(\pi x_1)\left(n-1+\sum_{i=2}^{n}\left(t_i^2-\cos(2\pi t_i)\right)\right)$ $t_i = x_i - \sin(0.5\pi x_i)$
F5	$f_1(x) = (1+g(x))\,x_1 x_2$ $f_2(x) = (1+g(x))\,x_1(1-x_2)$ $f_3(x) = (1+g(x))(1-x_1)$ $g(x) = 2\sin(\pi x_1)\left(n-1+\sum_{i=3}^{n}\left(t_i^2-\cos(2\pi t_i)\right)\right)$ $t_i = x_i - x_1 x_2$
F6	$f_1(x) = (1+g(x))\cos(0.5\pi x_1)\cos(0.5\pi x_2)$ $f_2(x) = (1+g(x))\cos(0.5\pi x_1)\sin(0.5\pi x_2)$ $f_3(x) = (1+g(x))\sin(0.5\pi x_1)$ $g(x) = 2\sin(\pi x_1)\left(n-1+\sum_{i=3}^{n}\left(t_i^2-\cos(2\pi t_i)\right)\right)$ $t_i = x_i - x_1 x_2$

4.1. Test Problem. The following modified ZDT [36] and DTLZ [38] instances are used in this paper. $g(x)$ functions are modified to increase the difficulty of these instances. These strategies for constructing these test problems are used in [39]. These functions with complicated Pareto set shapes can cause difficulties for MOEAs. Their search space is $[0,1]^n$ and $n = 10$. All these instances are for minimization and described in Table 1.

4.2. Parameter Settings. Algorithms are implemented on a personal computer (Intel Xeon CPU 2.53 GHz, 3.98 G RAM). The individuals are all coded as the real vectors. Polynomial mutation [40] operators are applied directly to real parameter values in three algorithms, that is, NSGAII, MOEA/D, and EASS. For crossover operators, simulated binary crossover (SBX [40]) is used in NSGAII and EASS; differential evolution (DE) [41] is used in MOEA/D. The parameter settings in this paper are as follows.

(1) Control parameters in reproduction operators:

   (a) distribution index is 20 and crossover probability is 1 in the SBX operator;

