

## Research Article

# An MOEA/D-ACO with PBI for Many-Objective Optimization

Tianbai Ling<sup>1</sup> and Chen Wang<sup>1,2</sup> 

<sup>1</sup>Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai University, Shanghai 200072, China

<sup>2</sup>Dept of Mechanical Engineering, Hubei Automotive Industries Institute, Shiyan 442002, China

Correspondence should be addressed to Chen Wang; 20090011@huat.edu.cn

Received 1 March 2018; Revised 15 May 2018; Accepted 6 August 2018; Published 30 August 2018

Academic Editor: Erik Cuevas

Copyright © 2018 Tianbai Ling and Chen Wang. 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.

Evolutionary algorithms (EAs) are an important instrument for solving the multiobjective optimization problems (MOPs). It has been observed that the combined ant colony (MOEA/D-ACO) based on decomposition is very promising for MOPs. However, as the number of optimization objectives increases, the selection pressure will be released, leading to a significant reduction in the performance of the algorithm. It is a significant problem and challenge in the MOEA/D-ACO to maintain the balance between convergence and diversity in many-objective optimization problems (MaOPs). In the proposed algorithm, an MOEA/D-ACO with the penalty based boundary intersection distance (PBI) method (MOEA/D-ACO-PBI) is intended to solve the MaOPs. PBI decomposes the problems with many single-objective problems, a weighted vector adjustment method based on clustering, and uses different pheromone matrices to solve different single objectives proposed. Then the solutions are constructed and pheromone was updated. Experimental results on both CF1-CF4 and suits of C-DTLZ benchmarks problems demonstrate the superiority of the proposed algorithm in comparison with three state-of-the-art algorithms in terms of both convergence and diversity.

## 1. Introduction

Over the past few decades, EAs have been successfully implemented to various real-world optimization problems with two or three objectives. However, EAs face many challenges when dealing with MOPs, which have more than three objectives and are commonly referred to as MaOPs [1]. The foremost reason is the case that the proportion of nondominated solutions in the population rises sharply with an increase in the number of objectives [2]. The Pareto dominate relation cannot provide sufficient selection pressure to ensure that the population evolve [3]. In recent years, more and more multiobjective evolutionary optimization algorithms (MOEAs) have been improved for solving MaOPs [4].

In recent years, as system functions become more and more complex. MaOPs have become a hot topic in the field of evolutionary multiobjective optimization community [5]. It has been found that EAs have succeeded in solving various optimization problems. However, as the number of optimization objectives increases, sufficient selection pressure will be released, resulting in a significant reduction in

the performance of the algorithm. It is a challenging major problem and challenge in the EAs to maintain the balance between convergence and diversity in MaOPs.

At present, more and more MOEAs have made extraordinary improvement in solving MaOPs. The most popular method is the Pareto-dominance and modified Pareto-dominance that relaxes the form of Pareto-dominance to increase the selection pressure towards the Pareto front, for example, H Jain and Deb based on the NSGAIII framework to solve generic MaOPs [6] and fuzzy Pareto-dominance [7]. The disadvantage of the above methods is introduced in one or more parameters and needs to be adjusted heuristically. The indicator-based algorithm is promising method for MaOPs. It uses a single indicator value to guide the evolution of the population, such as R2 indicator [8] and hypervolume indicator [9]. The last method is based on decomposition. Decomposition strategy and neighbourhood concept are introduced in this method. As one of the popular algorithms, MOPs based on decomposition (MOEA/D) were proposed by Zhang and Li [10]. The MOEA/D uses aggregation function to compare the solution with the uniformly distributed weight vector to retain solution convergence and diversity [4, 11].

MOEA/D gets a real good convergence and diversity and low computational complexity and thus has an effective method.

ACO originally proposed dealing with single-objective combination optimization problems [12]. The successful application of ACO in single-objective optimization has led some researchers to develop ACO algorithms for multiobjective optimization problems. Recently, a multiobjective evolutionary algorithm for multiobjective combinatorial optimization problem based on decomposition and ant colony algorithm (MOEA/D-ACO) was proposed. Ke et al. combined ACO and decomposition-based multiobjective evolutionary algorithm (MOEAD/ACO) to solve biobjective TSP [13]. Despite the advance in adapting MOEAs for leading with MaOPs, the MOEA/D-ACO performed much better than the BicriterionAnt in all the test instances. The MOEA/D-ACO algorithm successfully solves the biobjective TSP problem. However, its performance in cases with three or more objectives requires further study. Murilo Zangari Souza et al. proposed a parallel implementation of MOEA/D-ACO on a graphics processing unit (GPU) using NVIDIA CUDA to improve efficiency and achieve high-quality results within a reasonable execution time [14].

