

Research Article

An Enhanced Differential Evolution Based Algorithm with Simulated Annealing for Solving Multiobjective Optimization Problems

Bili Chen,¹ Wenhua Zeng,¹ Yangbin Lin,² and Qi Zhong²

¹ School of Software, Xiamen University, Xiamen 361005, China

² School of Information Science and Engineering, Xiamen University, Xiamen 361005, China

Correspondence should be addressed to Wenhua Zeng; whzeng@xmu.edu.cn

Received 1 December 2013; Revised 27 March 2014; Accepted 7 April 2014; Published 7 May 2014

Academic Editor: Frank Werner

Copyright © 2014 Bili Chen et al. 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.

An enhanced differential evolution based algorithm, named multi-objective differential evolution with simulated annealing algorithm (MODESA), is presented for solving multiobjective optimization problems (MOPs). The proposed algorithm utilizes the advantage of simulated annealing for guiding the algorithm to explore more regions of the search space for a better convergence to the true Pareto-optimal front. In the proposed simulated annealing approach, a new acceptance probability computation function based on domination is proposed and some potential solutions are assigned a life cycle to have a priority to be selected entering the next generation. Moreover, it incorporates an efficient diversity maintenance approach, which is used to prune the obtained nondominated solutions for a good distributed Pareto front. The feasibility of the proposed algorithm is investigated on a set of five biobjective and two triobjective optimization problems and the results are compared with three other algorithms. The experimental results illustrate the effectiveness of the proposed algorithm.

1. Introduction

Many real world problems are MOPs and it has prompted a wide research boom about the MOPs over the past few decades. A lot of multiobjective evolutionary algorithms (MOEAs) have been suggested such as these famous algorithms in [1–4]. For comprehensive overviews, the reader can refer to [5–7].

In the present study, we emphasize on the differential evolution (DE) for solving MOPs. DE, one of the most popular evolutionary algorithms, was initially presented by Storn and Price [8, 9]. DE has a simple principle and is easy to be implemented. DE was originally used for solving single objective optimization problems in [10]. In 2001, Abbass et al. [11] presented a Pareto-frontier differential evolution algorithm (PDE) for solving MOPs. In 2002, Abbass [12] further improved the PDE with self-adaptive crossover and mutation operator. In the same year, the nondominated sorting and the concept of ranking used in NSGA-II are incorporated in the DE to solve MOPs in [13]. Xue et al. [14]

proposed an algorithm called Pareto based multiobjective differential evolution (MODE), which incorporates the concept of Pareto optimal into the mutation operator. Parsopoulos et al. [15] presented a vector evaluated differential evolution (VEDE) for solving MOPs. A domination selection operator is adopted to improve the performance of the algorithm. Kukkonen and Lampinen [16, 17] proposed a generalized differential evolution (GDE) and an improved version of GDE for solving MOPs with constraints. In 2006, Hernández-Díaz et al. [18] combined the DE and the rough set theory for MOPs. Wang et al. [19] proposed a self-adapted differential evolution algorithm, which adopts an external elitist and a crowding entropy diversity measure tactic. Huang et al. [20, 21] improved their self-adaptive differential evolution (SaDE) to solve MOPs by a multiobjective adaptive differential evolution (MOSaDE). Moreover, they further extended MOSaDE by objective-wise learning strategies and called it as WO-MOSaDE. Ali et al. [22] proposed a modified differential evolution (MDE) for solving MOPs. In 2012, they [23] further continued their research by giving an enhanced

version of MDE, named multiobjective differential evolution algorithm (MODEA), which incorporates the opposition-based learning [24] and the concept of random localization in mutation.

Besides the DE method, in this paper, the concept of simulated annealing is utilized for controlling the acceptance of the candidate solutions. Simulated annealing, proposed by Kirkpatrick et al. [25] is a popular probabilistic metaheuristic for solving the optimization problems. Simulated annealing is inspired by the annealing in metallurgy, a technique of controlling the temperature of a material from a high temperature to low one in a probability for increasing the volume of its crystals and reducing their defects. The simulated annealing algorithm works for finding minimal cost solutions by minimizing the associated energy function. In the process, the temperature is first in a high level (T_{max}) and then decreases gradually, if the cooling is sufficiently slow, the global minimum will be reached. The viewpoint has been proven by S. Geman and D. Geman in [26]. There are some attempts of adopting simulated annealing to solve MOPs. Especially, there have been a few studies that utilize the concept of Pareto-dominance in the multiobjective simulated annealing, the reader can refer to [27] for a review about this research. In [28], an archived multiobjective simulated annealing algorithm named AMOSA is proposed. The algorithm incorporates the concept of archive and adopts a domination status to determine the acceptance of a new solution.

The rest of the paper is organized as follows. In Section 2, the problem formulation is introduced. The proposed algorithm is presented in Section 3. In Section 4, some numerical examples are given to show the performance of the proposed algorithm with the other algorithms, while the conclusion is reached in Section 5.

2. Problem Formulation

We assume that the MOPs given in this paper are minimization problems, while each maximization problem can be transformed into a minimization problem. A MOP can be formally described as follows:

$$\min F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \quad (1)$$

$$\text{s.t. } g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p \quad (2)$$

$$h_i(\mathbf{x}) = 0, \quad i = 1, \dots, q, \quad (3)$$

where m is the number of objective functions and $f_i(\mathbf{x})$ is the i th objective function to be optimized. $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ are the set of inequality and equality constraints, respectively. p is the number of inequality constraints and q is the number of equality constraints. $\mathbf{x}(x_1, x_2, \dots, x_n)$ is the vector of variables which is called a solution for a MOP, where n is the number of variables. That is to say, a MOP is to find all $\mathbf{x}(x_1, x_2, \dots, x_n)$ which satisfies (2) and (3).

Suppose that there are two solutions $\mathbf{x}(x_1, x_2, \dots, x_n)$ and $\mathbf{y}(y_1, y_2, \dots, y_n)$, \mathbf{x} is said to dominate \mathbf{y} if $\forall i \in \{1, 2, \dots, m\}$, $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists i \in \{1, 2, \dots, m\}$, $f_i(\mathbf{x}) < f_i(\mathbf{y})$. \mathbf{x} dominates \mathbf{y} which can be denoted as $\mathbf{x} < \mathbf{y}$. A solution

\mathbf{x} is called a Pareto-optimal solution if there is not another solution \mathbf{x}' satisfying $\mathbf{x}' < \mathbf{x}$. A set containing all the Pareto-optimal solutions is called the Pareto-optimal set (PS). A set containing all the Pareto-optimal objective vectors is called the Pareto front (PF) such that $PF = \{F(\mathbf{x}) \mid \mathbf{x} \in PS\}$ [29].

3. The Proposed Algorithm

The proposed algorithm is an extension of MODEA algorithm proposed by Ali et al. [23]. In the proposed MODESA, we focus on improving the convergence of the obtained solution towards the Pareto-optimal front, and the diversity and distribution of the solutions. Besides using the simulated annealing process to control the acceptance of the candidate solutions, an efficient diversity maintenance mechanism [30] is also adopted to improve the distribution of these obtained solutions.