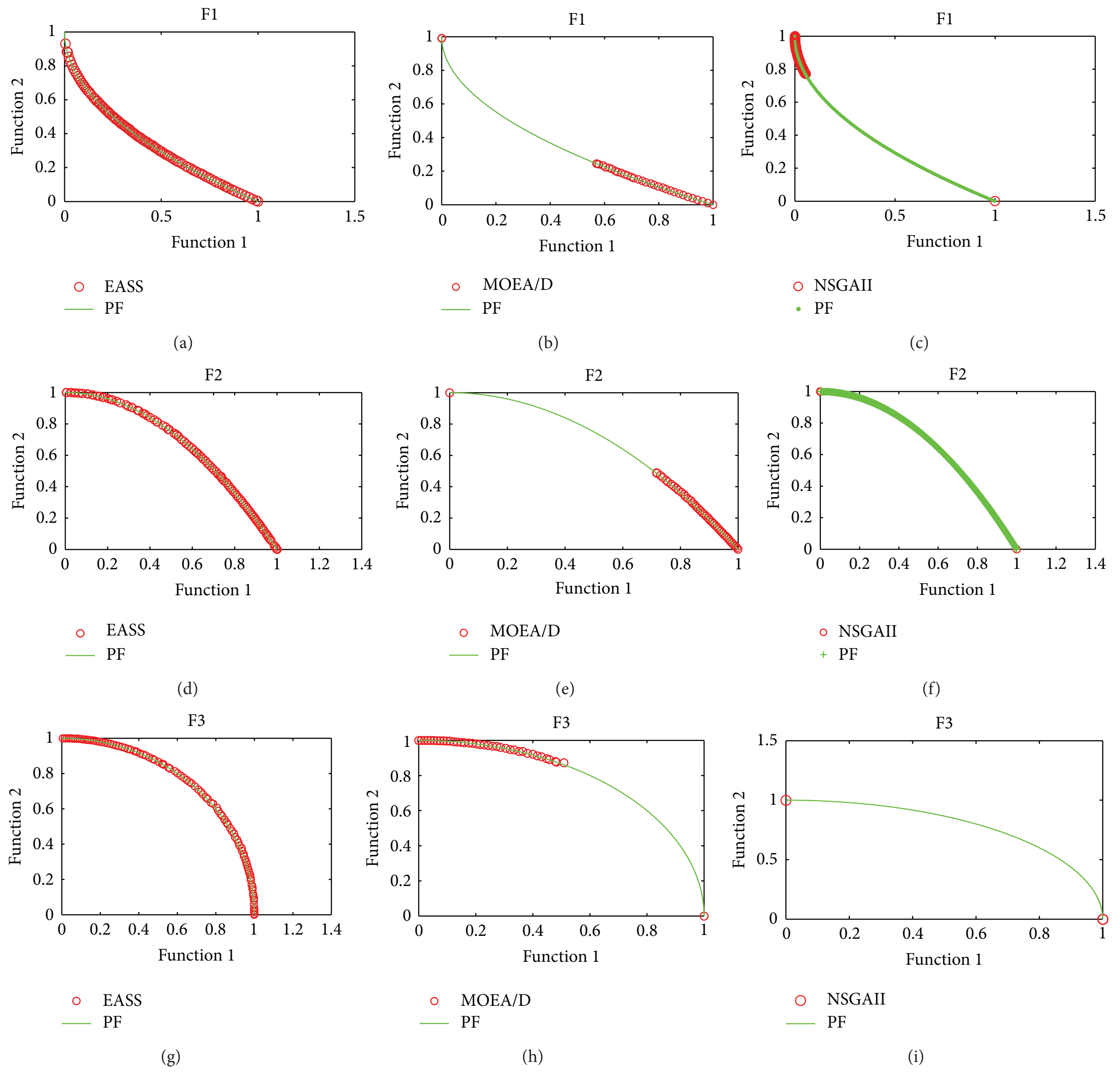
   (b) crossover rate is 1.0 and scaling factor is 0.5 in the DE operator;

   (c) distribution index is 20 and mutation probability is 0.1 in mutation operator.

(2) For two-objective and three-objective instances, the population size is 105 in the three algorithms. Weight vectors are generated by using the method which is used in [22]. The direction vectors of EASS are the weight vectors. The number of weight vectors is the same as the population size. The number of the weight vectors in the neighborhood in MOEA/D is 20 for all test problems. The Tchebycheff approach [22] is used for MOEA/D as an aggregate function. Other control parameters in MOEA/D are the same as in [22].

(3) Each algorithm is run 20 times independently for each test instance. All the three algorithms stop after 1000 generations.

4.3. Experimental Measures. In order to compare the performance of the different algorithms quantitatively, performance metrics are needed. In this paper, the following performance metrics are used to compare the performance of the different algorithms quantitatively: generational distance (GD) [42], inverted generational distance (IGD) [42], and hypervolume indicator (HV) [43]. GD measures how far the known Pareto front is away from the true Pareto front. If GD is equal to 0, all points of the known PF belong to the true PF. GD allows us to observe whether the algorithm can converge to some region in the true PF. IGD measures how far the true PF is away from the known PF. If IGD is equal to 0, the known PF contains every point of the true PF. IGD shows whether points of the known PF are evenly distributed throughout the true PF. Here, GD and IGD indicators are used simultaneously to observe whether the solutions are distributed over the entire PF and define a suboptimal solution as a good solution. In experiments, we select 500 evenly distributed points on the PF for four two-objective test instances and 1000 points for three-objective test instances. The hypervolume indicator has been used widely in evolutionary multiobjective optimization to evaluate the performance of search algorithms. It computes the volume of the dominated portion of the objective space relative to a reference point. Higher values of this performance indicator imply more desirable solutions. The hypervolume indicator measures both the convergence to the true Pareto front and diversity of the obtained solutions. In our experiments, the reference point is set to $(1,\ldots,1)$.



