

Research Article

Multiobjective Differential Evolution Algorithm with Multiple Trial Vectors

Yuelin Gao¹ and Junmei Liu²

¹ Institute of Information and System Science, Beifang University of Nationalities, Yinchuan 750021, China

² Department of Mathematics, Yinchuan College, China University of Mining and Technology, Yinchuan 750011, China

Correspondence should be addressed to Yuelin Gao, gaoyuelin@263.net

Received 29 May 2012; Accepted 5 June 2012

Academic Editor: Yonghong Yao

Copyright © 2012 Y. Gao and J. Liu. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a multiobjective differential evolution algorithm with multiple trial vectors. For each individual in the population, three trial individuals are produced by the mutation operator. The offspring is produced by using the crossover operator on the three trial individuals. Good individuals are selected from the parent and the offspring and then are put in the intermediate population. Finally, the intermediate population is sorted according to the Pareto dominance relations and the crowding distance, and then the outstanding individuals are selected as the next evolutionary population. Comparing with the classical multiobjective optimization algorithm NSGA-II, the proposed algorithm has better convergence, and the obtained Pareto optimal solutions have better diversity.

1. Introduction

Multiobjective optimization problems (MOPs) are different from single objective optimization problems (SOPs). For MOPs, the objective functions are often conflict, so the best solution often does not exist. It is difficult to find the optimal solution. The MOPs need to find a noninferior solution set that is called Pareto optimal solution set (or nondominated optimal solution set). The key is to find a solution set which is as close to the Pareto front and uniformly distributed as possible. In the past ten years, evolutionary algorithms have been widely used for solving multiobjective optimization problems. The typical algorithm contains NSGA [1], NSGA-II [2], SPEA [3], SPEA2 [4], and so on.

Differential evolution (DE) [5] is an important branch of the evolutionary algorithms. Because of its easiness to use and robustness, it has been widely applied in many fields. The basic DE algorithm is a greedy algorithm and is used to solve the SOPs. The individual choice is based on the objective function value. In solving MOPs, the objective function value is a vector value and the individual choice is based on the vector value. Some scholars have

improved DE algorithm to solve MOPs. Abbas et al. [6] proposed a differential evolution algorithm based on the Pareto front's (PDE) to solve MOPs. Later, the PDE algorithm is improved to an adaptive PDE algorithm (SPDE) [7]. Madavan [8] proposed the differential evolution algorithm based on the Pareto (PDEA). Xue et al. [9] and Robi and Filipi [10] and Qian and Li [11] proposed the different kind of multiobjective differential evolution algorithm. Gong and Cai [12] proposed an improved multiobjective differential evolution based on the Pareto-adaptive ε -dominance and the orthogonal design.

Here, a multiobjective differential evolution algorithm with multiple trial vectors (MTVDE) is proposed. In this algorithm, for each individual in the parent, three different mutation operators and a crossover operator are used to produce three trial individuals. First, we compare the three trial individuals. If an individual Pareto-dominate another, the better one is retained. If two individuals do not dominate each other, both two individuals are retained. Secondly, retained trial individuals compare with the parent individual: if an individual Pareto-dominate another, put the nondominated individual in the intermediate population, and if the two individuals do not dominate each other, two individuals are put in the intermediate population. Finally, in the intermediate population, the individuals are sorted according to the Pareto dominance relations and the crowding distance. We eliminate some low-quality individuals from the intermediate population and retain some fine individuals to form the next evolutionary population. The individuals are sorted according to the nondominated relations, and individuals with low ordinal value are better individuals. When individuals are with the same ordinal, the individual of larger crowding distance is better than the individual of smaller crowding distance.

2. Multiobjective Optimization Problem and Related Concepts

Below, we give several concepts related to this paper. Because the maximum optimization problems and minimum optimization problem can be transformed into each other, here we only consider the minimum multiobjective optimization problem.

Definition 2.1 (multiobjective optimization problems, MOPs). The MOPs are described as follows:

$$\begin{aligned}
 \min \quad & y = f(x) = (f_1(x), f_2(x), \dots, f_k(x)), \\
 \text{s.t.} \quad & g(x) = (g_1(x), g_2(x), \dots, g_l(x)) \leq 0, \\
 & h(x) = (h_{l+1}(x), h_{l+2}(x), \dots, h_m(x)) = 0, \\
 & x = (x_1, x_2, \dots, x_n) \in X, \\
 & y = (y_1, y_2, \dots, y_k) \in Y,
 \end{aligned} \tag{2.1}$$

where $y \in R^k$ indicates the target vector, $x \in R^n$ indicates the decision vector, X indicates decision space of the decision vector x , Y indicates the target space of the target vector y .