3.1. Population Initialization. In MODESA, initial population is generated in the same way as that of MODEA [23]. Because of the lack of a prior information about the solution, its corresponding opposite estimate is considered for getting a better solution. Here, the concept of opposite solution is given. Suppose that the current solution is $\mathbf{x}(x_1, x_2, \dots, x_i, \dots, x_n)$ and the lower and upper bounds of x_i are $v_{i,\min}$ and $v_{i,\max}$, respectively. Then the opposite solution $\mathbf{x}'(x'_1, x'_2, \dots, x'_i, \dots, x'_n)$ of \mathbf{x} is calculated as $x'_i = v_{i,\min} + v_{i,\max} - x_i$ [23].

The advantage of doing this is that we can choose the better one between the solution and the opposite one as the initial solution. The procedure of initialization in the paper is given as follows. A population with N solutions is generated using uniform random distribution and a population containing their opposite solutions is constructed too. Next, a combined population with the population and the opposite population is constructed and the best N solutions of the combined population are selected by using the nondominated sorting in NSGA-II [2] and the diversity maintenance mechanism based on vicinity distance [30].

3.2. Differential Evolution (DE). Differential Evolution (DE) presented by Storn and Price [8, 9] is an easy and efficient evolutionary algorithm. DE maintains a population of N solutions and contains three main operators: mutation, crossover, and selection.

The mutation operation is defined as follows:

$$\mathbf{z} = \mathbf{x}_{r1} + f * (\mathbf{x}_{r2} - \mathbf{x}_{r3}), \quad (4)$$

where \mathbf{x}_{r1} , \mathbf{x}_{r2} , and \mathbf{x}_{r3} are randomly selected from the current population and $r1 \neq r2 \neq r3 \neq i$. f is a control parameter which is set in $(0, 2]$ in Storn and Price [8, 9]. For the sake of convenient description, the solution which is selected to plus a perturbed value is called as the base vector. \mathbf{x}_{r1} (the base vector) is perturbed by adding to it the product of the control parameter and the difference of \mathbf{x}_{r2} between \mathbf{x}_{r3} . $\mathbf{z}(z_1, z_2, \dots, z_n)$ is the generated perturbed solution. It is important to note that if z_i is out of bounds, it must be repaired. As mentioned above, the lower and upper bounds of

variable z_i are $v_{i,\min}$ and $v_{i,\max}$, respectively. In this study, if z_i is less than $v_{i,\min}$, z_i is set to $v_{i,\min}$; if z_i exceeds $v_{i,\max}$, z_i is set to $v_{i,\max}$. Ali et al. [23] have proved that using the one having the best fitness value in these three randomly selected individuals as the base vector has two advantages. The first advantage is that it is neither merely random nor merely greedy. The other one is that it can offer a localized effect to help in finding more different regions of the search space around the underlying solutions [23]. Therefore, in MODESA, after choosing the three individuals, we select the best one of them to be the base vector.

After the perturbed solution is generated, a trial solution \mathbf{w} is produced by the crossover operation which can be defined as follows. For each variable j ,

$$w_j = \begin{cases} z_j, & \text{rand}_j \leq \text{cr} \vee j = jj \\ x_j, & \text{otherwise,} \end{cases} \quad (5)$$

where $jj \in \{1, \dots, n\}$ is a random number and $\text{cr} \in [0, 1]$ is the crossover rate. In general DE, the selection operator is utilized to decide whether the trial solution \mathbf{w} replaces the target solution \mathbf{x} . The selection operation can be defined according to the following:

$$\mathbf{x} = \begin{cases} \mathbf{w}, & \mathbf{w} < \mathbf{x} \\ \mathbf{x}, & \text{otherwise.} \end{cases} \quad (6)$$

In the proposed MODESA, the target individual \mathbf{x} is not simply replaced by the trial solution \mathbf{w} . A simulated annealing procedure introduced in Section 3.3 is employed to determine the acceptance of the trial solution.

3.3. Simulated Annealing Procedure. After the mutation phase, the crossover operator is performed, and the algorithm enters the selection phase. In MODESA, we incorporate the principle of simulated annealing into the selection phase. Before introducing the proposed selection phase, we first give a concept of domination in simulated annealing.

3.3.1. The Concept of Domination. To calculate the acceptance probability of a new solution, the amount of domination used in AMOSA [28] is adopted here. Given two solutions \mathbf{x} and \mathbf{y} , the amount of domination is calculated by the following equation:

$$\Delta\text{dom}_{a,b} = \prod_{i=1, f_i(\mathbf{x}) \neq f_i(\mathbf{y})}^m \left(\frac{|f_i(\mathbf{x}) - f_i(\mathbf{y})|}{R_i} \right), \quad (7)$$

where m is the number of objectives and R_i is the range of the i th objective. When R_i is unknown, the solutions in the current population are used for calculating it.

3.3.2. Simulated Annealing Used in the Selection Operator of the Proposed MODESA. As (6), if the trial solution dominates the current solution, the general DE simply replaces the current solution by the trial solution. However, doing this may make the algorithm over converge and easy to trap into local optimal. As is known, there are three possibilities when

comparing two solutions, the current solution \mathbf{x} and the new solution \mathbf{y} , as follows:

- (1) \mathbf{x} dominates \mathbf{y} ;
- (2) \mathbf{x} is dominated by \mathbf{y} ;
- (3) \mathbf{x} and \mathbf{y} are incomparable.

If \mathbf{x} dominates \mathbf{y} , it is obvious that \mathbf{y} is not better than \mathbf{x} , so that we can simply replace \mathbf{y} by \mathbf{x} . If \mathbf{x} is dominated by \mathbf{y} , we accept the new solution \mathbf{y} distinctly. If \mathbf{x} and \mathbf{y} are incomparable, the solution \mathbf{y} may have the potential to guide the solutions towards an unreached area of the search space. Therefore, in the DE process, we use the simulated annealing to accept \mathbf{y} under a certain probability. Meanwhile, we need to control the acceptance probability, and the reason is that if we do not control the acceptance probability, all these potential solutions which are not absolutely good will enter to the next generation which will confuse the algorithm and make it unable to have a right direction towards the Pareto-optimal front. That is to say, we must decrease the acceptance probability. In simulated annealing, with the decreasing of the temperature, the acceptance probability is decreased too, which will avoid the algorithm producing too much potential solutions in the end. It is worth mentioning that if we allow some potential solutions existing during the procedure, the next nondominated sorting procedure may again remove these potential solutions, which will make the purpose of allowing some potential solutions to search more areas of the searching space fail. Therefore, in the proposed MODESA, each individual is assigned a prior life cycle. The number of prior life cycles means how many times the individual can be selected preferentially in the nondominated sorting. For the sake of convenience, the life cycle value of solution x_i is denoted as $life_i$. Once the individual is selected preferentially in the nondominated sorting, the prior life cycle of the individual is minus one when it is greater than zero. Note that, in the population initialization, the prior life cycle of each individual is set to zero.

In this paper, given a temperature temp , the new solution is selected with a probability prob as follows:

$$\text{prob} = \exp\left(-\frac{\Delta\text{dom}_{\text{avg}}}{\text{temp}}\right), \quad (8)$$

$$\Delta\text{dom}_{\text{avg}} = \frac{\sum_{i=1}^k \Delta\text{dom}_{i,\text{new-solution}}}{k}, \quad (9)$$

where k is the number of solutions that dominates the new solution. It is worth mentioning that $\Delta\text{dom}_{\text{avg}}$ represents the average amount of domination of the new solution and the k solutions that dominate it. Suppose that $\Delta\text{dom}_{\text{avg}}$ is constant, the acceptance probability decreases when the temperature decreases. The simulated annealing used in the selection operator of MODESA is as shown in Algorithm 1.

