An Adaptive Multiobjective Differential Evolution Algorithm

Fangqing Gu, Hai-lin Liu
School of Applied Mathematics Guangdong University of Technology Guangzhou, China
Email: gufangqing84@sina.com
E-mail: hlliu@gdut.edu.cn

Abstract— **The mutation strategy and control parameter have a significant influence on the performance of differential evolution. An local and global mutation operator based subregion and external set strategies are proposed in this paper. They use the idea of direct simplex method of mathematical programming. It is advantageous to search the better solutions in the search space. The local mutation operator is applied to improve the local search performance of the algorithm and accelerate convergence speed. The global mutation operator is used to exploit a wider area and jump out of the local optima. An adaptive strategy for assigning mutation strategies and control parameters is proposed in this paper. The more successful is a mutation strategy and control parameter setting in the previous search, the more chance it will be used in the further search. Moreover, a novel crossover operator based subregion and external set strategy also is introduced. In order to demonstrate the performance of the proposed algorithm, it is compared with the MOEA/D-DE and the hybrid-NSGA-II-DE. The result indicates that the proposed algorithm is efficient.**

Index Terms— **differential evolution, multiobjective optimization, mutation strategy, adaptive strategy**

I. Introduction

In the past few decades, DE algorithm has been successfully used to solve various optimization problems [1], [2], such as electromagnetic design [3], network learning [4], filter design [5] and optimization of reservoir systems [6]. It attracts wide scholar's attentions. Since the first version of the DE algorithm emerged in [7], many variants have been developed [8], [11]. At the same time, the robustness and effectiveness of the algorithm has been verified in many application domains by comparison with the traditional evolutionary algorithms [1].

The DE operator uses the direction from one individual to other one to simulate the descent direction of the function. This will surely do much to search the better individual. However, the multiobjective optimization problem (MOP) differs from the single objective optimization problem. The Pareto optimal solutions of MOP are distributed in the decision space. If these candidate individuals used to generate the difference vectors are away from each other, the direction from one to the other may not simulate the descent direction of function. Thus, a subregions strategy is introduced in this paper. The objective space is divided into some subregions. For more details about subregion, refer to [12], [13].

Corresponding author: Hai-lin Liu.

Additionally, the direction from the worse individual to the better one is better suited to simulate the descent direction of the function, but the dominated individuals are eliminated at once in most of multiobjective evolutionary algorithms, including the current state-of-the-art algorithms [14], [15], [16]. It is not conducive to utilize the dominated individuals to construct the simulative descent direction. In order to overcome the drawbacks, an external set is employed for each subregion, which stores some individuals ever found before in this subregion. This idea differs from previous works in [17], [18], in which an archive population is used to store the best solutions found so far. It can not store the dominated individuals and utilize them to construct the difference vectors.

As is well known, the mutation strategy has a significant influence on the performance of differential evolution. Researchers have made great efforts in improving the mutation strategy and proposed some mutation strategies for solving single objective problems [17]. A local and global difference mutation strategies based subregions and external sets are proposed in this paper. For each individual, called target vector, in a subregion, a difference vector is constructed by using the idea of direct simplex method of mathematical programming [19]. The local difference vectors and global difference vectors are all constructed with three candidate individuals. The only difference is the selection of the candidate individuals. For the local difference vector, one of the candidate individuals is from the subregion which includes the target vector and the other two are from its corresponding external set. Such difference vector can largely plays a role in exploring the good individuals in this area. For the global difference vector, one of the candidate individuals is randomly selected from the subregions which not include the target vector and the others also are from the external set corresponding to target vector. The global difference vector is very useful for exploiting a wider area.

The performance of DE is sensitive to the mutation strategy and control parameter setting [20]. During the last decades, there are many empirical guidelines for choosing mutation strategies and control parameter settings are proposed [21]. A systematic framework for combining different mutation strategies with different control parameter settings is presented in [22]. CoDE proposed by Wang et al. improves the DE performance combining several trial vector generation strategies with several different

© 2013 ACADEMY PUBLISHER
doi:10.4304/jcp.8.2.294-301

control parameter settings [23]. These algorithms are for the single objective optimization problems. To the best of our knowledge, there is not attempt to solve MOP by combining different mutation strategies and control parameter settings in one framework.

Noting that the best mutation strategy and control parameter setting can be different for different MOP. Moreover, the different search regions of the same MOP and the different stages of DE also may require different mutation strategies and control parameter settings. It is very significant to assign mutation strategies and control parameter settings according to the characteristic of the problem adaptively. An adaptive multiobjective differential evolution algorithm for MOP, called AMODE, is proposed along this line in this paper. The more successful is a strategy and control parameter setting in the previous search, the more chance it will be used in the further search. Extensive experiments have been conducted to compare the proposed algorithm with MOEA/D-DE [24] and the hybrid NSGA-II-DE [18] on sixteen commonly used test instances. The result indicates that the algorithm is efficient.

The rest of this paper is organized as follows: Section II introduces the multiobjective differential evolution algorithm. Section III proposes the new algorithm. Section IV gives the experimental results. Section V concludes this paper.