Definition 2.2 (the dominance relationship of the target vector). The target vector $u = (u_1, u_2, \dots, u_k)$ is better than $v = (v_1, v_2, \dots, v_k)$ if and only if for all $i \in (1, 2, \dots, k)$ $u_i \leq v_i$ and there exists at least one $i \in (1, 2, \dots, k)$ such that $u_i < v_i$ was established, then the target the vector u Pareto better than v , or v Pareto inferior to u .

Definition 2.3 (decision vector relations of domination). If the decision vector $z = (z_1, z_2, \dots, z_n)$ Pareto-dominates $w = (w_1, w_2, \dots, w_n)$ if and only if $f(z)$ Pareto is superior to $f(w)$, we call the decision vector z Pareto-dominating w or the decision vector w is Pareto dominated by z .

Definition 2.4 (Pareto optimal solution). Decision vector $x_u \in F$ (F is the feasible solution set) is a Pareto optimal solution, if and only if an individual does not exist in the feasible solution set F dominate x_u , then x_u is called the MOPs' Pareto optimal solution, or x_u is called the MOPs' nondominated optimal solution. The set of all Pareto optimal solution is known as the MOPs' Pareto optimal solution set or nondominated optimal solution set.

Definition 2.5 (Pareto front). For given MOPs, Pareto optimal solution set corresponding to the objective function value vector is known as the MOPs' Pareto front. Therefore, our task is to make the current nondominated optimal solution set as close as possible to the theoretical Pareto optimal solution set, from the view of Pareto front is to ask the Pareto front as possible as close to the theoretical Pareto front and uniformly distributed.

Definition 2.6 (individual nondominant relationship sort, INDR). There are n individuals in the population P . For each individual x_i , $i = 1, 2, \dots, n$, S_i represents the set of individuals that are dominated by x_i , n_i represents the number of individuals dominating the individual x_i . Finally, if $n_i = 0$, then x_i belongs to the first level. The sequence of x_i is $\text{rank}_i = 1$, x_i joined to F_1 (F_1 indicates the individual set of ordinal value is 1). The counter of the initial sequence value $\text{rank} = 1$, if $F_{\text{rank}} \neq \phi$, (F_{rank} represents the set of individuals' sequence value $\text{rank} = 1$) each individual x_i in F_{rank} , each individual x_i , $n_j = n_j - 1$ in the corresponding S_i . Finally, if $n_i = 0$, then x_j belongs to the $\text{rank} + 1$ level. The sequence value of x_j is $\text{rank}_j = \text{rank} + 1$. Update sequence value's counter $\text{rank} = \text{rank} + 1$, x_j joined to F_{rank} . As a result, we can calculate the ordinal value of each individual in the population P .

Definition 2.7 (the same level of individuals' crowding distance) (ICD). Let I be a set of individuals of a certain grade. All the target m ($m = 1, 2, \dots$) individuals are sorted according to the objective function value in I . The first and the last individuals' crowding distances are infinite. The other individuals' crowding distance is calculated as subtract the two neighbor individuals' objective function and then square it. Then sum the crowding distance on all objective functions as the individuals' crowding distance. The individuals are sorted according to crowding distance in the same level.

3. Multiobjective Differential Evolution Algorithm with Multiple Trial Vectors (MTVDE)

3.1. The Mutation Operation

Mutation components are difference vectors of the parent generation. Each vector is generated by the different individuals in the parent population. The mutation operation on the basis of the difference vector and its equation is

$$\begin{aligned} v_{i1t} &= x_{it} + F(x_{r_1t} - x_{r_2t}), \\ v_{i2t} &= x_{r_1t} + F(x_{r_2t} - x_{r_3t}), \\ v_{i3t} &= x_{it} + F(x_{r_1t} - x_{r_2t}) + F(x_{r_3t} - x_{r_4t}), \end{aligned} \quad (3.1)$$

where $v_{i1t}, v_{i2t}, v_{i3t}$ are the mutated individuals, x_{it} is individual of parent population. $x_{r_1t}, x_{r_2t}, x_{r_3t}, x_{r_4t}$ are different from each other, and they are different from x_{it} . $F \in (0, 2]$ is the scaling factor. It indicates the influence degree of the difference vector to the offspring. From the mutation operator, each individual in parent population produces three individuals through the mutation operator, which can enhance the diversity of the population.

The value of F has a certain influence on the algorithm performance. If F is too large, the convergence rate slows, and the algorithm runs longer. If F is too small, the diversity of the population decreases, according to experimental tests, and F is often set to 0.5.