3.4. Diversity Maintenance Mechanism. The diversity maintenance mechanism adopted in the paper is presented by Kukkonen and Deb [30] for pruning of nondominated solutions. Instead of using the crowding distance in Deb

```

input:  $x_i, w_i, temperature, T_{min}, T_{max}, P_t, P_{temp}, max\_life\_cycle$ 
If ( $temperature > T_{min}$ ) {
  Compute the value of  $\Delta dom_{avg}$  by (9).
  Calculate the acceptance probability prob:
    prob =  $\exp(-\Delta dom_{avg}/temperature)$ 
  With a probability prob Do
    Put the current individual  $x_i$  to the temporary population:  $P_{temp} = P_{temp} \cup x_i$ .
    Replace the current individual  $x_i$  by the trial individual  $w_i$ :  $x_i = w_i$ .
    Assign the life value  $max\_life\_cycle$  to the new individual  $x_i$ :  $life_i = max\_life\_cycle$ ,
  where  $life_i$  represents the life cycle value of solution  $x_i$ .
  otherwise
    Put the current individual  $x_i$  to the temporary population:  $P_{temp} = P_{temp} \cup x_i$ .
} // End If

```

ALGORITHM 1: The proposed simulate annealing algorithm.

et al. [2], the method proposed a new crowding estimation called vicinity distance using nearest neighbors. In this paper, we call this method as the vicinity distance based pruning method for convenience. The basic process of the method is to remove the most crowded solutions in the nondominated solution one by one. Moreover, after each removal, the vicinity distance values of the remaining solutions must be updated again. The vicinity distance is used to get an estimate of the density of solutions surrounding a particular solution, it calculates the product of k nearest neighbors of the particular solution. The solutions with the smallest value of vicinity distance are considered to be the most crowded ones. In [30], the value of k is assigned to the number of objectives; that is, $k = m$. The reader can refer to [30] to get more information about this method.

3.5. The Framework of the Proposed MODESA Algorithm

3.5.1. The Framework of the Proposed MODESA Algorithm. As mentioned before, the proposed algorithm first uses the population initialization phase to generate a population with N solutions. After the nondominated sorting, each solution has a ranking value. Next, the main loop begins. We introduce the t th generation of MODESA in the following.

At the t th generation, an empty temporary population is created first. The temporary population is going to store the new good solutions generated by the DE in the loop. Then for each individual in the current population, the DE operators are used to generate a trial individual. Next, there are three possibilities when comparing the current individual and the trial individual. If the trial individual dominates the current individual, we simply replace the current solution by the trial solution. If the trial individual and the current solution are incomparable, the proposed simulated annealing process is adopted to determine whether the trial individual is accepted. If the trial individual is dominated by the current solution, we also put the trial individual into the temporary population as MODEA does. After all the individuals are handled, the current population and the temporary population are combined as a union population. Note that, if the current generation is the last generation, the members that are

dominated by any other individuals must be removed in the union population. The reason of doing this is that, because we allow some potential solutions which are not absolutely good and have a positive life cycle to exist in the population; therefore, at the end of the generation, these solutions must be removed or they will influence the effectiveness of the results. Finally, the nondominated sorting and the vicinity distance based pruning approach are used for getting the best N solutions.

In this paper, the parameters that need to be set a priori are listed in Table 1 and the Pseudocode of MODESA is shown in Pseudocode 1.

3.5.2. The Computational Complexity Analysis. The proposed MODESA approach is a simple algorithm to be implemented. In the following, the overall computational complexity of MODESA is analyzed. Basic operations and their worst case complexities are given as follows.

- (1) The initialization phase is selecting N solutions out of $2N$ solutions by using nondominated sorting and the vicinity distance base pruning method: $O(m \times 4N^2) + O(m^2 \times 2N \log(2N))$.
- (2) The procedure of mutation is $O(m \times 2N)$.
- (3) The procedure of checking the domination status of the trial solution and the current solution is $O(m \times N)$, and the procedure of simulated annealing is $O((m + m \times N) \times N)$.
- (4) The procedure of selecting N solutions out of $2N$ solutions by using nondominated sorting and the vicinity distance base pruning method: $O(m \times 4N^2) + O(m^2 \times 2N \log(2N))$.

Here, m and N represent the number of objectives and the population size, respectively. We can see that the overall worst case complexity of the MODESA is $O(m^2 N \log N + mN^2)$.

3.5.3. The Difference between MODESA and MODEA. Considering that the proposed algorithm MODESA is an improvement of MODEA, it is necessary to provide the

TABLE 1: The parameters used in the proposed algorithm.

Parameter's name	Description	Value
N	The population size	100
m	Number of objectives	As the test problem
n	Number of variables in a solution	As the test problem
T	Maximum number of generations	250
cr	The crossover rate used in DE	0.3
f	The control parameter used in DE	0.5
T_{\max}	Maximum temperature in simulated annealing	100
T_{\min}	Minimal temperature in simulated annealing	$1e-7$
γ	The cooling rate in simulated annealing	0.6
max_life_cicle	Maximum prior life cycle value of the individual	1

difference between MODESA and MODEA. There are two main differences between MODESA and MODEA as follows.

- (1) When comparing the new solution and the current solution, there exist two processing ways according to two different results of comparison. If the new solution dominates the current solution, the MODEA replaces the current solution with the new solution and puts the current solution into the temporary population; otherwise, the new solution is taken into the temporary population. After each solution in the population is handled, the two populations are combined. It indicates that the new solutions are all retained. However, MODESA works in three kinds of ways according to different results. If the new solution dominates the current solution, the current solution is replaced with the new solution and the current solution is removed directly, which means that if the current solution is worse than the new solution, it would not be retained. If the new solution and the current solution are incomparable, the proposed simulated annealing algorithm is adopted to determine the acceptance of the current solution; moreover, the usage of a prior life cycle enables some potential solutions preferentially to be selected in the next generation. Otherwise, the new solution is added to the temporary population and the current solution is retained in the current population.
- (2) The diversity maintenance mechanism is different. In MODEA, the diversity preserving strategy based on crowding distance in NSGA-II is utilized, while in MODESA, the vicinity distance based pruning method in [30] is adopted.

4. Results and Discussion

4.1. Test Problems. The performance of the proposed algorithm is tested on a set of five biobjective and two triobjective optimization problems. The first five problems are ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6, which were described in [31]. The two triobjective optimization problems: DTLZ1 and DTLZ2 are taken from Deb [32]. None of these problems has any constraint. All objective functions are to be minimized.

The number of variables, their bounds, the Pareto-optimal solutions, and the characteristics of the Pareto-optimal front of these test problems are also shown in Table 2 [2]. For the test problems DTLZ1 and DTLZ2, the number of objectives is set to 3. Therefore, the number of variables for DTLZ1 and DTLZ2 is equal to 7 and 12, respectively.