II. Multiobjective Optimization Problems and Differential Evolution

A. Multiobjective Optimization Problem

A multiobjective optimization problem can be described in the following ways:

$$\begin{aligned} &minimize \quad G(x) = (g_1(x), g_2(x), ..., g_m(x))^T \\ &subject \quad to \quad x \in \prod_{i=1}^{n} [a_i, b_i] \end{aligned} \tag{1}$$

where $\prod_{i=1}^{n} [a_i, b_i]$ is the decision space, $G : R^n \rightarrow R^m$ contains m number real-value objective functions. R^m is objective space. Since these objectives are contradictory and conflicting, there is no solution to minimize all objectives simultaneously. Therefore, we need to employ a fitness assignment scheme [14], [24], [12] to choose better individuals. The min-mix strategy is used in this paper. Then the fitness function is defined as:

$$f(x|\omega) = \max_{1 \le i \le m} \{\omega_i (g_i(x) - \bar{g_i}^t)\} \tag{2}$$

where $\omega = (\omega_1, \cdots, \omega_m)$ is weight vector, $\bar{g_i}^t$ is the minimum of ith objective function found so far. Let $\omega^1, \cdots, \omega^N$ be a set of even spread weight vectors, we can define N fitness functions using these weight vectors [12].

B. DE Operator

The main idea of DE is to generate trial vectors (offsprings) by mutation operator and crossover operator.

1) Mutation: Let $X_i^{(t)} = (x_{i,1}^{(t)}, x_{i,2}^{(t)}, ..., x_{i,n}^{(t)})$ (called a target vector) be the ith individual of population in tth generation, $i = 1, \cdots, N$, where N is the size of the population. A mutation vector $V_i^{(t)}$ is generated by perturbation the base vector with several difference vectors. The five widely used DE mutation operators are

a. 'DE/rand/p'

$$V_i^{(t)} = X_{r_1}^{(t)} + F \times (\sum_{s=1}^{p} (X_{r_2,s}^{(t)} - X_{r_3,s}^{(t)})) \tag{3}$$

b. 'DE/current-to-rand/p'

$$V_i^{(t)} = X_i^{(t)} + F \times (\sum_{s=1}^{p} (X_{r_2,s}^{(t)} - X_{r_3,s}^{(t)})) \tag{4}$$

c. 'DE/best/p'

$$V_i^{(t)} = X_{best}^{(t)} + F \times (\sum_{s=1}^{p} (X_{r_2,s}^{(t)} - X_{r_3,s}^{(t)})) \tag{5}$$

d. 'DE/rand-to-best/p'

$$V_i^{(t)} = X_{r_1}^{(t)} + F \times (\sum_{s=1}^{p} (X_{best}^{(t)} - X_{r_3,s}^{(t)})) \tag{6}$$

e. 'DE/current-to-best/p'

$$V_i^{(t)} = X_i^{(t)} + F \times (\sum_{s=1}^{p} (X_{best}^{(t)} - X_{r_3,s}^{(t)})) \tag{7}$$

where p is the number of the difference vector, $X_{best}^{(t)}$ is best individual in the current population, and $X_{r_1}^{(t)}$, $X_{r_2,s}^{(t)}$, $X_{r_3,s}^{(t)}$, $s = 1, \cdots, p$ are individuals randomly selected from current population and also different from $X_i^{(t)}$. F is a real constant, called scaling factor, to control the step length of the difference vector.

As $p = 1$, DE mutation operator generates a difference vector to disturb the base vector; As $p = 2$, there are two difference vectors to disturb the base vector. Generally, more difference vectors are added to the base vector might lead to better perturbation.

Moreover, the scaling factor F has a great influence on the performance of the algorithm. Large values of this parameter can increase the population diversity and hence promote exploration, but if F becomes too large, the convergence speed will decrease very quickly. Small values of F can increase the convergence speed, but it must be above a certain value to avoid premature convergence. Thus, F should be chosen from $[0.4, 0.9]$.

In the first two strategies, the individual for mutation are selected from the population at random. They are effective for solving multimodal problems and maintain the population diversity. The other strategies utilize the information of the best individual found so far, and thus it can speed up the convergence.

© 2013 ACADEMY PUBLISHER

2) Crossover and Selection: The Trial vector $U_i^{(t)} = (u_{i,1}^{(t)}, ..., u_{i,j}^{(t)}, ...u_{i,n}^{(t)})$ is produced by the crossover between mutation vector $V_i^{(t)} = (v_{i,1}^{(t)}, ..., v_{i,j}^{(t)}...v_{i,n}^{(t)})$ and the target vector $X_i^{(t)} = (x_1^{(t)}, ..., x_j^{(t)}...x_n^{(t)})$. The trial vector is generated as follows:

$$u_{i,j}^{(t)} = \begin{cases} v_j^{(t)} \ if \ rand() > C_r \ or \ Z = j \\ x_{i,j}^{(t)} \ otherwise \end{cases} \tag{8}$$

$j = 1, \cdots, n$, where $C_r \in [0, 1]$ is the crossover probability, and Z is a random parameter index in $\{1, ..., n\}$.

The parameter C_r is sensitive to the performance of DE. It has been investigated in [20]. When the functions is separable, Cr lying in $(0, 0.2)$ is the best, while in $(0.9, 1)$ when the function's parameters are dependent or the function is multi-modal. Then C_r can be set in $[0.1, 0.9]$.

After crossover, the selection operator is performed to determine whether the target or the trial vector survives to the next generation. If the trial vector is better than the target vector, it replaces the corresponding target vector; otherwise the target is retained in the population.