3.2. Crossover Operator

The mutated individual $v_i(t+1)$, ($i = i_1, i_2, i_3$), and the individual $x_i(t)$ in parent population use the crossover operator to generate the trial individual $u_i(t+1)$, ($i = i_1, i_2, i_3$). The crossover operator is expressed as

$$u_{ij}(t+1) = \begin{cases} v_{ij}(t+1), & \text{if } (\text{rand}(0,1) \leq \text{CR}) \quad \text{or} \quad (j = \text{rand } i(1,D)), \\ x_{ij}(t), & \text{otherwise,} \end{cases} \quad (3.2)$$

where $\text{rand}(0,1)$ is a uniformly distributed random number between $(0,1)$, $\text{rand } i(1,D)$ is the integer randomly selected from $\{1, 2, \dots, D\}$, which can insure at least one $u_i(t+1)$ is from $v_i(t+1)$. $\text{CR} \in [0, 1]$ is the crossover probability. It is used to control the variable in $u_i(t+1)$ to be from $v_i(t+1)$ or $x_i(t)$. If $\text{CR} = 1$, then $u_i(t+1) = v_i(t+1)$.

3.3. Selecting Operation

For MOPs, this paper uses the following criteria to select the next evolution population.

- (1) For each individual in the parent population, three different mutated operators and a crossover operator are used to produce three trial individuals. For the three trial individuals and the parent individual, the one who Pareto-dominates another is retained in the intermediate population. If the two individuals do not dominate each other, both are retained.
- (2) Some low-quality individuals in the intermediate population need elimination. Individuals are sorted in the intermediate population according to the Pareto dominance relations and the crowding distance of individuals. First of all, the individuals are divided into several levels according to the non-dominant relations in the intermediate population. Individuals in the same level are sorted according to the crowding distance. We choose the former NP individuals to form the next evolution population.

3.4. MTVDE Description of the Algorithm

Step 1. Set the basic parameters: the population size NP, the scaling factor F , the maximum evolution generation, and the crossover probability factor CR.

Step 2. Initialize the population.

Step 3. Mutation is according to the mutation operation (3.1) to generate three variation individuals.

Step 4. Crossover is according to the crossover operation (3.2) to generate three test individuals.

Step 5. Select the next evolution population.

Step 6. If the maximum iteration is reached, stop and output optimal solution, otherwise return to Step 3.

4. Numerical Experiments

4.1. Algorithm Performance Evaluation Criteria

Multiobjective optimization algorithm performance can be evaluated by two criteria.

- (1) The obtained nondominated optimal solution set needs to be as close as possible to the true Pareto front.
- (2) The obtained nondominated optimal solution set need as possible as uniformly distribute to the true Pareto front.

The two indexes are used to test the convergence and distribution of the solution set, respectively. The definitions of the approximation index and the uniformity index are given below.

- (a) Approximation index is defined as follows:

$$\gamma = \frac{\sum_{i=1}^n d_i}{n}, \quad (4.1)$$

where n is the number of vectors in the set of nondominated solutions found so far, d_i is the Euclidean distance (measured in objective space) between each of these solutions and the nearest member of the Pareto optimal set. d_i is calculated as $d_i = \min\{|X_i - Y_j|, j = 1, 2, \dots, N\}$, where N denotes the total number of the true Pareto front vectors, X_i denotes the nondominated optimal solutions set, Y_j denotes the true Pareto front, and $|\cdot|$ denotes Euclidean distance. The value of γ is smaller, the degree of algorithm approximating to true Pareto front is higher. It is clear that a value $\gamma = 0$ indicates that all the generated elements are in the Pareto front.

- (b) The uniformity index is defined as follows:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{n-1} |d_i - \bar{d}|}{d_f + d_l + (n-1)\bar{d}}, \quad (4.2)$$

where d_i is the Euclidean distance between neighboring solutions in the obtained nondominated solutions set and \bar{d} is the mean of all d_i . The parameters d_f and d_l

are the Euclidean distances between the extreme solutions and the boundary solutions of the obtained nondominated set. d_f is calculated as $d_f = |X_f - Y_f|$, where X_f is the leftmost vector of obtained nondominated optimal target vector, Y_f is the leftmost vector of theoretical Pareto front. d_l is calculated as $d_l = |X_l - Y_l|$, where X_l is the rightmost vector of obtained nondominated optimal target vector, Y_l is the rightmost vector of theoretical Pareto front, $|\cdot|$ denotes Euclidean distance. The value of Δ is smaller, the degree of obtained nondominated solutions' diversity and uniform distribution is better.