4.2. Performance Measures. In this section, three performance measures will be introduced to evaluate the effectiveness of the proposed algorithm MODESA with the other algorithms. Here, let P^* and P denote the Pareto-optimal front and the Pareto front which are obtained by the algorithm, respectively. In this study, three performance measures are used for evaluating the convergence, the diversity, and both of them, separately. It is worth mentioning that all the three performance measures can be used only when the Pareto-optimal front of the particular test problem is known. In this paper, 500 evenly distributed solutions in PF are selected as P^* for each biobjective problem, and 990 solutions for triobjective problem, which is the same as that in [33]. The reader can download these solutions from the website <http://dces.essex.ac.uk/staff/qzhang/index.html>.

4.2.1. Inverted Generational Distance (IGD). The concept of inverted generational distance (IGD) is proposed by Coello and Cortés in [34]. The metric can be formally described as follows:

$$IGD = \frac{\sum_{x \in P^*} \min_{ed}(x, P)}{|P^*|}, \quad (10)$$

where $\min_{ed}(x, P)$ is the minimum Euclidean distance between the solution x and the solutions in P . The IGD-metric value can evaluate both the convergence and diversity of P to a certain degree. Algorithms with smaller value of IGD are satisfying.

4.2.2. Generational Distance (GD). van Veldhuizen and Lamont [35] proposed the concept of generational distance (GD). The metric calculates the distance between P and P^* . It can be defined as follows:

$$GD = \frac{\sqrt{\sum_{x \in P} \min_{ed}(x, p^*)^2}}{|P|}, \quad (11)$$

```

Step 1. Initialization
Initialize the values of the parameters in Table 1.
Generate the initial population  $P_0$  using uniform distribution.
Generate  $N$  opposite solutions denoted as  $P_{obl}$  by using OBL operation on  $P_0$ .
Select  $N$  best individuals as the current population by using non-dominated sorting and the
vicinity distance based pruning method on  $P_0 \cup P_{obl}$ .
For ( $i = 0; i < N; ++i$ ) {
     $life_i = 0$ .
} // End For
 $t = 0$ 

Step 2. Main loop
While ( $t \leq T$ ) {
     $temperature = T_{max}$ .
    Construct a temporary population which is denoted as  $P_{temp}$  and set  $P_{temp} = \emptyset$ .
    For ( $i = 0; i < N; ++i$ ) {
        Randomly select three distinct individuals:  $x_1, x_2$  and  $x_3$ , which are different
        from the current individual  $x_i$  from the current population.
        Select the best individual  $x_{best}$  from the three distinct individuals as the base vector
        for the next mutation operation.
        Produce a trial individual  $w_i$ :
        generate a random number  $jj \in \{1, \dots, n\}$ 
        For each variable  $j$  of  $w_i$  {
            With a probability  $cr$  or  $j$  is equal to  $jj$ 
            do
                 $w_{j,i} = x_{j,1} + f * (x_{j,2} - x_{j,3})$ .
            otherwise
                 $w_{j,i} = x_{j,best}$ .
            If ( $w_{j,i} > v_{j,max}$ )  $w_{j,i} = v_{j,max}$ .
            If ( $w_{j,i} < v_{j,min}$ )  $w_{j,i} = v_{j,min}$ .
        } // End For
        Check the domination status of the trial individual  $w_i$  and the current individual  $x_i$ .
        If ( $w_i$  dominates  $x_i$ ) {
            Replace the current individual  $x_i$  by the trial individual  $w_i$ :  $x_i = w_i$ .
        }
        Else if ( $w_i$  and  $x_i$  are incomparable) {
            Use the proposed simulated annealing to allocate the two individuals:  $w_i$  and  $x_i$ .
            SimulateAnnealing( $x_i, w_i, temperature, T_{min}, T_{max}, P_i, P_{temp}, max\_life\_cycle$ ).
             $temperature = \gamma * temperature$ .
        }
        Else if ( $w_i$  is dominated by  $x_i$ ) {
            Put the trial individual  $w_i$  to the temporary population.
        } // End If
    } // End For
    Union the two population:  $P_i \cup P_{adv}$ .
    If ( $t == T$ ) {
        Use the non-dominated sorting to the union population to remove the individuals that
        are dominated by any other solutions.
    } // End If
    Select  $N$  individuals as the next population from the union population by using the
    non-dominated sorting.
    For ( $i = 0; i < N; ++i$ ) {
        If ( $life_i > 0$ )
             $life_i --$ ;
    } // End For
    Apply the vicinity distance based pruning method.
} // End While

```

TABLE 2: Test problems used in the paper.

Problem	n	Variable bounds	Objective functions	Optimal solutions	Comments
ZDT1	30	$[0, 1]$	$f_1 = x_1$ $f_2 = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} \right]$ $g(x) = 1 + \frac{\sum_{i=2}^n x_i}{n-1}$	$x_1 \in [0, 1]$ $x = 0$ $i = 2, \dots, n$	Convex
ZDT2	30	$[0, 1]$	$f_1 = x_1$ $f_2 = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} \right]^2$ $g(x) = 1 + \frac{\sum_{i=2}^n x_i}{n-1}$	$x_1 \in [0, 1]$ $x = 0$ $i = 2, \dots, n$	Nonconvex
ZDT3	30	$[0, 1]$	$f_1 = x_1$ $f_2 = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} - \frac{x_1}{g(x)} \sin(10\pi x_1) \right]$ $g(x) = 1 + 9 \frac{\sum_{i=2}^n x_i}{n-1}$	$x_1 \in [0, 1]$ $x = 0$ $i = 2, \dots, n$	Convex disconnected
ZDT4	10	$x_1 \in [0, 1]$ $x_i \in [-5, 5]$	$f_1 = x_1$ $f_2 = g(x) \left[1 - \sqrt{\frac{x_1}{g(x)}} \right]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^n [x_i^2 - 10 \cos(4\pi x_i)]$	$x_1 \in [0, 1]$ $x = 0$ $i = 2, \dots, n$	Nonconvex
ZDT6	10	$[0, 1]$	$f_1 = 1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $f_2 = g(x) \left[1 - \left(\frac{f_1}{g(x)} \right)^2 \right]$ $g(x) = 1 + 9 \left(\frac{\sum_{i=2}^n x_i}{n-1} \right)^{0.25}$	$x_1 \in [0, 1]$ $x = 0$ $i = 2, \dots, n$	Nonconvex
DTLZ1	$m+4$	$[0, 1]$	$f_1 = \frac{1}{2} x_1 x_2 \dots x_{m-1} (1 + g(x_m))$ $f_2 = \frac{1}{2} x_1 x_2 \dots x_{m-1} (1 - x_{m-1}) (1 + g(x_m))$ \vdots $f_{m-1} = \frac{1}{2} x_1 (1 - x_2) (1 + g(x_m))$ $f_m = \frac{1}{2} (1 - x_1) (1 + g(x_m))$	$x_i = 0.5$ $i = 1, 2, \dots, n$	
			$g(x_m) = 100 \left[x_m + \sum_{x_i \in x_m} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5)) \right]$		

TABLE 2: Continued.