(a) (b) (c)

(d) (e) (f)

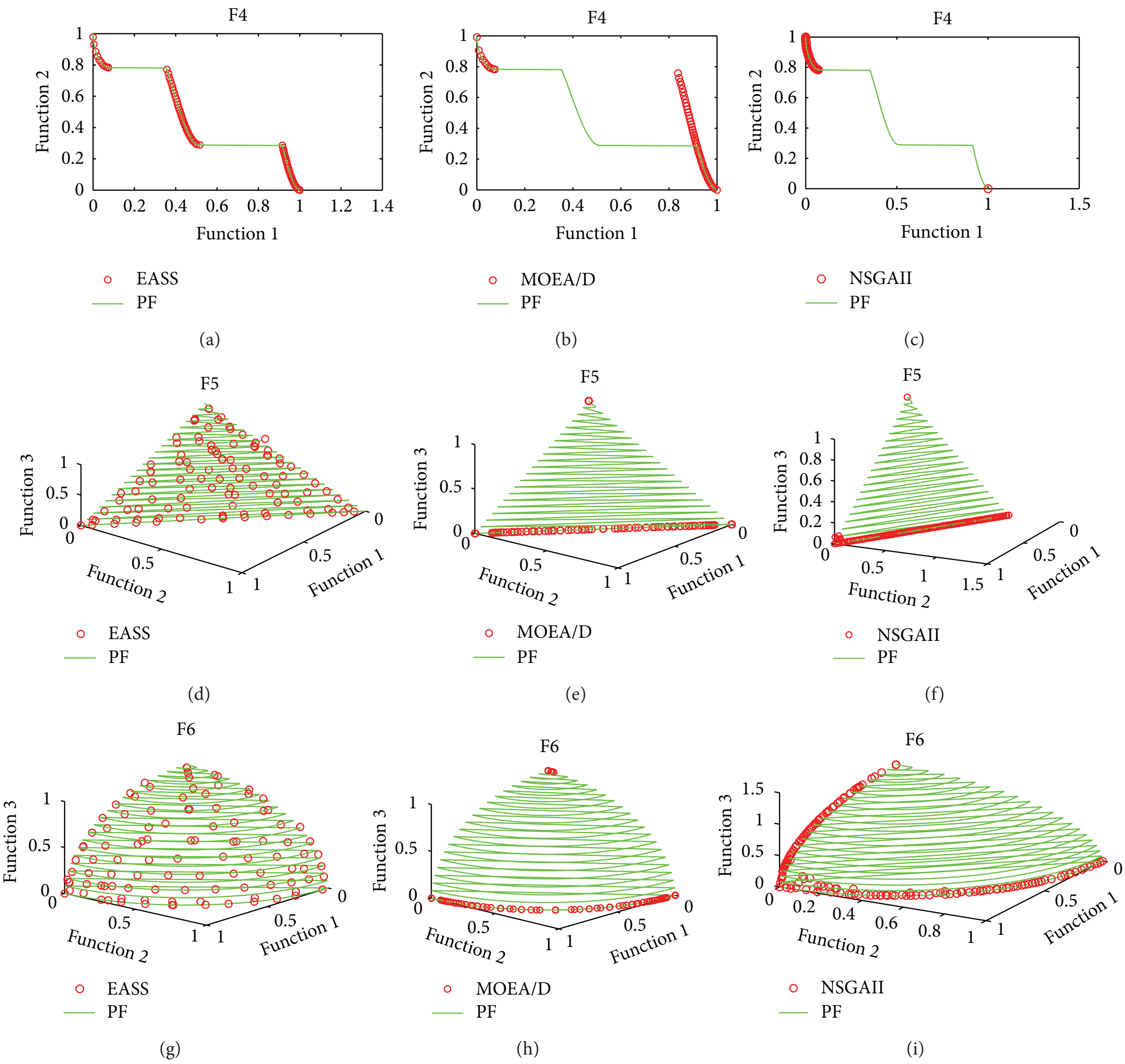
(g) (h) (i)

Figure 1: Solutions obtained by EASS, MOEA/D, and NSGAII on F1–F3.