4.2. Numerical Experiments and Analysis

In order to know how competitive the MTVDE algorithm was, it was compared with NSGA-II. In algorithm NSGA-II, $\eta_c = 20$, $\eta_m = 20$ Pool size = 100, Tour size = 2. In MTVDE algorithm, CR = 0.9 except the problem ZDT4 that CR = 0.3. For all test functions, the population size of the two algorithms is taken as 100, and the number of iterations is taken as 250. Figures 1, 2, 3, 4, 5, 6, 7 and 8 show the results of the MTVDE algorithm by testing eight multiobjective functions. It can be seen from the figures the MTVDE algorithm produces a more uniform solution, and the solution quality is better than NSGA-II.

From Tables 1, 2, 3, 4, 5 and 6 eight testing functions are chosen to test the approximation index, uniformity index (mean value and variance in 10 runs). For the first five functions, the MTVDE algorithm is compared with the five classic algorithms in [8, 9, 11, 12]. For the rest three functions, the MTVDE algorithm is compared with NSGA2.

N/A denotes the algorithm does not calculate the index; the PDEA algorithm in [7] calculates the index GD (generational distance) instead of calculating the index (the convergence metric). The MODE algorithm in [7] did not calculate the index (diversity metric). We can be seen from Tables 1–6 that MTVDE algorithm has better convergence solutions except the problems ZDT1, ZDT2, and SCH.

In this paper, the population size is 250. Through experiments, we found that the increasing of population size has no effect on performance of the algorithm but only can increase the running time. In order to detect the impact of the scaling factor and the crossover probability on performance of algorithm, we make 50 times independent experiments for problem SCH. In order to test the impact of F and CR on the algorithm performance, it can be found that the F and CR have little effect on the algorithm performance by the testing data. Through comparison, $F = 0.5$, CR = 0.9 results are a little better, but the algorithm performance (approximation index and uniformity index) is not particularly sensitive to the values of F and CR (Table 7).

5. Summary

In MTVDE algorithm, the improved DE algorithm is used to solve MOPs. The main difference between MTVDE algorithm and other MOEA algorithms is the three test vectors and the new select method. Eight benchmark functions are used to test the MTVDE algorithm. Experimental results show that MTVDE algorithm is better than most MOEA. In order to find the best solution, different values of crossover probability are tested, the results of ZDT4 are very sensitive to the crossover probability (crossover probability is 0.3 better results). In short, the algorithm in this paper can effectively converge to the Pareto front of the problem

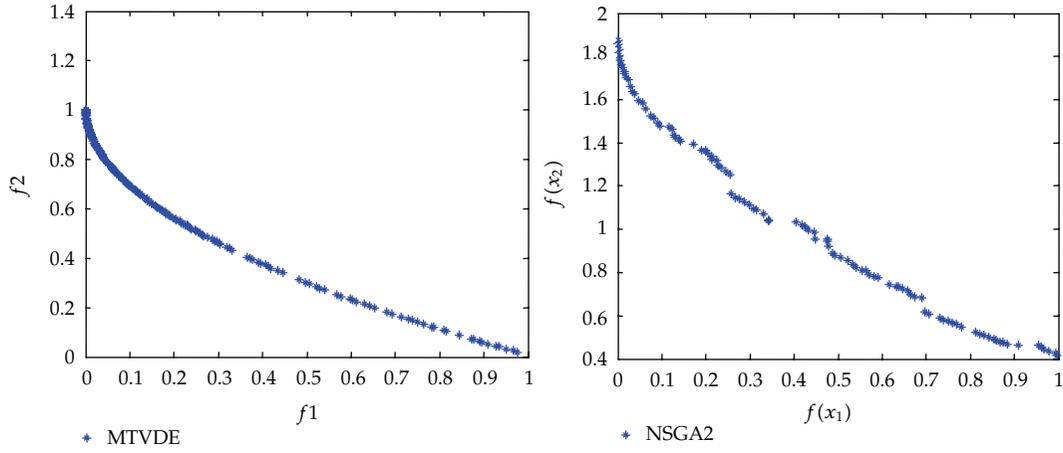


Figure 1: ZDT1.