Problem	n	Variable bounds	Objective functions	Optimal solutions	Comments
			$f_1 = (1 + g(x_m)) \cos\left(\frac{x_1\pi}{2}\right) \cos\left(\frac{x_2\pi}{2}\right) \cdots \cos\left(\frac{x_{m-2}\pi}{2}\right) \cos\left(\frac{x_{m-1}\pi}{2}\right)$		
	$m + 9$	$[0, 1]$	$f_2 = (1 + g(x_m)) \cos\left(\frac{x_1\pi}{2}\right) \cos\left(\frac{x_2\pi}{2}\right) \cdots \cos\left(\frac{x_{m-2}\pi}{2}\right) \sin\left(\frac{x_{m-1}\pi}{2}\right)$		
DTLZ2			$f_3 = (1 + g(x_m)) \cos\left(\frac{x_1\pi}{2}\right) \cos\left(\frac{x_2\pi}{2}\right) \cdots \sin\left(\frac{x_{m-2}\pi}{2}\right)$		$x_i = 0.5$ $i = 1, 2, \dots, n$
			\vdots		
			$f_m = (1 + g(x_m)) \sin\left(\frac{x_1\pi}{2}\right)$		
			$g(x_m) = \sum_{x_i \in x_m} (x_i - 0.5)^2$		

TABLE 3: The mean and standard deviation IGD-metric values of nondominated solutions in 10 runs on ZDTs where $N = 100$ and $T = 250$.

IGD	NSGA-II (mean \pm std)	MODEA (mean \pm std)	MODESA (mean \pm std)
ZDT1	0.00564974 \pm 0.000254893	0.00436863 \pm 0.000243648	0.00403528 \pm 7.01666e - 05
ZDT2	0.00557228 \pm 0.000242512	0.00451929 \pm 0.000134666	0.00408871 \pm 4.65746e - 05
ZDT3	0.00592808 \pm 0.000205118	0.00521698 \pm 0.000159453	0.00494596 \pm 0.000126657
ZDT4	0.0069977 \pm 0.00102386	0.0301279 \pm 0.0752722	0.0171826 \pm 0.0362604
ZDT6	0.0137079 \pm 0.00170579	0.00304881 \pm 0.000125906	0.0022843 \pm 0.00014073

TABLE 4: The mean and standard deviation Δ -metric values of nondominated solutions in 10 runs on ZDTs where $N = 100$ and $T = 250$.

Spread	NSGA-II (mean \pm std)	MODEA (mean \pm std)	MODESA (mean \pm std)
ZDT1	0.372425 \pm 0.0513842	0.333973 \pm 0.0772084	0.297395 \pm 0.0308609
ZDT2	0.385209 \pm 0.0291848	0.336548 \pm 0.0181084	0.296842 \pm 0.0282846
ZDT3	0.389909 \pm 0.0557332	0.355518 \pm 0.0370526	0.266801 \pm 0.0211668
ZDT4	0.399555 \pm 0.0786079	0.377719 \pm 0.0574188	0.352871 \pm 0.107095
ZDT6	0.407279 \pm 0.0276428	0.407335 \pm 0.0487992	0.368972 \pm 0.0463564

where $\min_{ed}(\mathbf{x}, P^*)$ is the minimum Euclidean distance of solution \mathbf{x} and the solutions in P^* . GD evaluates the closeness of the obtained Pareto front and the Pareto-optimal front. Algorithms with smaller value of GD are desirable.

4.2.3. *Spread* (Δ). The spread metric Δ was firstly proposed by Deb et al. [2] to measure the spread effect of the nondominated solutions obtained by the algorithm. However, it is only suitable for biobjective problems. Wang et al. [19] further extended the metric Δ and made it suitable to measure problems that have more than two objectives. The improved Δ can be stated mathematically as follows:

$$\Delta = \frac{\sum_{i=1}^m d(E_i, \Omega) + \sum_{X \in \Omega} |d(X, \Omega) - \bar{d}|}{\sum_{i=1}^m d(E_i, \Omega) + (|\Omega| - m)\bar{d}}$$

$$d(X, \Omega) = \min_{Y \in \Omega, Y \neq X} \|F(X) - F(Y)\| \quad (12)$$

$$\bar{d} = \frac{1}{|\Omega|} \sum_{X \in \Omega} d(X, \Omega),$$

where Ω is a set of solutions, m is the number of objectives, and (E_1, \dots, E_m) are m extreme solutions in the set of true Pareto front (PF). The smaller the value of Δ , the better the distribution and diversity of the obtained nondominated solutions.

4.3. *Experimental Setup*. In this paper, the population size and the number of generations are set to 100 and 250, respectively. For the other parameters in NSGA-II and MODEA, they have remained the same with their original studies [3, 23]. For MODESA, there are four parameters: T_{\max} , T_{\min} , γ , and \max_life_cycle that need to be set. Here we set $T_{\max} = 100$, $T_{\min} = 1e - 7$, $\gamma = 0.6$, and $\max_life_cycle = 1$ for the study. Each algorithm runs 10 times independently for each problem. All the test problems have been executed on Microsoft Window XP Intel core 2 Duo CPU E8400 3.00 GHz, with 2 GB RAM. All methods have been implemented in C++.

4.4. Discussion of the Results

4.4.1. *Results of Biobjective Optimization Problems*. Tables 3 and 4 show the mean and standard deviation IGD-metric and Δ -metric values of the obtained nondominated solutions in 10 runs on the group of ZDTs. It is obvious from Tables 3 and 4 that the IGD-metric values of MODESA achieve the best value in the three algorithms except for the test problem ZDT4.

The test problems ZDT1 and ZDT2 may be the most simple problems in the series of ZDTs. The Pareto-optimal front of ZDT1 is convex while that of ZDT2 is nonconvex. It is clear from Tables 3 and 4 that both the values of the IGD metric and the spread metric of MODESA are better than that of NSGA-II and MODEA. Figures 1 and 2 demonstrate that MODESA converges to the Pareto-optimal front and has a good spread over the entire front on the test problems ZDT1 and ZDT2. It can be concluded that the proposed MODESA can well deal with these two kinds of test problems.

The third test problem ZDT3 is different from the above two test problems. The Pareto-optimal front of ZDT3 consists of five disjoint curves. From Figure 3, it is obvious that the nondominated solutions obtained by the MODESA almost converge to the Pareto-optimal front and have a good distribution over the entire front. From Tables 3 and 4, MODESA obtains the best values of IGD-metric and spread-metric between the three algorithms.

The fourth test problem is ZDT4 which is perhaps the most difficult problem in the five problems. It has 21^9 different local Pareto-optimal fronts that confuse the multiobjective optimization algorithm [2]. From Tables 3 and 4, NSGA-II obtains the best values on the IGD-metric and the spread-metric in these three algorithms. In fact, in the 10 independent runs, some results of MODEA and MODESA are better than that of NSGA-II; however, MODEA and MODESA get bad results in a few runs which affects the mean value of the 10 runs. Indeed, the best IGD-metric value of MODESA in the 10 runs is 0.0044915, which is better than NSGA-II. Figure 4 demonstrates the plots of the nondominated solutions with the lowest IGD-metric

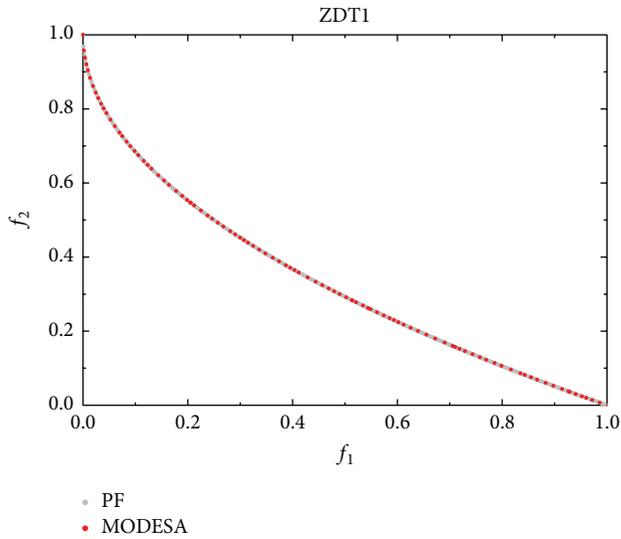


FIGURE 1: Plots of the nondominated solutions with the lowest IGD-metric values found by MODESA in 10 runs in the objective space on ZDT1 where $N = 100$ and $T = 250$.