III. Proposed Algorithm

A. Subregion Strategy and External Set

1) Subregion Strategy: The objective space is divided into S subregions as follow, where S is the number of the subregions. Firstly, the objective space is projected onto the first octant's unit hyper-sphere. S points $T^1, T^2, \cdots, T^S$ are uniformly distributed on the first octant's unit hyper-sphere. T^k serves as the central point of kth subregion. kth subregion is denoted as B_k, $k = 1, 2, \cdots, S$. We can design N weight vectors $\omega^1, \cdots, \omega^N$ evenly distributed on the first octant's unit hyper-sphere. And then assign they to these subregions by calculating the minimum distance from the weight vector to the central points. The upper limit size n_k of B_k is the number of weight vectors assigned to kth subregion. For individual $X_i^{(t)}$, its function vector is $G(X_i^{(t)})$. By calculating the Euclidean distances between the unit vector of $G(X_i^{(t)})$ and the central points T^k $(k = 1, 2, ...S)$, ascertain which central point is the nearest to the unit vector. Suppose that the unit vector is closest to T^j, the individual $X_i^{(t)}$ is classified into kth subregion. A more detailed description of subregion is specified in [13].

$5N$ individuals are uniformly and randomly generated in the initial step of the algorithm. And then they are divided into these subregions by above-mentioned manner. When the number of the individuals assigned to kth subregion is less than or equal to n_k, B_k is composed of all these individuals, otherwise we choose n_k different best individuals from all these ones by Equ. 2. In the algorithm generation t, we assign the offsprings into these subregions by above-mentioned manner, and then select not more than n_k different best individuals from individuals of the B_k and the offsprings assigned to this subregion by min-mix strategy.

2) External Set: In order to utilize the dominated individuals to construct the simulative descent direction, an external set is introduced for each subregion, which store some individuals ever found before in this subregion. The external set corresponding to kth subregion is denoted as Ω_k. In the initial step of algorithm, randomly select $5n_k$ individuals from initial $5N$ individuals to constitute the external set Ω_k, $k = 1, \cdots, S$. The updating method of Ω_k is as follows. If there is not any remaining individuals in updating subregion B_k, Ω_k need not to be updated; otherwise the remaining individuals randomly replace the same number of individuals in Ω_k.

B. Local and Global Mutation Operator Based Simplex

For each target vector $X_i^{(t)}$, suppose $X_i^{(t)} \in B_{k_i}$ and the corresponding external set is Ω_{k_i}, a global mutation operator and a local mutation operator are proposed based on subregion and external set strategy in this paper. The difference vector is created by using the idea of direct simplex method of mathematical programming. The local and global mutation vector are generated as follows:

The local mutation vector:

$$V_i^{(t)} = X_i^{(t)} + F\left(X_{l_i}^{(t)} - \frac{W_1^i + W_2^i}{2}\right) \tag{9}$$

where W_1^i and W_2^i are randomly selected from Ω_{k_i}, denoted as $W_1^i \lhd$, $W_2^i \lhd \Omega_{k_i}$, $X_{l_i}^{(t)} \lhd B_{k_i}$ and $X_{l_i}^{(t)} \neq X_i^{(t)}$.

The global mutation vector:

$$V_i^{(t)} = X_i^{(t)} + F\left(X_{g_i}^{(t)} - \frac{W_1^i + W_2^i}{2}\right) \tag{10}$$

where W_1^i and W_2^i as local mutation vector, that $W_1^i \lhd \Omega_{k_i}$ and $W_2^i \lhd \Omega_{k_i}$, $X_{g_i}^{(t)}$ and the target vector $X_i^{(t)}$ are from different subregions. That is $X_{g_i}^{(t)}$ is randomly selected from the subregions except B_{k_i}.

If jth component $v_{i,j}^{(t)}$ of $V_i^{(t)}$ is out of the boundary, it is reset as follows:

$$v_{i,j}^{(t)} = \begin{cases} \min\left\{b_j, 2a_j - v_{i,j}^{(t)}\right\} & \text{if } v_{i,j}^{(t)} < a_j \\ \max\left\{a_j, 2b_j - v_{i,j}^{(t)}\right\} & \text{if } v_{i,j}^{(t)} > b_j \end{cases} \tag{11}$$

Since Pareto optimal solutions in Equ.(1) can be regarded as the optimal solution of a single objective optimization problems with appropriate weight vector [16]. Therefore, we just need to explain the geometric significance of difference vector in the multimodal single objective optimization problem. It is easily promoted to the multiobjective optimization problem. As shown in Figure 1, it is the contour of peaks. There are two local minimum points O_2, O_3 and one global minimum point O_1. Individuals X_b, X_r, $X_{target} \lhd B_i$ but X_3. Individuals $W_k \lhd \Omega_i$, $k = 1, 2, 3, 4$. The implications of external set, local mutation operator and global mutation operator are illustrated in the following.