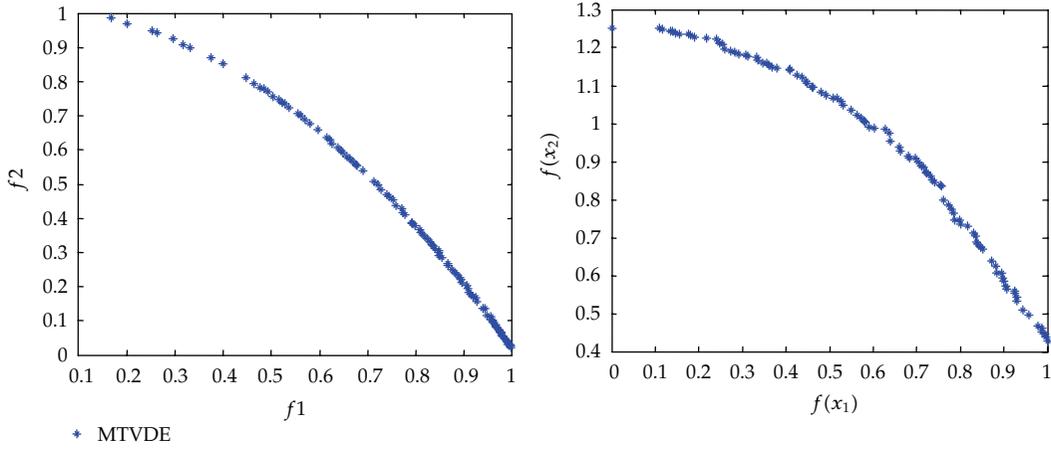


Figure 2: ZDT2.

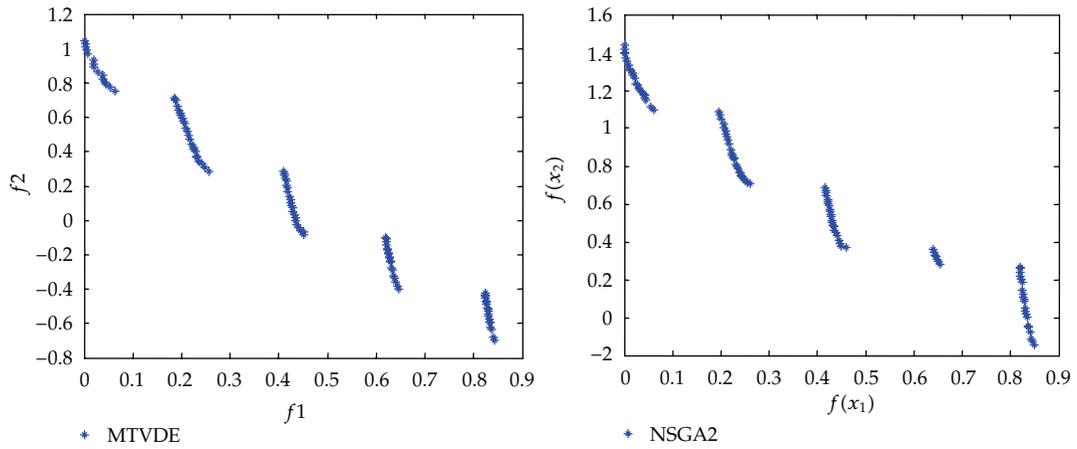


Figure 3: ZDT3.

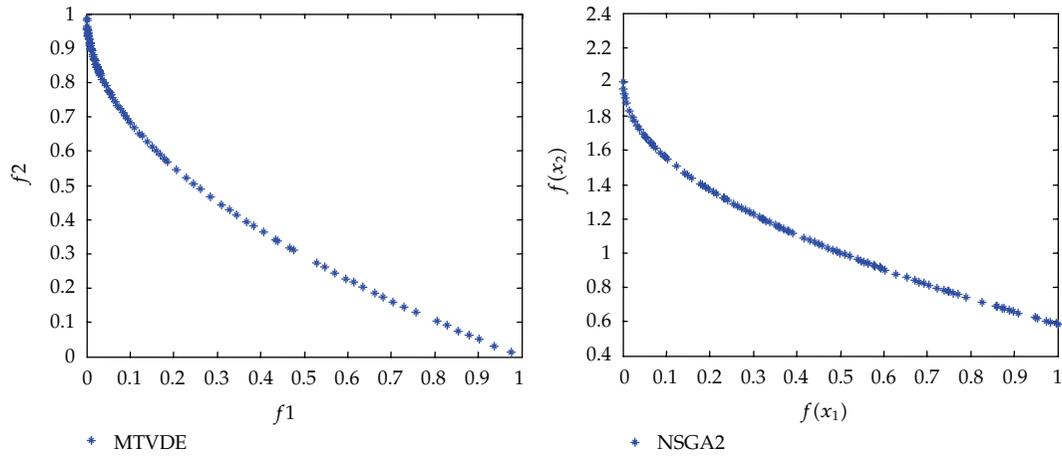


Figure 4: ZDT4.