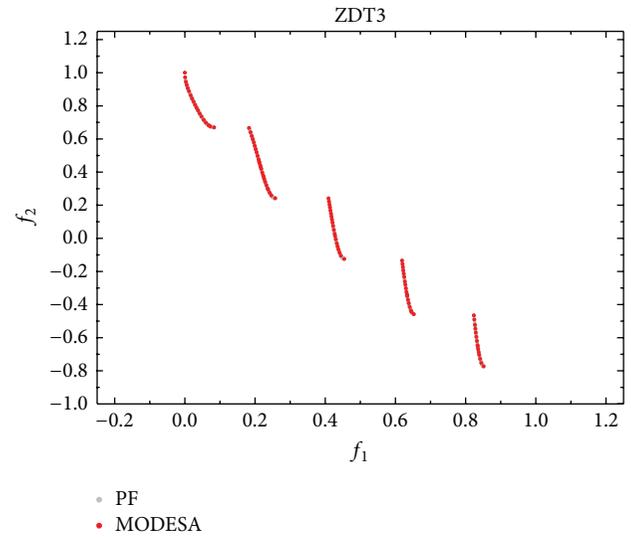


FIGURE 3: Plots of the nondominated solutions with the lowest IGD-metric values found by MODESA in 10 runs in the objective space on ZDT3 where $N = 100$ and $T = 250$.

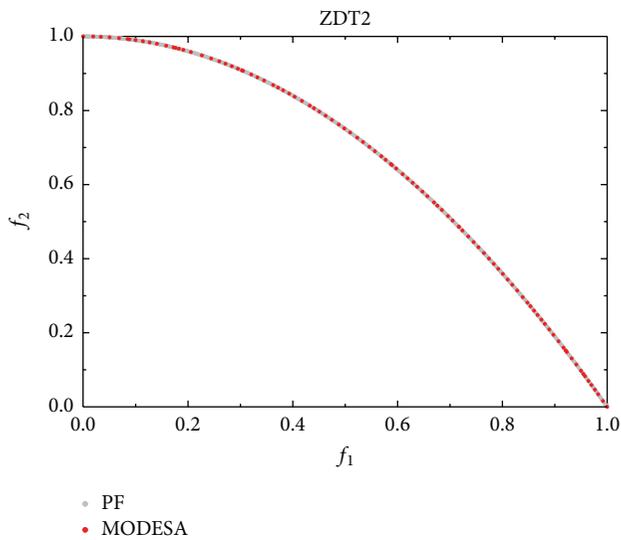


FIGURE 2: Plots of the nondominated solutions with the lowest IGD-metric values found by MODESA in 10 runs in the objective space on ZDT2 where $N = 100$ and $T = 250$.

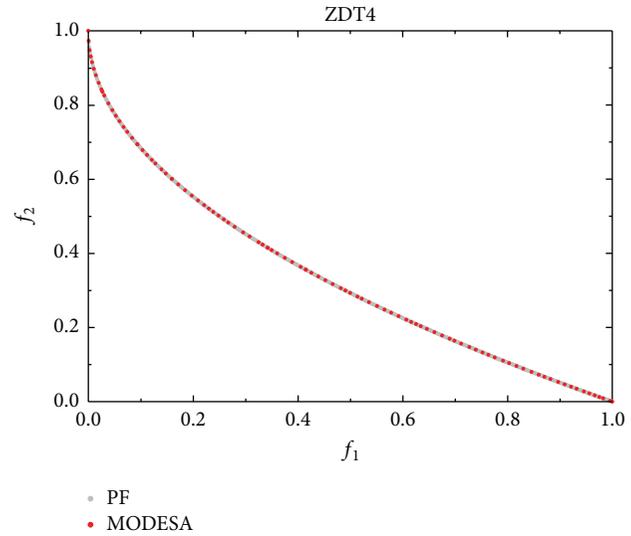


FIGURE 4: Plots of the nondominated solutions with the lowest IGD-metric values found by MODESA in 10 runs in the objective space on ZDT4 where $N = 100$ and $T = 250$.

values found by MODESA in the 10 runs. It is clear from Figure 4 that MODESA can get quite good convergence and distribution over the Pareto-optimal front, which reveals that the proposed algorithm is a promising but unstable algorithm for solving the test problem ZDT4. We will leave this problem for future work.

The last test problem ZDT6 is such a problem with thin density and nonuniform spread of solutions [23]. We can observe from Tables 3 and 4 that MODESA finds the best convergence and spread in comparison to the other two algorithms. To show the superiority of MODESA, plots of the nondominated solutions found by MODESA and MODEA on ZDT 6 are shown in Figures 5 and 6, respectively. It is clear

from Figures 5 and 6 that MODESA gets a better spread than MODEA.

4.4.2. Results of Triobjective Optimization Problems. In this section, to show the effectiveness of the proposed MODESA for solving problems that have more than two objectives, we choose two triobjective optimization problems DTLZ1 and DTLZ2 for comparison. The results of NSGA-II, MODEA, and MODESA are given in Table 5. It is obvious from Table 5 that MODESA performs better than the other two algorithms in terms of both the IGD-metric and spread-metric. Figures 7, 8, 9, and 10 show the nondominated solutions obtained by MODEA and MODESA, and it is obvious that MODESA

TABLE 5: The mean and standard deviation IGD-metric and Δ -metric values of nondominated solutions in 10 runs on DTLZ1 and DTLZ2 where $N = 100$ and $T = 250$.