4.4. Comparisons of EASS with MOEA/D and NSGAII. In this section, some simulation results and comparisons that prove the potential of EASS are presented and the comparisons mainly focus on three aspects: (1) the convergence of the obtained solutions to the true PF; (2) the coverage of the obtained solutions to the true PF; and (3) the diversity of the obtained solutions. Although such comparisons are not sufficient, they provide a good basis to estimate the performance of EASS.

To visually compare the performance of the three algorithms, the solutions obtained by them on these test problems are shown in Figures 1 and 2. Obviously, both MOEA/D and NSGAII cannot locate the global PF on any instance. In contrast, EASS can approximate the PFs of these instances quite well. These Figures indicate that the diversity of solutions obtained by the algorithm EASS is better than those obtained by MOEA/D and NSGAII on these test problems.

Table 2 presents the mean and standard deviation of IGD, GD, and HV obtained by EASS, MOEA/D, and NSGAII. Best solutions obtained are highlighted bold in this table. From the table, we can obtain that the mean values of IGD obtained by EASS are much smaller than those obtained by the other two algorithms for these six test problems, which indicates that for these test problems, the coverage



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

FIGURE 2: Solutions obtained by EASS, MOEA/D, and NSGAII on F4–F6.

of solutions obtained by EASS to the true PF is better than those obtained by MOEA/D and NSGAII. In addition, standard deviations of IGD obtained by EASS are very small for these test problems, which show that the performance of EASS is well stable on these problems. We can also see that the mean values of GD obtained by EASS are smaller than those obtained by the other two algorithms for test problems F1 and F3, which indicate that the convergence of solutions obtained by EASS to the true PF is better than those obtained by MOEA/D and NSGAII. For problems F2 and F4, the mean values of GD obtained by EASS are slightly larger than those obtained by NSGAII but smaller than those obtained by MOEA/D. For problems F5 and F6, the mean values of GD obtained by EASS are larger than those obtained by MOEA/D and NSGAII. For these four test problems, the convergence of solutions obtained by EASS to the true PF is worse than that obtained by NSGAII, which is because the solutions obtained by NSGAII and MOEA/D are concentrated in certain regions of the PF, but solutions obtained by EASS are distributed throughout the PF and converge to the PF. From the mean values of GD obtained by EASS, we can obtain that solutions obtained by EASS can well converge to the true PF for these six test problems. In terms of the mean value of HV, the values of HV obtained by EASS are much bigger than those obtained by other two algorithms, which illustrate that the diversities of solutions obtained by EASS are better than those obtained by MOEA/D and NSGAII.

Table 2: IGD, GD, and HV obtained by EASS, MOEA/D, and NSGAII on F1–F6.

	IGD		GD		HV	
	Mean	SD	Mean	SD	Mean	SD
F1						
EASS	**0.0042**	**0.0000**	**0.0012**	**0.0001**	**0.6605**	**0.0001**
MOEA/D	0.1271	0.0415	0.0053	0.0012	0.4308	0.0847
NSGAII	0.2697	0.0040	0.0037	0.0017	0.2890	0.0077
F2						
EASS	**0.0051**	**0.0002**	0.0014	0.0001	**0.3260**	**0.0002**
MOEA/D	0.1141	0.0106	0.0025	0.0008	0.2506	0.0088
NSGAII	0.3437	0.0427	**0.0005**	**0.0001**	0.0116	0.0465
F3						
EASS	**0.0055**	**0.0002**	**0.0019**	**0.0001**	**0.2065**	**0.0003**
MOEA/D	0.1758	0.1098	0.0019	0.0011	0.0978	0.0549
NSGAII	0.3753	0.0364	0.0030	0.0002	0.0027	0.0108
F4						
EASS	**0.0046**	**0.0001**	0.0014	**0.0001**	**0.5139**	**0.0003**
MOEA/D	0.2313	0.0263	0.0728	0.0187	0.1977	0.0729
NSGAII	0.2650	0.0010	**0.0011**	0.0018	0.2119	0.0009
F5						
EASS	**0.0423**	0.0008	0.0209	0.0005	**0.7719**	0.0025
MOEA/D	0.2999	**0.0000**	**0.0066**	0.0000	0.4810	**0.0000**
NSGAII	0.2898	0.0341	0.0171	**0.0005**	0.5062	0.0443
F6						
EASS	**0.0556**	0.0016	0.0259	0.0006	**0.4039**	0.0015
MOEA/D	0.3698	**0.0000**	**0.0075**	**0.0000**	0.1992	**0.0000**
NSGAII	0.2611	0.0326	0.0209	0.0011	0.3110	0.0306