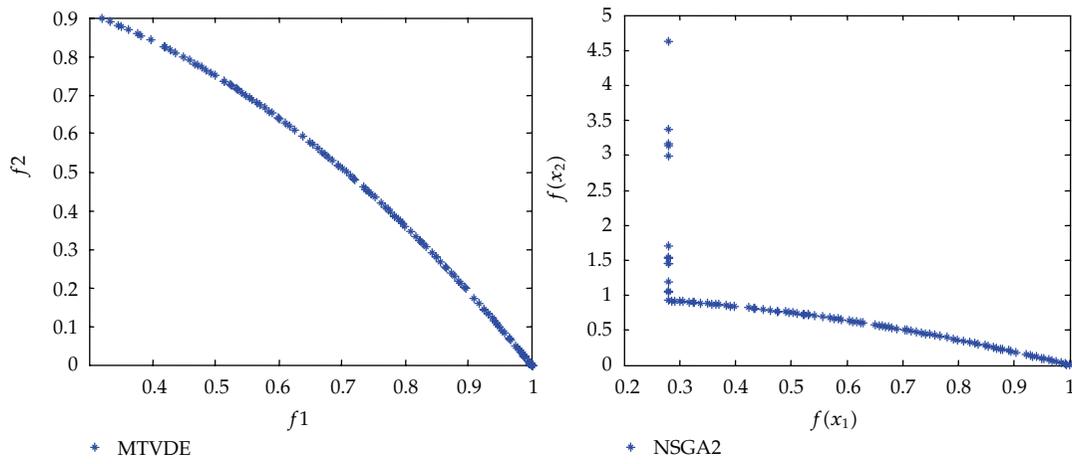


Figure 5: ZDT6.

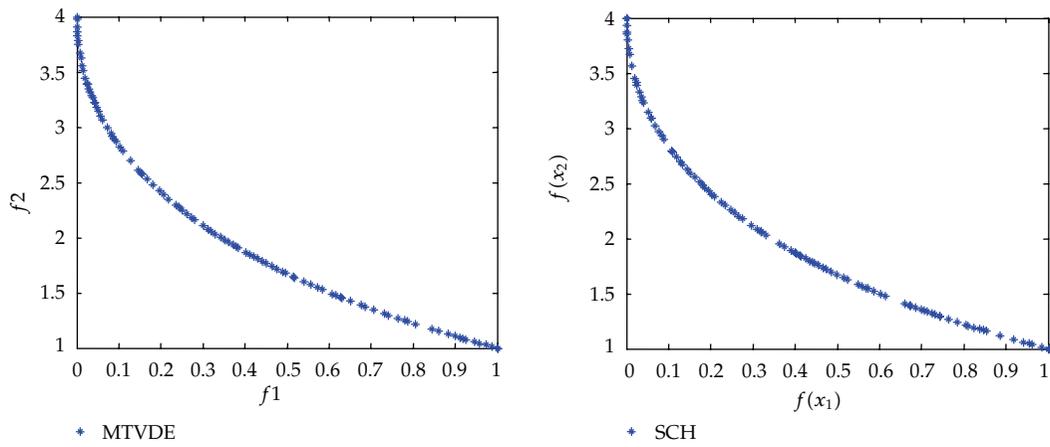


Figure 6: SCH.

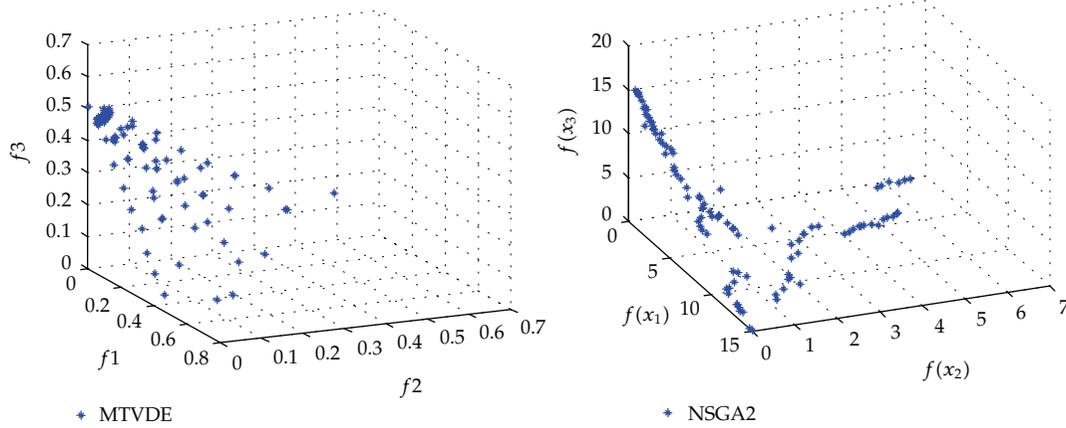


Figure 7: DTLZ1.