Metric Problem	IGD		Spread	
	DTLZ1	DTLZ2	DTLZ1	DTLZ2
NSGA-II (mean \pm std)	0.025008 \pm 0.001199	0.077375 \pm 0.021572	0.60056 \pm 0.076489	0.530832 \pm 0.029126
MODEA (mean \pm std)	0.022851 \pm 0.000557	0.060254 \pm 0.001664	0.41962 \pm 0.019617	0.425707 \pm 0.033528
MODESA (mean \pm std)	0.020602 \pm 0.00035	0.055011 \pm 0.000799	0.26489 \pm 0.045448	0.260335 \pm 0.02856

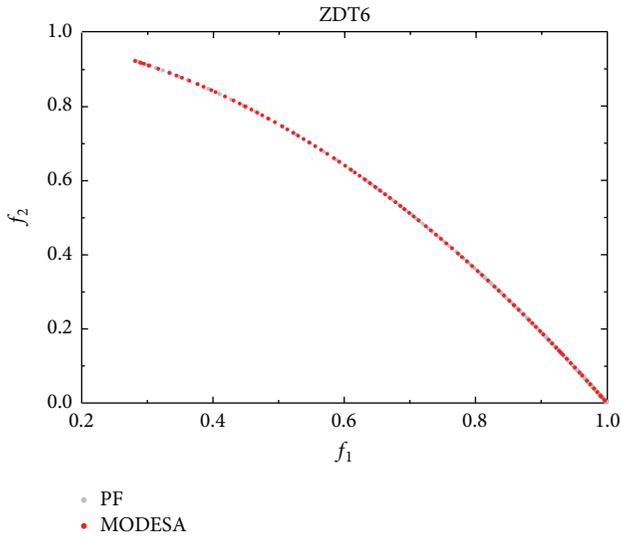


FIGURE 5: Plots of the nondominated solutions with the lowest IGD-metric values found by MODESA in 10 runs in the objective space on ZDT6 where $N = 100$ and $T = 250$.

outperforms MODEA on both of the two triobjective optimization problems. In particular, it is clear that MODESA has a better distribution than MODEA, the reason is that the vicinity distance based pruning method performs better than the crowding distance of Deb in maintaining diversity and distribution of the nondominated solutions.

4.5. Effect of the Proposed Simulated Annealing Procedure. As mentioned in Section 3.5.3, one of the main differences of MODEA and MODESA is the employment of the proposed simulated annealing procedure. To show the effect of the proposed simulated annealing procedure, a version of MODESA: MODESA-CD (multiobjective differential evolution with crowding distance) is proposed for comparison. The MODESA-CD is analogous to MODESA except that it replaces the vicinity distance based pruning method in MODESA with the crowding distance based approach in NSGA-II. In other words, the difference between MODEA and MODESA-CD is the using of the proposed simulated annealing procedure. Table 6 shows the mean and standard deviation GD-metric values of nondominated solutions in 10 runs where $N = 100$ and $T = 250$. It is obvious from Table 6 that all the GD-metric values of MODESA-CD are better than that of MODEA. The reason of the bad performance of MODEA is that MODEA always accepts the current and the new solutions to the next generation and that makes

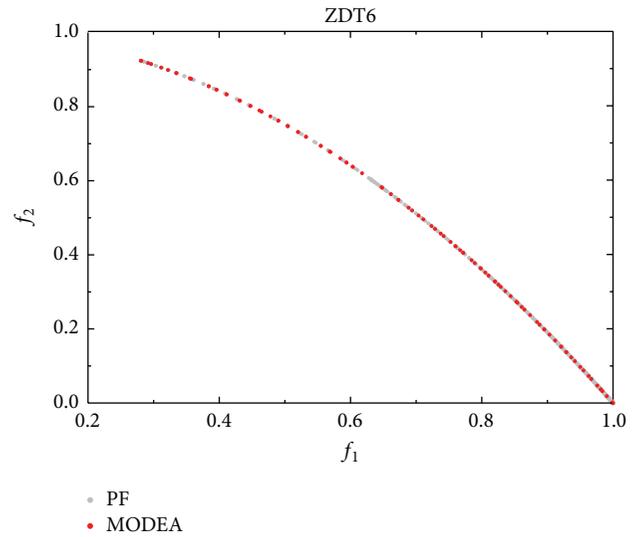


FIGURE 6: Plots of the nondominated solutions with the lowest IGD-metric values found by MODEA in 10 runs in the objective space on ZDT6 where $N = 100$ and $T = 250$.

it chaotic and unable to have a right direction towards the Pareto-optimal front. In MODESA-CD, we remove the bad solutions and only accept some potential solutions in an alterable probability.

5. Conclusion

In this paper, an enhanced differential evolution based algorithm with simulated annealing, named MODESA, is presented for solving MOPs. During the selection, MODESA employs a procedure of simulated annealing to control the acceptance of every candidate solution, and a concept of the prior life circle is proposed to allow some potential solutions entering the next generation to escape from the local optimum. Finally, a fast and efficient diversity maintenance mechanism is adopted. The proposed algorithm is tested on five biobjective and two triobjective optimization problems in terms of IGD-metric and spread metric, and the experimental results show that MODESA outperforms the other two algorithms on these test problems except on ZDT4. Furthermore, the effect of convergence is also tested in comparison to MODEA, the computational results show the superiority of the proposed MODESA.

Improvement can be made further in the performance of the proposed MODESA on the test problem ZDT4. In a word, MODESA is an easy and efficient method for solving MOPs.

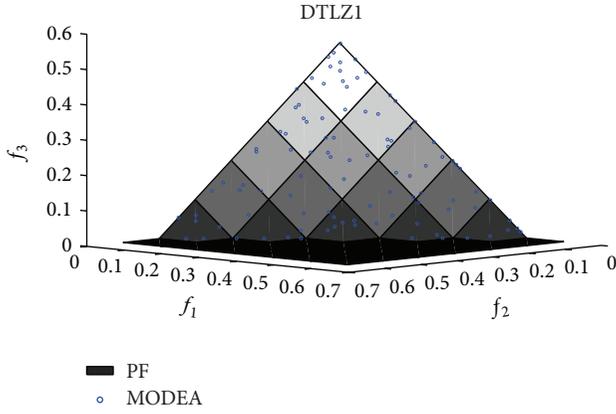


FIGURE 7: Plots of the nondominated solutions with the lowest IGD-metric values found by MODEA in 10 runs in the objective space on DTLZ1 where $N = 100$ and $T = 250$.

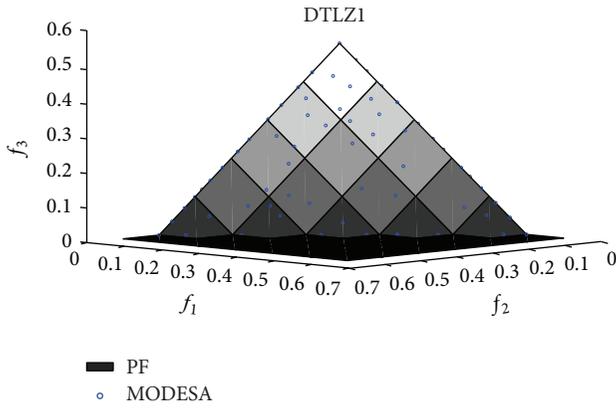


FIGURE 8: Plots of the nondominated solutions with the lowest IGD-metric values found by MODESA in 10 runs in the objective space on DTLZ1 where $N = 100$ and $T = 250$.

TABLE 6: The mean and standard deviation GD-metric values of nondominated solutions in 10 runs where $N = 100$ and $T = 250$.

GD	MODEA (mean \pm std)	MODESA-CD (mean \pm std)
ZDT1	$0.000124 \pm 8.236e - 06$	$0.000116 \pm 1.437e - 05$
ZDT2	$4.819e - 05 \pm 1.642e - 06$	$4.771e - 05 \pm 2.596e - 06$
ZDT3	$0.00035 \pm 6.128e - 06$	$0.000345 \pm 2.714e - 05$
ZDT4	0.002812 ± 0.008057	0.001365 ± 0.003738
ZDT6	$0.000429 \pm 1.78e - 05$	$0.000404 \pm 2.614e - 05$
DTLZ1	$0.000308 \pm 7.454e - 05$	$0.00029 \pm 2.129e - 05$
DTLZ2	0.000677 ± 0.000103	$0.000633 \pm 3.871e - 05$

In the near future, applying MODESA for constrained MOPs and real-life application problems are our work.