As descried in Figure 1, the direction from the worse individual W_1 to the better individual X_b is superior to the direction from the better individual X_a to the better individual X_b to simulates the descent direction

© 2013 ACADEMY PUBLISHER

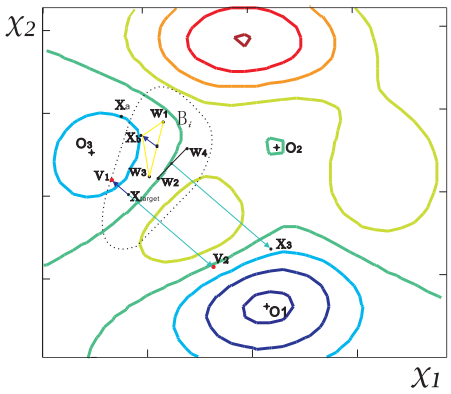


Figure 1. Illustration the global mutation operator and the local mutation operator

of the function. However, in most of the multiobjective evolutionary algorithm, the dominated individuals are eliminated at once. It is not conducive to utilize the dominated individuals to construct the simulative descent direction. As a result, an external set is introduced for each subregion in this paper. It is used to save some individuals ever found in this subregion before.

The individuals to take part in local mutation operation are from the same subregion and the corresponding external set. As shown in Figure 1, the difference vector is constructed with W_1, W_3 and X_b. The mutation vector $\mathbf{V}_1$ is generated by the difference vector and the target vector X_{target}. Obviously, It is favorite to approach the optimum O_3.

The global mutation operator exchanges the information among different subregions to discover the new region. As shown in Figure 1, W_2, $W_4 \lhd \Omega_i$ are in the external set corresponding to B_i and $X_{target} \lhd B_i$, but X_3 is not from the subregion B_i. The new individual V_2 is generated by the global mutation operator. It plays an important role in making the DE algorithm jump out of the local optimum.

C. Crossover Operator

For a tradeoff accelerating the convergence speed between maintaining diversity, a reasonable selection of crossover method is proposed in this paper. For mutation vector $V_i^{(t)}$ corresponding to target vector $X_i^{(t)}$, suppose $X_i^{(t)}$ is included in subregion B_{k_i} and the corresponding external set is Ω_{k_i}. Then, we perform the crossover between $V_i^{(t)}$ and another individual P_i which is selected as follows:

$$P_i = \begin{cases} P_i \lhd B_{k_i} & \text{if } rand > 0.3 \\ P_i \lhd \Omega_{k_i} & \text{otherwise} \end{cases} \tag{12}$$

where $rand$ is a random number in $[0, 1]$. Then, the new individual $U_i^{(t)} = (u_{i,1}^{(t)}, \cdots, u_{i,n}^{(t)})$ is generated as follows:

$$u_{i,j}^{(t)} = \begin{cases} v_{i,j}^{(t)} & \text{if } rand > C_r \text{ or } Z = j \\ p_{i,j} & \text{otherwise} \end{cases} \tag{13}$$

$j = 1, \cdots, n$, where $rand$ is a random number in $[0, 1]$, C_r is the crossover probability, Z is a random parameter index in $\{1, ..., n\}$, $v_{i,j}^{(t)}$ is the jth component of the mutation vector $V_i^{(t)}$, and $p_{i,j}$ is jth component of the individual P_i.

Generally, the individuals in the same subregion are close each other. The crossover between two adjacent individuals can largely play a role in exploring the good individuals in this area. Thus, we perform the crossover between the mutation vector $V_i^{(t)}$ and an individual $P_i \lhd B_{k_i}$.

However, the crossover operation only performing between the adjacent individuals is not very useful to exploit a wider area. The external set extends the search range of each subregion. Therefore, we perform the crossover between a mutation vector $V_i^{(t)}$ and an individual $P_i \lhd \Omega_{k_i}$.

N new individuals are generated after mutation and crossover. And then the subregions and external sets are updated by above-mentioned manner.

D. Adaptively Assigning Strategies and Control Parameters

Noting that the characteristics of the mutation strategies, different multiobjective optimization problems, even the different search region of the same multiobjective optimization problem might require different strategies. Moreover, the DE algorithm will do better in the different stages with different strategies. We also observed that in most DE variants including adaptive and self-adaptive DE variants, only one trial vector generation strategy is employed at each generation for each target vector. As a result, the search ability of these algorithms could be limited.

Meanwhile, there are two main control parameters of the DE algorithm: the mutation scaling factor F and the crossover probability C_r. Many attempts have also been made to improve the convergence speed and robustness of DE by tuning the control parameters [23] and the effect of each of these parameter on the performance has been studied in [20]. Clearly, the selection of the mutation strategies and control parameter depend on the characteristics of problems. However, there is no a priori knowledge about the characteristics of problems. Based on the above considerations, an adaptive assigning strategies and control parameters method is proposed in this paper.

A pool of distinct mutation strategies and a pool of values for the control parameters F and C_r are introduced in this paper. In order to verify the efficacy of the proposed mutation operators, we choose three mutation strategies to constitute the strategies candidate pool. The three selected mutation strategies are:

1) DE/current-to-rand/1 (Equ. 4)
2) Local mutation opterator (Equ. 9)
3) Global mutation operator (Equ. 10)

Note that the candidate individuals X_{r_2} and X_{r_3} in Equ. 4 are randomly selected from the external set corresponding to target vector $X_i^{(t)}$.

© 2013 ACADEMY PUBLISHER

The pool of C_r is chosen from 0.1 to 0.9 in step of 0.1 and F is chosen from 0.4 to 0.9 in step of 0.1.