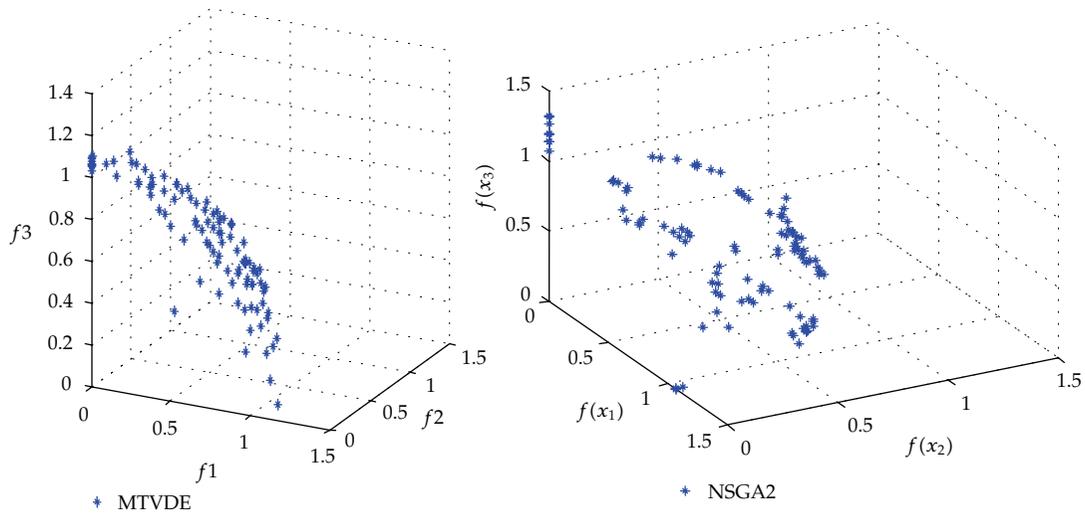


Figure 8: DTLZ2.

Table 1: Testing result's statistics of the problem ZDT1.

Algorithm	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.033482 ± 0.004750	0.390307 ± 0.001876
NSGA-II (binary-coded)	0.000894 ± 0.000000	0.463292 ± 0.041622
SPEA	0.001799 ± 0.000001	0.784525 ± 0.004440
PAES	0.082085 ± 0.008679	1.229794 ± 0.000742
PDEA	N/A	0.298567 ± 0.000742
MODE	0.005800 ± 0.000000	N/A
MTVDE	0.001200 ± 0.000002	0.505800 ± 0.003200

Table 2: Testing result's statistics of the problem ZDT2.

Algorithm	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.072391 ± 0.031689	0.430776 ± 0.004721
NSGA-II (binary-coded)	0.000824 ± 0.000000	0.435112 ± 0.024607
SPEA	0.001339 ± 0.000000	0.755184 ± 0.004521
PAES	0.126276 ± 0.036877	1.165942 ± 0.007682
PDEA	N/A	0.317958 ± 0.001389
MODE	0.005500 ± 0.000000	N/A
MTVDE	0.004700 ± 0.000006	0.645900 ± 0.002600

Table 3: Testing result's statistics of the problem ZDT3.

Algorithm	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.114500 ± 0.004940	0.738540 ± 0.019706
NSGA-II (binary-coded)	0.043411 ± 0.000042	0.575606 ± 0.005078
SPEA	0.047517 ± 0.000047	0.672938 ± 0.003587
PAES	0.023872 ± 0.000010	0.789920 ± 0.001653
PDEA	N/A	0.623812 ± 0.000225
MODE	0.021560 ± 0.000000	N/A
MTVDE	0.006000 ± 0.000002	0.604400 ± 0.000326

Table 4: Testing result's statistics of the problem ZDT4.

Algorithm	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.513053 ± 0.118460	0.702612 ± 0.064648
NSGA-II (binary-coded)	3.227636 ± 7.307630	0.479475 ± 0.009841
SPEA	7.340299 ± 6.572516	0.798463 ± 0.014616
PAES	0.854816 ± 0.527238	0.870458 ± 0.101399
PDEA	N/A	0.840852 ± 0.035741
MODE	0.638950 ± 0.500200	N/A
MTVDE	0.005200 ± 0.000002	0.744500 ± 0.003500

Table 5: Testing result's statistics of the problem ZDT6.