At present, the MOEA/D-ACO optimization algorithm has been introduced to solve MaOPs in numerous studies. However, there are few studies on the further use of MOEA/D-ACO to solve the MaOPs. In this paper, we introduce the PBI method for MOEA/D-ACO to solve the MaOPs. And many-objective optimization techniques can be incorporated into our method to maintain convergence and diversity of algorithm.

From the discussion above, we propose a new algorithm with MOEA/D-ACO-PBI designed to solve the MaOPs. This paper mainly focuses on the algorithm that promotes the performance of convergence and diversity in many-objective optimization.

This remainder of this paper is organized as follows. In Section 2, the related work and background were discussed. The MOEA/D-ACO-PBI algorithm is described in Section 3. In Section 5, the experimental studies are provided to demonstrate the efficiency of the proposed method as well as some discussions on this paper. Finally, Section 6 makes a conclusion and points out some future research directions.

## 2. Related Background

**2.1. Basic Definitions.** General MaOPs can be define as follows:

$$\text{minimize } F(x) = (f_1(x), f_2(x), \dots, (f_m(x)^T)) \quad (1)$$

$$\text{subject to } x \in \Omega \subseteq \mathbb{R}_n$$

where  $x = (x_1, \dots, x_n)^T$  is the  $n$ -dimensional decision variable vector,  $\Omega$  is the decision space, and  $F : \Omega \rightarrow \mathbb{R}^m$  consists of  $m$  objective functions,  $f_i(x)$  ( $i = 1, 2, \dots, m$ ),  $m \geq 4$ .  $\mathbb{R}^m$  is called the objective space.

Given two decision vectors  $x, y \in \Omega$ ,  $x$  is said to Pareto dominate  $y$  ( $x \prec y$ ), if

$$\forall i \in \{1, 2, \dots, m\}: f_i(x) \leq f_i(y) \quad (2)$$

$$\exists i \in \{1, 2, \dots, m\}: f_i(x) < f_i(y)$$

A solution  $x^* \in \Omega$  is Pareto optimal if there is no  $x \prec x^*$ .  $f(x^*)$  is called a Pareto optimal objective vector. The set of all Pareto optimal objective vectors is called the Pareto front (PF) and the set of all Pareto optimal solutions is called the Pareto optimal set (PS).

**2.2. MOEA/D.** MOEA/D: it is the most popular multiobjective evolutionary algorithm in recent years. It first initializes  $N$  uniform and widely distributed vectors for each individual evolving in the direction of each vector, which is equivalent to dividing multiobjective optimization problems. The solution to the  $N$  scalar optimization subproblems is through the constant. Evolution can simultaneously obtain the optimal solution of these  $N$  subproblems. The algorithm proposes 3 weighted methods that can be used: weighted sum aggregation, Tchebycheff approach, and PBI approach.

The traditional weighted sum aggregation is the most direct method to deal with multiobjective optimization problems, and it is also a method used earlier. This method combines or aggregates all the optimized subobjectives into a single-objective, and the multiobjective optimization problem is converted into a single-objective optimization problem.

The expression of weighted sum aggregation is

$$\min g^{ws}(x | \lambda) = \sum_{i=1}^m \lambda_i f_i(x) \quad (3)$$

The weighted sum aggregation method generates a set of different Pareto optimal solutions by generating different weight vectors in the above scalar optimization problem. However, in the case that the shape of the optimal Pareto surface is nonconvex, this method cannot obtain every Pareto optimal vector.

The expression of Tchebycheff approach is

$$\min g^{te}(x | \lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i | f_i(x) - z_i^*\} \quad (4)$$

For each Pareto optimality  $x$  there is always a weight vector  $\lambda$  such that the solution of (4) is a Pareto optimal solution, which corresponds to the Pareto optimal solution of the original multiobjective problem. And by modifying the weight vector, different solutions on the Pareto optimal surface can be obtained. One disadvantage of this approach is that the aggregate method curves are not smooth when dealing with continuous multiobjective problems.