Then the adaptive assigning strategies and control parameters method works as follow. Each member in the initial subregions is randomly assigned with a mutation strategy and control parameter from the pools. The trial vector is produced by the assigned mutation strategy and control parameter. The mutation strategy and control parameter of the offspring (trial vector) inherit parent's (target vector) the mutation strategy and control parameter with a given probability. Otherwise, the trial vector is randomly reinitialized with a new mutation strategy and control parameter from the pools. This leads to an increased probability to produce the offspring by the better combination of mutation strategy and control parameter in the future generations.

E. The Main Framework Of AMODE

The AMODE Algorithm

Input:
- The population size :N;
- The number of the subregions: S;
- The maximum number of evolution generations: Max_t;
- A uniform spread of N weight vectors: $\omega^1, \cdots, \omega^N$;
- A uniform spread of S center points: $T^1, T^2, \cdots, T^S$;
- The pool of mutation strategies;
- The pools of control parameter settings to each mutation strategy;

Output:
- The nondominant solutions;

Step 1. Initialization:

Step 1.1. Assign these weight vectors to the subregions, the upper limit size n_k of kth subregion is the number of the weight vectors assigned to this subregion and the corresponding external set is $4n_k$, $k = 1, \cdots, S$;

Step 1.2. Generates $5N$ initial individuals randomly and uniformly, initializes the subregions and the external sets.

Step 1.3. Each member in subregions is randomly assign with one of the mutation strategy from the pool and the associated control parameter setting is chosen from the corresponding pool;

Step 2. Mutation:

For i=1:N do

Randomly select individual X_i from the subregions, generate a mutation vector V_i corresponding to the target vector X_i using the mutation strategy and parameters associated with the target vector;

Step 3. Crossover:

Generate a trial vector (offspring) U_i for each target vector X_i by Equs 12 and 13. The mutation strategy and parameters associated with U_i is update as follow:

If $rand < 0.1$

Randomly select a mutation strategy and parameters from the pool;

Else

Which inherits the mutation strategy and parameters from the target vector X_i;

Step 4. Updating:

Step 4.1. Assigns N new individuals to each subregion by using the above-mentioned manner;

Step 4.2. Updates the subregions and external sets;

Step 5. Stopping Criteria: If stopping criteria is satisfied, then stop and output all the nondominant solutions. Otherwise, go to Step 2.

IV. Experimental Results

We use five bi-objectives ZDT (ZDT1, ZDT2, ZDT3, ZDT4, ZDT6) instances and two 3-objectives DTLZ (DTLZ1, DTLZ2) test instances [16] in comparing AMODE, MOED/D-DE and hybrid NSGA-II-DE [18]. The parameters are set as follows: population size N is 100 for bi-objectives test instances and 300 for 3-objectives test instances. The number of the subregions is set to $S = \sqrt{N}$. All the algorithms stop after 250 generations.

The other nine test instances F1-F9 [24] are used in comparing AMODE and MOED/D-DE. The parameters are set the same as ones used in [24]. N is set to 300 for bi-objective and 600 for 3-objectives test instances. For instances F1-F9, both algorithms stop after 150000 function evaluations, that is the maximum generations is $round(\frac{150000}{N})$.

The inverted generational distance (IGD) [16] is used to evaluate the performance of the algorithms in our experimental studies. Let Q be a set of uniformly distributed points in the objective space along the PF. Let Q^* be a set of searched nondominated solutions in objective space by algorithm. So the IGD from Q to Q^* is defined as follows:

$$IGD(Q, Q^*) = \frac{\sum_{v \in Q} d(v, Q^*)}{K} \tag{14}$$

where $K = |Q|$ is the size of set Q. The $d(v, Q^*)$ is the minimum Euclidean distance between v and the points in Q^*. K is 500 for bi-objectives and 1000 for 3-objectives test instances.

Table I shows the minimum and standard deviation of the IGD values of AMODE, hybrid-NSGA-II-DE, MOEA/D-DE for ZDT and DTLZ test instances in 20 independently runs. This table reveals that in terms of IGD, the final solutions obtained by AMODE are better than hybrid-NSGA-II-DE and MOEA/D-DE for all test instances except ZDT3. The poor performance of AMODE on ZDT3 could be attributed to the fact that the objectives in ZDT3 are disparately scaled.

© 2013 ACADEMY PUBLISHER

TABLE I.
THE IGD VALUES OF AMODE, HYBRID NSGA-II-DE AND MOEA/D-DE IN 20 INDEPENDENT RUNS FOR EACH TEST INSTANCE

IGD	**AMODE**		**NSGA-II-DE**		**MOEA/D-DE**	
Instance	best	standard	best	standard	best	standard
ZDT1	0.0039	3.5263e-8	0.0046	0.0012	0.0055	0.0039
ZDT2	0.0036	1.7037e-7	0.0048	0.0002	0.0065	0.0109
ZDT3	0.0086	5.9624e-6	0.0075	0.0054	0.0143	0.0091
ZDT4	0.0044	0.0035	0.124	0.0102	0.0076	0.0023
ZDT6	0.0044	0.0024	0.0147	0.0063	0.042	0.0003
DTLZ1	0.0230	3.5024e-6	0.0438	0.0515	0.0317	0.0005
DTLZ2	0.0292	4.9085e-6	0.0391	0.0003	0.0389	0.0001