Algorithm	Convergence metric	Diversity metric
NSGA-II (real-coded)	0.296564 ± 0.013135	0.668025 ± 0.009923
NSGA-II (binary-coded)	7.806798 ± 0.001667	0.644477 ± 0.035042
SPEA	0.221138 ± 0.000449	0.849389 ± 0.002713
PAES	0.085469 ± 0.006664	1.153052 ± 0.003916
PDEA	N/A	0.473074 ± 0.021721
MODE	0.026230 ± 0.000861	N/A
MTVDE	$0.000414 \pm 3.4014e - 009$	0.926300 ± 0.000895

Table 6: Testing result's statistics of the problems SCH, DTLZ1, and DTLZ2.

Problem	Algorithm	Convergence metric	Diversity metric
SCH	NSGA-II	$0.0076 \pm 6.2023e - 008$	$0.8471 \pm 1.9253e - 004$
	MTVDE	$0.0084 \pm 1.2296e - 007$	$0.8728 \pm 8.0547e - 005$
DTLZ1	NSGA-II	7.459200 ± 12.42210	1.02050 ± 0.005900
	MTVDE	0.023200 ± 0.000014	0.774900 ± 0.002300
DTLZ2	NSGA-II	0.659100 ± 0.004000	0.914300 ± 0.002700
	MTVDE	0.512900 ± 0.001100	0.647000 ± 0.001500

Table 7

F	CR	Convergence metric	Diversity metric
0.1	0.9	$0.0086 \pm 1.9976e - 007$	$0.9551 \pm 1.8152e$
0.3	0.9	$0.0090 \pm 1.4876e - 007$	$0.9361 \pm 1.6769e$
0.5	0.9	$0.0084 \pm 1.2296e - 007$	$0.8728 \pm 8.0547e$
0.7	0.9	$0.0091 \pm 1.9254e - 007$	$0.9281 \pm 1.4343e$
0.9	0.9	$0.0088 \pm 9.4253e - 008$	$0.9276 \pm 1.9030e$
0.5	0.1	$0.0088 \pm 7.6655e - 008$	$0.9258 \pm 1.6304e$
0.5	0.3	$0.0089 \pm 1.4744e - 007$	$0.9283 \pm 2.0532e$
0.5	0.5	$0.0091 \pm 2.8769e - 008$	$0.9327 \pm 1.1066e$
0.5	0.7	$0.0091 \pm 1.5230e - 007$	$0.9313 \pm 5.6436e$
0.5	0.9	$0.0084 \pm 1.2296e - 007$	$0.8728 \pm 8.0547e$

and maintain the diversity of the Pareto optimal solution set. It is an effective algorithm for solving multiobjective optimization problems. In the future work, we intend to use MTVDE algorithm or appropriate improved MTVDE algorithms for solving other complex constrained MOPs.

Acknowledgments

The work is supported by the Foundations of National Natural Science China under Grant no. 60962006 and by the Foundations of Research Projects of State Ethnic Affairs Commission.

References

- [1] N. Srinivas and K. Deb, "Multi-objective optimization using non-dominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.
- [2] 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.
- [3] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.
- [4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm," Tech. Rep. 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Zurich, Switzerland, 2001.
- [5] 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.

- [6] H. A. Abbass, R. Sarker, and C. Newton, "PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems," in *Proceedings of the Congress on Evolutionary Computation (CEC'01)*, pp. 971–978, May 2001.
- [7] H. A. Abbass, "PDE: the self-adaptive Pareto differential evolution algorithm," in *Proceedings of the Congress on Evolutionary Computation (CEC'02)*, vol. 1, pp. 831–836, IEEE Service Center, Piscataway, NJ, USA, 2002.
- [8] N. K. Madavan, "Multi-objective optimization using a Pareto differential evolution approach," in *Proceedings of the Congress on Evolutionary Computation (CEC'02)*, vol. 2, pp. 1145–1150, IEEE Service Center, Piscataway, NJ, USA, 2002.
- [9] F. Xue, A.C. Sanderson, and R.J. Graves, "Pareto-based multi-objective differential evolution," in *Proceedings of the Congress on Evolutionary Computation (CEC'03)*, vol. 2, pp. 862–869, IEEE Press, Canberra, Australia, 2003.
- [10] T. Robi and B. Filipi, "DEMO: differential evolution for multi-objective optimization," in *Lecture Notes in Computer Science*, pp. 520–533, Springer, Berlin, Germany, 2005.
- [11] W. Qian and A. li, "Adaptive differential evolution algorithm for multiobjective optimization problems," *Applied Mathematics and Computation*, vol. 201, no. 1-2, pp. 431–440, 2008.
- [12] W. Gong and Z. Cai, "An improved multiobjective differential evolution based on Pareto-adaptive ϵ -dominance and orthogonal design," *European Journal of Operational Research*, vol. 198, no. 2, pp. 576–601, 2009.