The PBI method aims to find the intersection of the uppermost boundary and a set of lines. In a sense, if the lines are evenly distributed, the resulting intersections can be seen as providing a good approximation to the overall Pareto optimal boundary. These methods can deal with the Pareto optimal boundary nonconvex problem. The PBI method will be described in detail later.

**2.3. MOEA/D-ACO.** The basic framework of MOEA/D-ACO first decomposes the MOPs into  $N_s$  single-objective subproblems by  $N$  weight vectors  $\lambda_1, \dots, \lambda_N$ . Then, MOEA/D-ACO utilizes  $N$  ants to solve these single-objective subproblems. Ant  $i_a$  is responsible for subproblems  $i_s$ . Finally,

- (1) Initialization: generate  $N_{ini}$  Solutions by weight vectors  $\lambda$
- (2) Decomposition by some aggregation approach
- (3) Solution construction
- (4) Update of EP
- (5) Update of pheromone
- (6) Check the solutions in the neighbourhood and update the solutions
- (7) Termination: judgment and output EP (a set of high-quality solutions:  $X_s$ )

ALGORITHM 1: MOEA/D-ACO.

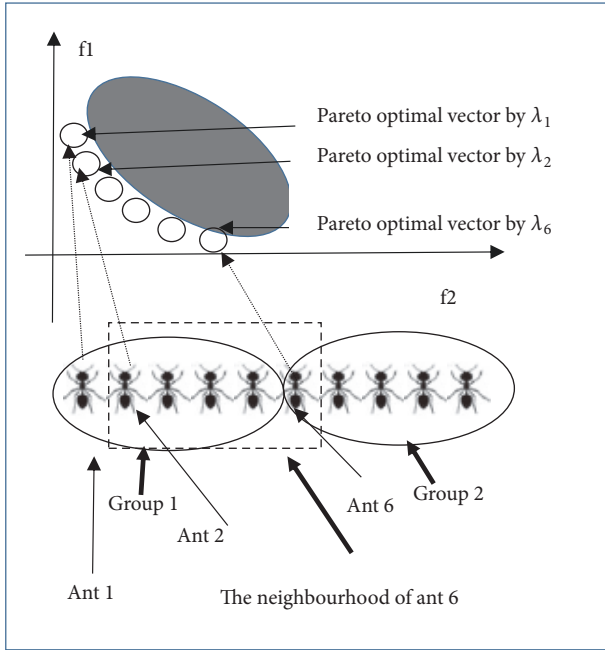


FIGURE 1: MOEA/D-ACO.

MOEA/D-ACO uses the concept of neighbourhood and group for decomposing the MaOPs. Figure 1 illustrates the method of above.

Firstly, MOEA/D-ACO decomposes the MOP into  $N_s$  single-objective subproblems by choosing  $N_w$  weight vectors ( $\lambda^1 \dots \lambda^{N_w}$ ). Then the subproblem  $i_s$  was associated with weight vector  $\lambda^{is}$ , and its objective function is denoted as  $g(x | \lambda^{is})$ . MOEA/D-ACO utilizes  $N_a$  ants for solving these single-objective subproblems, while ant  $i_a$  is responsible for subproblem  $i_s$ . Both the weighted sum and Tchebycheff approaches are used in MOEA/D-ACO and are recognized as MOEA/D-ACO<sup>W</sup> and MOEA/D-ACO<sup>T</sup>, respectively. However, recently, Deb et al. proposed that PBI decomposition method is more suitable for solving MOPs.