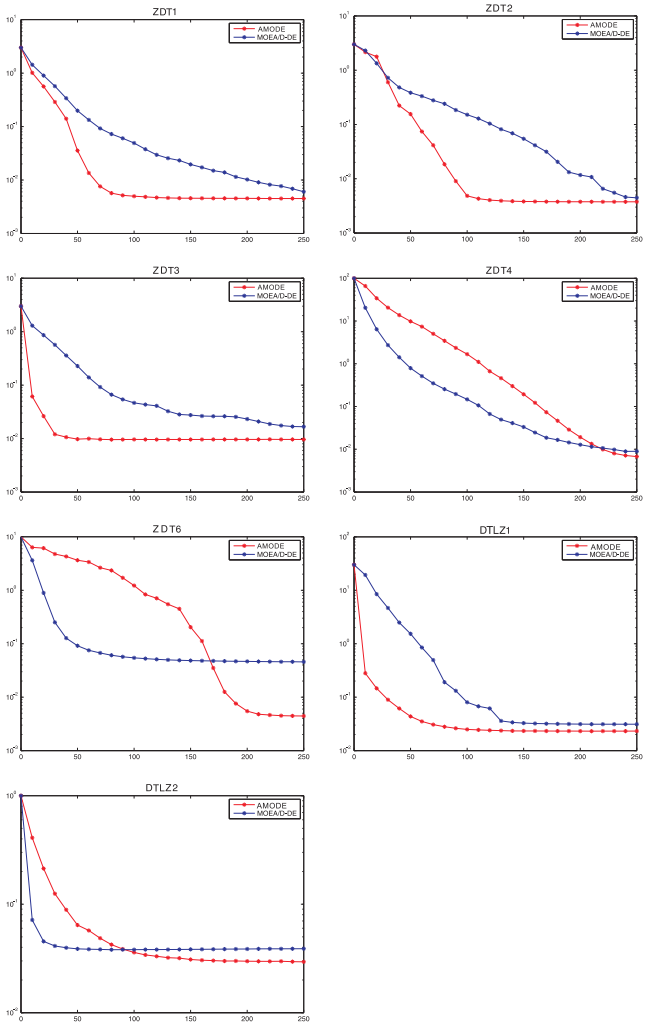


Figure 2. The evolution of average IGD-value in both AMODE and MOEA/D-DE in 20 independent runs.

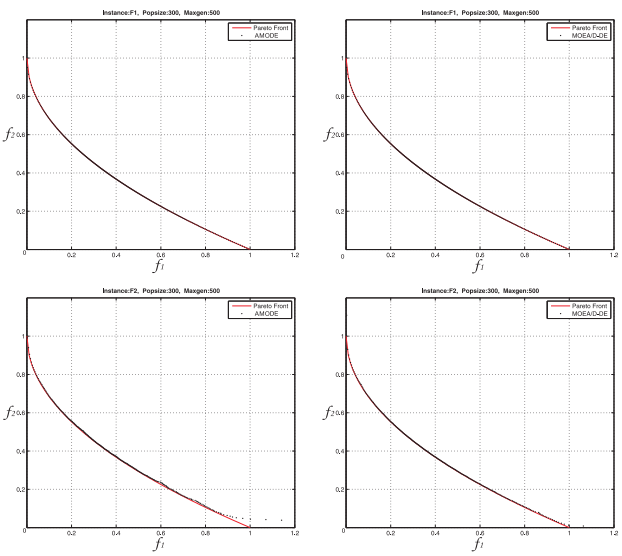
Figure 2 shows the evolution of the average IGD of the current population to with the number of function evaluations in AMODE and MOEA/D-DE for each test instance. As shown in Figure 2, IGD on average 100 generations is needed for instances ZDT1, ZDT2, ZDT3, DTLZ1 and DTLZ2 to find the approximate PF. But the number of generations is 200 for instances ZDT4 and ZDT6. These results indicate that AMODE converges faster than MOEA/D-DE for ZDT1, ZDT2, ZDT3 and DTLZ1, and slower than MOEA/D-DE for the other three test instances.