5. Conclusion

In this paper, a new evolutionary algorithm based on decomposition of the objective space is designed to well maintain the diversity of obtained solutions. In order to achieve the goal, the objective space of an MOP is decomposed into a number of subobjective spaces, and the obtained solutions are classed to make each subobjective space have a solution. For each subobjective space, if a new solution will replace its current solution whose objective vector locates in this sub-region, the new solution must dominate the current solution and its objective vector locates in the sub-region; if the objective vector of the current solution does not locate in this sub-region, the new solution dominate the current solution or its objective vector locates in the sub-region. In such a way, good population diversity can be achieved, which is essential for solving some MOPs. In addition, in order to improve the convergence of all obtained solutions, the crowding distance is used to calculate the fitness value of a solution for the selection operators, which can make dominated solutions be more likely to be selected to generate new solutions. Experimental studies on six test instances have implied that this proposed algorithm can significantly outperform MOEA/D and NSGAII on these test problems.

The future work includes combination of this algorithm and other evolutionary search techniques and investigation of its performance on other hard multiobjective optimization problems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (nos. 61272119 and 61203372).

References

[1] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithms with dynamic population size and local exploration for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 5, no. 6, pp. 565–588, 2001.

[2] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multiobjective Problems*, Kluwer Academic Publishers, New York, NY, USA, 2002.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[4] A. Konak, D. W. Coit, and A. E. Smith, “Multi-objective optimization using genetic algorithms: a tutorial,” *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 992–1007, 2006.

[5] C. K. Goh, K. C. Tan, D. S. Liu, and S. C. Chiam, “A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design,” *European Journal of Operational Research*, vol. 202, no. 1, pp. 42–54, 2010.

[6] Z. Yong, G. Dun-wei, and G. Na, “Multi-objective optimization problems using cooperative evolvement particle swarm optimizer,” *Journal of Computational and Theoretical Nanoscience*, vol. 10, no. 3, pp. 655–663, 2013.

[7] D. Matjaz, T. Roman, and F. Bogdan, “Asynchronous master-slave parallelization of differential evolution for multi-objective optimization,” *Evolutionary Computation*, vol. 21, no. 2, pp. 261–291, 2013.

[8] R. Liu, X. Wang, J. Liu, L. Fang, and L. Jiao, “A preference multi-objective optimization based on adaptive rank clone and differential evolution,” *Natural Computing*, vol. 12, no. 1, pp. 109–132, 2013.

[9] M. Gong, L. Jiao, H. Du, and L. Bo, “Multiobjective immune algorithm with nondominated neighbor-based selection,” *Evolutionary Computation*, vol. 16, no. 2, pp. 225–255, 2008.

[10] R. Shang, L. Jiao, F. Liu, and W. Ma, “A novel immune clonal algorithm for MO problems,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 35–50, 2012.

[11] Q. H. Wu, Z. Lu, M. S. Li, and T. Y. Ji, “Optimal placement of FACTS devices by a group search optimizer with multiple producers,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’08)*, pp. 1033–1039, June 2008.

[12] C. A. Coello Coello, “Evolutionary multi-objective optimization: a historical view of the field,” *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[13] X. Li and H.-S. Wong, “Logic optimality for multi-objective optimization,” *Applied Mathematics and Computation*, vol. 215, no. 8, pp. 3045–3056, 2009.

[14] A. C. Coello Coello and G. T. Pulido, “Multiobjective Optimization using a micro-genetic algorithm,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO ’01)*, pp. 274–282, San Francisco, Calif, USA, 2001.

[15] J. D. Knowles and D. W. Corne, “The Pareto-envelope based selection algorithm for multiobjective optimization,” in *Proceedings of the 6th International Conference on Parallel Problem Solving from Cature (PPSN ’00)*, pp. 839–848, 2000.