The MOEA/D-ACO is shown in Algorithm 1. In Algorithm 1, for each generation, each ant constructs a solution in Step (3). In Step (4), the newly constructed solution is updated in the external archive, and an ant updates the pheromone matrix of its group in Step (5), if it has found a new nondominated solution. In Step (6), each ant updates its current solution if there is a solution that (1) is better than its

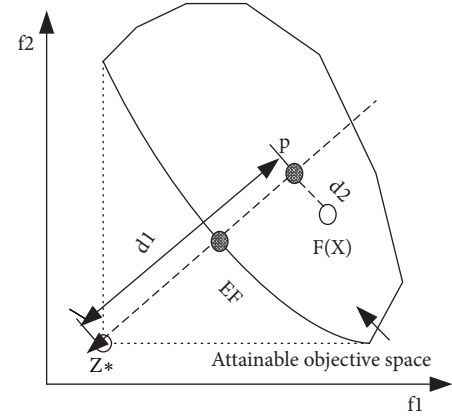


FIGURE 2: Illustration of PBI approach.

current solutions; and (2) has not been used for updating any other old solution. The algorithm is terminated in Step (7).

**2.4. PBI Method.** In our paper, we use the PBI approach, because of its promising performance for many-objective optimization problems. The optimization problem of the PBI approach is defined as

$$\begin{aligned} \text{minimize} \quad & g_{pbi}(x | w, z^*) = d_1 + \theta d_2 \\ \text{subject to} \quad & x \in \Omega \end{aligned} \quad (5)$$

where

$$\begin{aligned} d_1 &= \frac{|(F(x) - z^*) T_w|}{|W|} \\ d_2 &= \frac{|F(x) - (z^* + d_1 W)|}{|W|} \end{aligned} \quad (6)$$

$Z^* = (z_1^* \dots z_m^*)^T$  is the ideal objective vector with  $z_i^* < \min f_i(x), x \in \Omega, i \in \{1 \dots m\}$ , a penalty parameter  $\theta$  predefined by the user. Figure 2 presents an example to illustrate  $d_1$  and  $d_2$  of a solution  $x_i$ . In the PBI approach,  $d_1$  is used to measure the convergence of  $x_i$  towards the EF;  $d_2$  is a kind of measure for the population diversity.  $g_{pbi}(x | w, z^*)$  is a composite measure of  $x$  for both diversity and convergence. The distance between  $d_1$  and  $d_2$  is controlled by the parameter  $\theta$ . In our paper, we set  $\theta = 5.0$  for empirical studies.

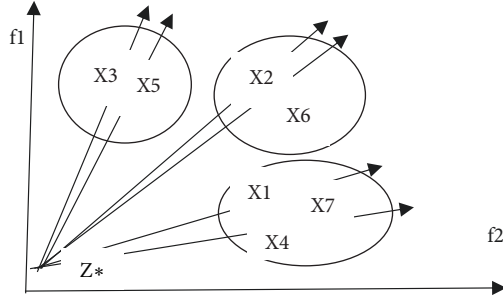


FIGURE 3: Cluster-based weight vector adjustment.

**2.5. Cluster-Based Weight Vector Adjustment.** For the MaOPs problem, this paper proposes a weighted vector adjustment method based on clustering and uses different pheromone matrices to solve different single objectives. This scalable approach can help the ant colony algorithm to better solve the MaOPs problem.

After the multiobjective problem is decomposed into  $N$  single-target subproblems by the PBI method,  $N$  ants are selected to solve these single-target subproblems. Ant  $i$  solves the subproblem  $i$ . The subproblem  $i$  is associated with  $\lambda_i$ . The corresponding objective function is  $g(x \mid \lambda_i)$ .  $N$  ants are divided into  $K$  groups according to their corresponding weight distances. Each group approaches a small area of the Pareto frontier. The same group of ants share a pheromone matrix that accommodates the location information of the learned Pareto frontier subregions. Each ant looks for a special point on the Pareto frontier for a single-target subproblem. Each ant has its own heuristic information matrix to store a priori knowledge of its subproblems. The ant  $i$ 's neighbour  $B(i)$  contains  $T$  ants whose weights are closest to  $\lambda_i$  in  $N$  weights, and  $i \in B(i)$ , and ant  $i$  is its own neighbour.

In each iteration of the algorithm, if the solution set in the dominating set remains unchanged for the first time, it means that the current weight vector can no longer meet the need of guiding population evolution, and weight adjustment is needed. In this case, nondominated populations are classified based on the clustering method, assuming that there are 7 individuals in the objective space  $(x_1, x_2, \dots, x_7)$ . They had three clusters,  $\{x_3, x_5\}$ ,  $\{x_2, x_6\}$ , and  $\{x_1, x_7, x_4\}$ . Among them,  $x_3$ ,  $x_6$ , and  $x_7$  are cluster centres, as shown in Figure 3.