TABLE II.
THE IGD VALUES OF AMODE AND MOEA/D-DE IN 20 INDEPENDENT RUNS FOR F1-F9 TEST INSTANCES

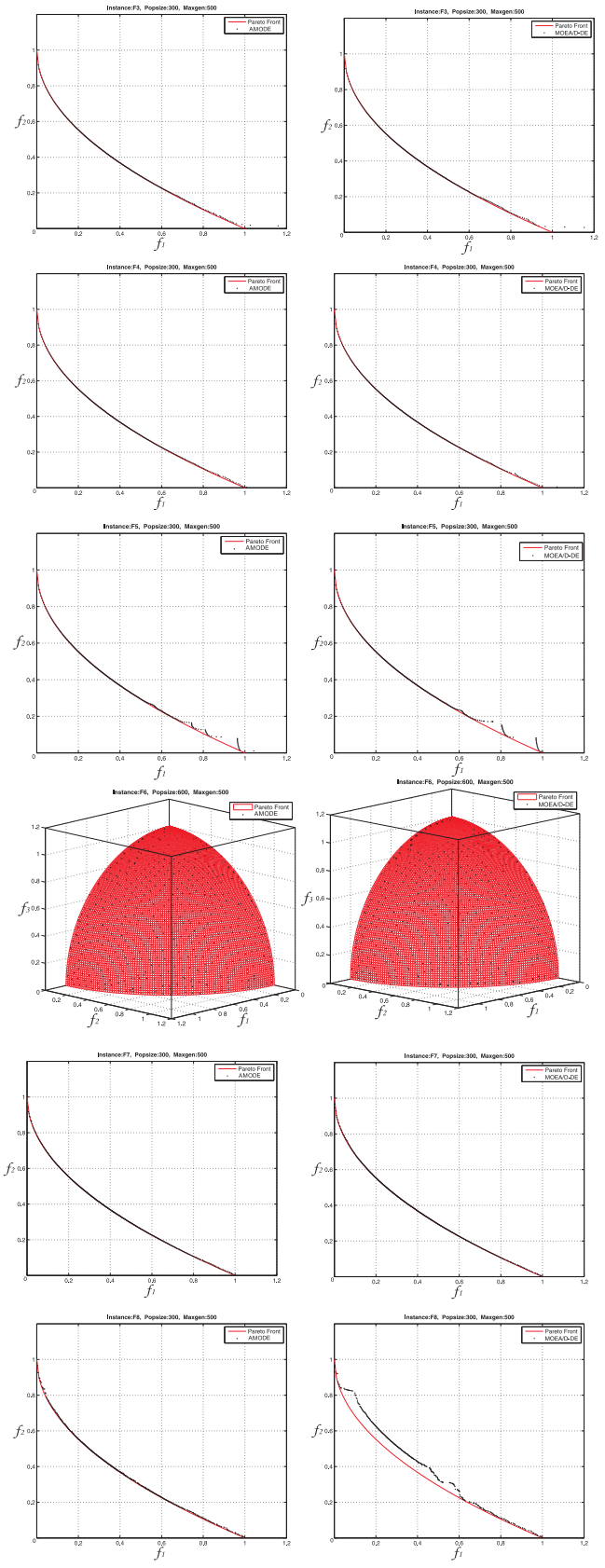
IGD	**AMODE**			**MOEA/D-DE**		
Instance	best	standard	mean	best	standard	mean
F1	0.0015	6.2442e-7	0.0015	0.0015	0	0.0015
F2	0.0059	3.2590e-3	0.0095	0.0023	0.0004	0.0028
F3	0.0031	2.1234e-4	0.0062	0.0022	0.0099	0.0068
F4	0.0024	9.0048e-5	0.0030	0.0025	0.0014	0.0040
F5	0.0065	3.5092e-4	0.0090	0.0073	0.0069	0.0127
F6	0.0233	2.2071e-5	0.0255	0.0276	0.0014	0.0289
F7	0.0015	7.3930e-6	0.0017	0.0015	0.0063	0.0049
F8	0.0023	2.7365e-5	0.00252	0.0487	0.0429	0.0998
F9	0.0064	5.2063e-3	0.0141	0.0025	0.0008	0.0035

Table II shows the minimum and standard deviation and mean of IGD values of AMODE and MOEA/D-DE for F1-F9 test instances in 20 independent runs. For the minimum IGD of F1-F9 test instances, AMODE is better than MOEA/D-DE for F4, F5, F6 and F8. It is about the same as MOEA/D-DE for F1, and F7, and worse than MOEA/D-DE for F2, F3 and F9. It is found that AMODE has comparable performance with MOEA/D-DE for F1-F9.

Figure 3 shows in the objective space, the distribution of the final solutions obtained in the runs with the lowest IGD-value of AMODE and MOEA/D-DE for F1-F9. It is clear that AMODE is better than MOEA/D-DE for F8 and slightly worse than MOEA/D-DE for F9 by Figure 3. It is about the same as MOEA/D-DE for others.



© 2013 ACADEMY PUBLISHER



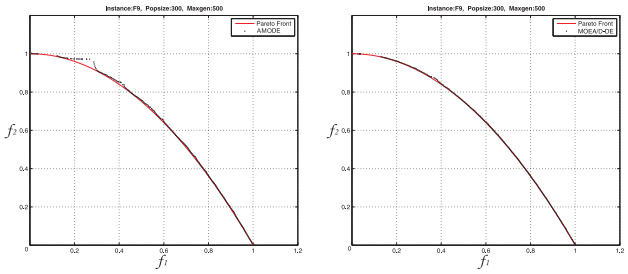


Figure 3. Plot of the distribution of the final solutions obtained in 20 independently runs with the lowest IGD-value of AMODE and MOEA/D-DE for F1-F9. The left panel is for the AMODE and the right panel for MOEA/D-DE

V. Conclusion

An adaptive multiobjective differential evolution (AMODE) was proposed based the subregion and external set strategies. The main contribution of AMODE can be summarized as follows: 1) proposing a local and a global mutation operator based the subregion and external set, which use the idea of direct simplex method to make they more efficient; 2) combining different mutation strategies and control parameter settings in one framework and adaptively selecting the best mutation strategy and control parameter according to the characteristic of the problems for solving multiobjective optimization problems; 3) presenting a novel crossover operator based subregion and external set strategy to accelerate the convergence speed and maintain the population diversity. We have compared AMODE with hybrid-NSGA-II-DE, MOEA/D-DE on continuous multiobjective optimization problems. The result indicated that the algorithm is efficient.