[16] A. Zinflou, C. Gagné, M. Gravel, and W. L. Price, “Pareto memetric algorithm for multiple objective optimization with an industria lapplication,” *Journal of Heuristics*, vol. 14, no. 4, pp. 313–333.

[17] A. Zinflou, C. Gagné, and M. Gravel, “GISMOO: a new hybrid genetic/immune strategy for multiple-objective optimization,” *Computers & Operations Research*, vol. 39, no. 9, pp. 1951–1968, 2012.

[18] R. Shang, L. Jiao, F. Liu, and W. Ma, “A novel immune clonal algorithm for MO problems,” *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 35–50, 2012.

[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, Mass, USA, 1989.

[20] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: improving the strength Pareto evolutionary algorithm,” in *Proceedings of the Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problem (EUROGEN ’02)*, pp. 95–100, Athens, Greece, 2002.

[21] B. Soylu and M. Köksalan, “A favorable weight-based evolutionary algorithm for multiple criteria problems,” *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 2, pp. 191–205, 2010.

[22] Q. Zhang and H. Li, “MOEA/D: a multiobjective evolutionary algorithm based on decomposition,” *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[23] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, “Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1820–1825, 2009.

[24] H. Li and Q. Zhang, “Multiobjective optimization problems with complicated pareto sets, MOEA/ D and NSGA-II,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[25] W. K. Mashwani, “MOEA/D with DE and PSO: MOEA/D-DE+PSO,” in *Proceedings of the 31st SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pp. 217–221, 2011.

[26] H. Li and D. Landa-Silva, “An adaptive evolutionary multi-objective approach based on simulated annealing,” *Evolutionary Computation*, vol. 19, no. 4, pp. 561–595, 2011.

[27] S. Z. MartInez and C. A. C. Coello, “A direct local search mechanism for decomposition-based multi-Objective evolutionary algorithms,” in *Proceedings of the IEEE World Congress on Computational Intelligence*, pp. 3431–3438, 2012.

[28] K. Sindhya, K. Miettinen, and K. Deb, “A hybrid framework for evolutionary multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 495–511, 2012.

[29] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, “Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’09)*, pp. 2508–2515, May 2009.

[30] V. A. Shim, K. C. Tan, and K. K. Tan, “A hybrid estimation of distribution algorithm for solving the multi-objective multiple traveling salesman problem,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’12)*, pp. 1–8, 2012.

[31] Y.-H. Chan, T.-C. Chiang, and L.-C. Fu, “A two-phase evolutionary algorithm for multiobjective mining of classification rules,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC ’10)*, July 2010.

[32] Y.-y. Tan, Y.-c. Jiao, H. Li, and X.-k. Wang, “MOEA/D + uniform design: a new version of MOEA/D for optimization problems with many objectives,” *Computers & Operations Research*, vol. 40, no. 6, pp. 1648–1660, 2013.

[33] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, “An adaptive weight vector adjustment based multiobjective evolutionary algorithm,” *Evolutionary Computation (MIT)*, 2013.

[34] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, “Adaptation of scalarizing functions in MOEA/D: an adaptive scalarizing function-based multiobjective evolutionary algorithm,” *Evolutionary Multi-Criterion Optimization Lecture Notes in Computer Science*, vol. 5467, pp. 438–452, 2009.

[35] D. A. van Veldhuizen, *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations [Ph.D. thesis]*, Dept. Electr. Comput. Eng. Graduate School Eng., Air Force Institute of Technology, Wright-Patterson AFB, Ohio, USA, 1999.

[36] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[37] Q. F. Zhang and P. N. Suganthan, Final report on CEC'09 MOEA competition. Technical report, the School of CS and EE, University of Essex, UK and School of EEE, Nangyang Technological University, Singapore, 2009, http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm.

[38] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multiobjective optimization test problems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '02)*, pp. 825–830, 2002.

[39] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[40] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, New York, NY, USA, 2001.

[41] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[42] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[43] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference (GECCO '06)*, pp. 1141–1148, Seattle, Wash, USA, July 2006.