### 3. Algorithm Description

**3.1. MOEA/D-ACO-PBI.** In this paper, we propose an algorithm called MOEA/D-ACO-PBI. The original MOEA/D-ACO used a weighted sum and Tchebycheff method to solve the problem. However, this is difficult; because the correlation between the algebraic and geometrical interpretation of the weight sets is not valid [15]. Recent studies conducted by [16] have shown that MOEA/D-PBI is the better method for many-objective optimization problems. Therefore, we introduce the PBI approach to MOEA/D-ACO, known as MOEA/D-ACO-PBI. The MOEA/D-ACO-PBI algorithm is presented in Algorithm 2.

- (1) Initialization
- (2) Decomposition by PBI approach
- (3) Solution construction
- (4) Update of EP
- (5) Update of pheromone
- (6) Update the solutions
- (7) Termination

ALGORITHM 2: MOEA/D-ACO-PBI.

**3.2. Initialization Procedure.** Each ant  $i_a$  has a heuristic information value  $\eta_{k,l}^{i_a}$ , each group  $j$  has pheromone trail  $\tau_{i_a,j}^k$ , and each individual solution  $x_{i_a}$  is a tour represented by permutation.

The initialization procedure of MOEA/D-ACO-PBI contains six main steps.

(1) Setting of  $N_w$  and  $\lambda^{N_w}$ : the sets of weight vectors  $\lambda_{N_w} = \{\lambda^1 \dots \lambda^{N_w}\}$  are generated by a systematic approach, developed from Das and Dennis's method [17]. Here, the size of  $\lambda$  is  $N_w$ .

$$N_w = C_{H+m-1}^{m-1} \quad (7)$$

where  $H > 0$  is the quantity of divisions considered along each objective coordinate. Therefore, a subproblem  $i_s$  has a weight vector  $\lambda^{i_s}$  that satisfies  $\sum_{l=1}^m \lambda_l^{i_s} = 1$  and  $\lambda_k^{i_s} \geq 0$ , where  $m$  is the number of objectives.

(2) Initialize the solution  $x_i$  to the subproblem  $i_s$  and set  $F_i = F(x^i)$ , for  $i = 1 \dots n$ .

(3) Set group: the number of groups is  $J$ , which is a value parameter that requires to be selected. The number of subproblems of each group is equally distributed by clustering the weights in terms of the Euclidean distance.

(4) Initialize the pheromone information matrix  $\tau_{i,j}^k = 1$ , for all  $j = 1 \dots k$  and  $k = 1 \dots n$ . The motivation is to encourage the search to focus on exploration in its early stage.

(5) Initialize  $\eta_{k,l}^i = 1 / \sum_{j=1}^m \lambda_j^i c_{k,l}^j$ , which is the heuristic information matrix to the subproblem  $i_s$ , for  $j = 1 \dots k$ .

(6) Initialize EP as the set of the entire high-quality solutions in  $\{F^1 \dots F^N\}$ ; EP is initialized to be empty.

**3.3. Solution Construction.** Assume that ant  $i_a$  is in group  $j$  and its current solution  $x_i = (x_1^i \dots x_n^i)$ . Ant  $i_a$  constructs its new solution through the following steps.

(1) The probability of choice of the link is set. For  $k, l = 1 \dots n$ , set

$$\Phi_{k,l} = \left[ \Gamma_{k,l}^{i_a} + \Delta * \ln(x^{i_a}, (k, l)) \right]^\alpha (\eta_{k,l}^{i_a})^\beta \quad (8)$$

$\Phi_{k,l}$  is representing the attractiveness of the link between the  $k$  and  $l$  to ant  $i_a$ . The indicator function in  $(x^{i_a}, (k, l))$  is equal to 1 if the link  $(k, l)$  is in tour  $x_{i_a}$  or 0 if otherwise.  $\alpha$ ,  $\beta$ , and  $\Delta$  are the three control parameters.