Acknowledgment

This work was supported in part by the Natural Science Foundation of China (60974077), and in part by the Natural Science Foundation of Guangdong Province (S2011030002886, S2012010008813), and in part by Guangzhou projects of science and technology (11C42110781), and in part by Innovation Programmes of department of education of Guangdong province (2012KJCX0042), and in part by Zhongshan projects of science and technology (20114A223).

References

[1] K. V. Price, R. M. Storn, and J. A. Lampinen, *"Differential evolution: A practical approach to global optimization,"* 1st ed., ser. Natural Computing Series, Springer, December 2005.

[2] U. K. Chakraborty, *"Advances in Differential Evolution,ser. Studies in Computational Intelligence,"* Berlin: Springer-Verlag, vol. 143, 2008.

[3] H.-K.Kim, J.-K. C. K.-Y. Park, and D. Lowther, "Differential evolution strategy for constrained global optimization and application to practical engineering problems," *IEEE Transactions on Magnetics,* vol. 43, no. 4, pp. 1565-1568, April 2007.

[4] J. Ilonen, J.-K. Kamarainen, and J. Lampinen, "Differential evolution training algorithm for feed-forward neural networks," *Neural Processing Letters*, vol. 7, no. 1, pp. 93-105, 2003.

© 2013 ACADEMY PUBLISHER

[5] S. Ch. Pei, C. C. Tseng, "A comb filter design using fractional-sample delay," *IEEE Transactions on Circuits and Systems II*, vol. 45, no. 5, pp. 649-653, 1998.
[6] M. Reddy and D. Kumar,"Multiobjective differential evolution with application to reservoir system optimization," *Journal of Computing in Civil Engineering,* vol. 21, no. 2, pp. 136-146, March-April 2007.
[7] K. V. Price and R. M. Storn, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, December 1997.
[8] P. Lei, Y. Wang, G. Dai and Z. Cao, "A novel differential evolution with uniform design for continuous global optimization," *Journal of Computers*, vol. 7, no. 1, pp. 3-11, 2012.
[9] Z. Huang, C. Wang and M. Ma, "A robust archived differential evolution algorithm for global optimization problems," *Journal of Computers*, vol. 4, no. 2, pp. 160-167, 2009.
[10] J. Yan, C. Guo, W. Gong, "Hybird differential evolution with convex mutation," *Journal of Software*, vol. 6, no. 11, pp. 2321-2328, 2011.
[11] N. K. Madavan, "Multiobjective optimization using a pareto differential evolution approach," *in Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1145-1150, May 2002.
[12] H. Liu, W. Chen, F. Gu "A novel multiobjective differential evolutionary algorithm based on subregion search," *in Proceeding Of IEEE Congress on Evolutionary Computation*, 2012.
[13] H. Liu, F. Gu, "A improved NSGA-II algorithm based on sub-regional search" *in Proceeding of IEEE World Congress on Computational Intelligence*, pp. 1906-1911, 2011.
[14] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitistmultiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182-197, 2002.
[15] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving thestrength pareto evolutionary algorithm for multiobjective optimization," *in Proceeding of Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems Conference*, pp. 95-100, 2001.
[16] Q. Zhang, H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712-731, 2007.
[17] Z. Yang, J. He, and X. Yao, "Making a difference to differential evolution," *in Advance in Metaheuristics for Hard Optimization*, pp. 397-414, 2008.
[18] L. S. Batista, F. G. Guimaraes, J. A. Ramirez, "A Differential Mutation operator for the archive population of multiobjective evolutionary algorithms," *in Proceeding Of IEEE Congress on Evolutionary Computation*, pp. 1108-1115, 2009.
[19] Martinez, S.Z., Montano, A.A., Coello Coello, C.A., "A nonlinear simplex search approach for multi-objective optimization," *in Proceeding Of IEEE Congress on Digital Object Identifier*, pp. 2367-2374, 2011.
[20] L. Das, P. N. Suganthan, "Differential Evolution: A Survey of the State-of-the-Art", *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4-31, 2011.
[21] K. Qin, L. Huang and N. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization", *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398-417, 2009.
[22] R. Mallipeddi, P. N. Suganthan, Q. K. Panb and M. F. Tasgetirenc, "Differential evolution algorithm with ensemble of parameters and mutation strategies", *Applied Soft Computing*, vol. 11, no. 2, pp. 1679-1696, 2011.
[23] Y. Wang, Z. Cai and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters", *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55-66, 2011.
[24] H. Li, Q. Zhang, "Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol.13 , no. 2, pp. 284-302, 2009.

Fangqing Gu received the BS degree in mathematics from Changchun university, Changchun, China, and MS degree in applied mathematics from Guangdong University of Technology, Guangzhou, China. He is a Ph.D candidate of Hong Kong Baptist University, Hong Kong SAR, China. His current research interests include evolutionary computation and optimization, data mining.

Hai-lin Liu received the BS degree in mathematics from Henan Normal University, Xinxiang, China, the MS degree in applied mathematics from Xidian University, Xi' an, China, the Ph.D degree in control theory and engineering from South China University of Technology, Guangzhou, China, and Post-doctor in the Institute of Electronic and Information, South China University of Technology, Guangzhou, China. He is currently a Professor of faculty of applied mathematics at the Guangdong University of Technology. His research interests include evolutionary computation and optimization, blind source separation.

© 2013 ACADEMY PUBLISHER