Conflict of Interests

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

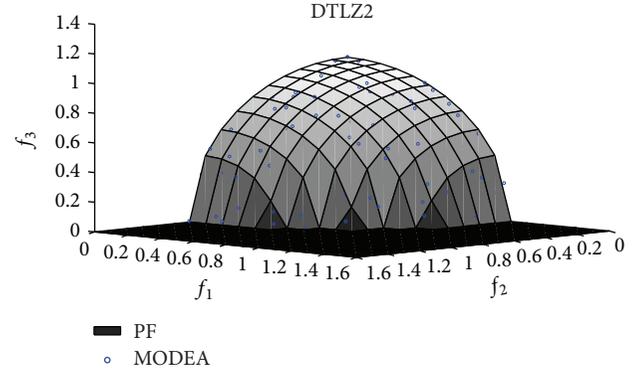


FIGURE 9: Plots of the nondominated solutions with the lowest IGD-metric values found by MODEA in 10 runs in the objective space on DTLZ2 where $N = 100$ and $T = 250$.

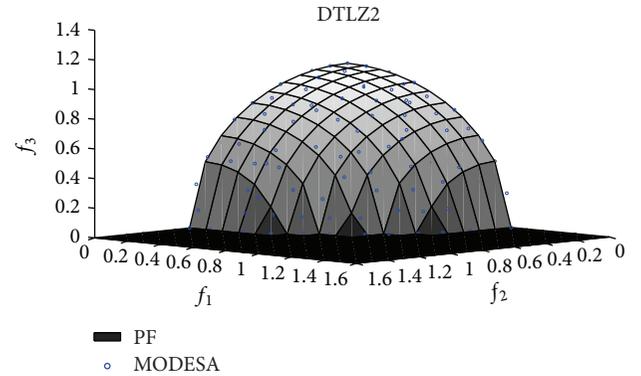


FIGURE 10: Plots of the nondominated solutions with the lowest IGD-metric values found by MODESA in 10 runs in the objective space on DTLZ2 where $N = 100$ and $T = 250$.

Acknowledgment

This work is supported by the National Science Foundation of China (nos. 60672018 and 40774065).

References

- [1] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [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, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," Tech. Rep., Swiss Federal Institute of Technology, Lausanne, Switzerland, 2003.
- [4] 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.
- [5] C. A. Coello Coello, "An updated survey of GA-based multiobjective optimization techniques," *ACM Computing Surveys*, vol. 32, no. 2, pp. 109–143, 2000.

- [6] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Springer, New York, NY, USA, 2007.
- [7] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhangd, “Multiobjective evolutionary algorithms: a survey of the state of the art,” *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [8] R. Storn and K. Price, “Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces,” Tech. Rep. TR-95-012, Berkeley, Calif, USA, 1995.
- [9] 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.
- [10] K. V. Price, “Differential evolution vs. the functions of the 2nd ICEO,” in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC '97)*, pp. 153–157, April 1997.
- [11] H. A. Abbass, R. Sarker, and C. Newton, “PDE: a Pareto-frontier differential evolution approach for multi-objective optimization problems,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '01)*, pp. 971–978, Piscataway, NJ, USA, May 2001.
- [12] H. A. Abbass, “The self-adaptive Pareto differential evolution algorithm,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 831–836, Honolulu, Hawaii, 2002.
- [13] N. K. Madavan, “Multiobjective optimization using a Pareto differential evolution approach,” in *Proceedings of the Congress on Evolutionary Computation*, pp. 1145–1150, Honolulu, Hawaii, 2002.
- [14] F. Xue, A. C. Sanderson, and R. J. Graves, “Multi-objective differential evolution and its application to enterprise planning,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3535–3541, Taipei, Taiwan, September 2003.
- [15] K. E. Parsopoulos, D. K. Tasoulis, N. G. Pavlidis, V. P. Plagianakos, and M. N. Vrahatis, “Vector evaluated differential evolution for multiobjective optimization,” in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 204–211, Portland, Ore, USA, June 2004.
- [16] S. Kukkonen and J. Lampinen, “An extension of generalized differential evolution for multi-objective optimization with constraints,” in *Parallel Problem Solving From Nature (PPSN2004)*, pp. 752–761, 2004.
- [17] S. Kukkonen and J. Lampinen, “GDE3: the third evolution step of generalized differential evolution,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '05)*, pp. 443–450, Edinburgh, UK, September 2005.
- [18] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. Coello, R. Caballero, and J. Molina, “A new proposal for multi-objective optimization using differential evolution and rough sets theory,” in *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, pp. 675–682, Seattle, Wash, USA, July 2006.
- [19] Y.-N. Wang, L.-H. Wu, and X.-F. Yuan, “Multi-objective self-adaptive differential evolution with elitist archive and crowding entropy-based diversity measure,” *Soft Computing*, vol. 14, no. 3, pp. 193–209, 2010.
- [20] V. L. Huang, A. K. Qin, P. N. Suganthan, and M. F. Tasgetiren, “Multi-objective optimization based on self-adaptive differential evolution algorithm,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '07)*, pp. 3601–3608, September 2007.
- [21] V. L. Huang, S. Z. Zhao, R. Mallipeddi, and P. N. Suganthan, “Multi-objective optimization using self-adaptive differential evolution algorithm,” in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 190–194, May 2009.
- [22] M. Ali, M. Pant, and A. Abraham, “A modified differential evolution algorithm and its application to engineering problems,” in *Proceedings of the International Conference on Soft Computing and Pattern Recognition (SoCPaR '09)*, pp. 196–201, December 2009.
- [23] M. Ali, P. Siarry, and M. Pant, “An efficient Differential Evolution based algorithm for solving multi-objective optimization problems,” *European Journal of Operational Research*, vol. 217, no. 2, pp. 404–416, 2012.
- [24] H. R. Tizhoosh, “Opposition-based learning: a new scheme for machine intelligence,” in *Proceedings of the International Conference on Computational Intelligence for Modelling Control and Automation (CIMCA '05)*, pp. 695–701, November 2005.
- [25] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [26] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.
- [27] B. Suman and P. Kumar, “A survey of simulated annealing as a tool for single and multiobjective optimization,” *Journal of the Operational Research Society*, vol. 57, no. 10, pp. 1143–1160, 2006.
- [28] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, “A simulated annealing-based multiobjective optimization algorithm: AMOSA,” *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, 2008.
- [29] K. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer Academic Publishers, 1999.
- [30] S. Kukkonen and K. Deb, “A fast and effective method for pruning of non-dominated solutions in many-objective problems,” in *Parallel Problem Solving from Nature—PPSN IX*, vol. 4193 of *Lecture Notes in Computer Science*, pp. 553–562, 2006.
- [31] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: empirical results,” *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [32] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, UK, 2001.
- [33] 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.
- [34] C. A. C. Coello and N. C. Cortés, “Solving multiobjective optimization problems using an artificial immune system,” *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163–190, 2005.
- [35] D. A. van Veldhuizen and G. B. Lamont, “Multiobjective evolutionary algorithm research: a history and analysis,” Tech. Rep. TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Inst Technol, Wright Patterson, Ohio, USA, 1998.