(2) First, ant  $i_a$  chooses an ant randomly as its starting point and builds its tour. Suppose that its current position is  $l$  and it has not completed its tour. From  $S$  (not visited so far),

TABLE 1: Test functions utilized in this experiment.

function	Formulation
CF1	$f_1(x) = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - x_1^{0.5(1.0+3(j-2)/(n-2))})^2$ $f_2(x) = 1 - x_1 + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - x_1^{0.5(1.0+3(j-2)/(n-2))})^2$
CF2	$f_1(x) = x_1 + \frac{2}{ J_1 } \sum_{j \in J_1} (x_j - \sin(6\pi x_1 + \frac{j\pi}{n}))^2$ $f_2(x) = 1 - \sqrt{x_1} + \frac{2}{ J_2 } \sum_{j \in J_2} (x_j - \cos(6\pi x_1 + \frac{j\pi}{n}))^2$
CF3	$f_1(x) = x_1 + \frac{2}{ J_1 } \left( 4 \sum_{j \in J_1} y_j^2 - 2 \prod_{j \in J_1} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2 \right)$
CF4	$f_2(x) = 1 - x_1^2 + \frac{2}{ J_2 } \left( 4 \sum_{j \in J_2} y_j^2 - 2 \prod_{j \in J_2} \cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2 \right)$ $f_1 = x_1 + \sum_{j \in J_1} h_j(y_j)$ $f_1 = 1 - x_1 + \sum_{j \in J_2} h_j(y_j)$

it selects the ant  $k$  to visit next, according to the following probability by the roulette wheel selection:

$$\frac{\Phi_{k,l}}{\sum_{s \in C} \Phi_{s,l}} \quad (9)$$

(3) If the ant has visited all the points, it returns its tour.

3.4. *Update of Pheromone.* The pheromone trail value of the link  $(k, l)$  for group  $j$  is updated as follows:

$$\tau_{k,l}^{ia} := \rho \tau_{k,l}^j + \sum_{x \in \Pi} \frac{1}{g(x | \lambda^{ia})} \quad (10)$$

$\rho$  is the persistence rate of the pheromone trail.  $\Pi$  is the set of all the new solutions; it was constructed by the ants in group  $j$  in the current iteration and was just added to EP. It contains the link between the ants'  $k$  and  $l$ .  $\tau_{\min}$  and  $\tau_{\max}$  are used to limit the range of the pheromone.  $\tau_{\min}$  is given by

$$\tau_{\min} = \varepsilon \tau_{\max} \quad (11)$$

where  $0 < \varepsilon < 1$  is a control parameter. Then, if  $\tau_{k,l}^{ia} < \tau_{\min}$  set  $\tau_{k,l}^{ia} = \tau_{\min}$  and if  $\tau_{k,l}^{ia} > \tau_{\max}$  set  $\tau_{k,l}^{ia} = \tau_{\max}$ .  $\tau_{\max}$  is set

$$\tau_{\max} = \frac{B + 1}{(1 - \rho) g_{\min}} \quad (12)$$

where  $B$  is the number of nondominated solutions found in the current iteration and  $g_{\min}$  is the smallest value obtained for the objective functions of all the  $N_s$  subproblems.

## 4. Simulation Results

In this section, we compare the MOEA/D-ACO-PBI algorithm with three state-of-the-art algorithms including MOEA/D-ACO<sup>W</sup> [13], MOEA/D [10], and Multiple Ant

Colony System (MACS) [18]. The test problems are introduced in Section 4.1. Section 4.2 describes the quality indicators. Three state-of-the-art algorithms used for comparison and the corresponding parameter settings are briefly introduced in Section 4.3.

4.1. *Test Problems.* The well-known MaOPs test function definitions of CF1–CF4 [19] are used to test the proposed algorithm. Table 1 shows the CF1–CF4 problems with the CF test suites. CF1–CF4 are test functions with biobjective problem. CF test suites are used to test the low dimensional objective function. To verify the ability of algorithms to deal with more objectives, we select C1-DTLZ1, C2-DTLZ2, and C3-DTLZ4 [20] used to test the high dimensional objective function. CF suites are popular test problem for MaOPs.

C1-DTLZ1 is a MaOPs problem; the problem is formulated as

$$C(x) = 1 - \frac{f_m(x)}{0.6} - \sum_{i=1}^{m-1} \frac{f_i(x)}{0.5} \geq 0 \quad (13)$$

C2-DTLZ2 is a MaOPs problem; the problem is formulated as

$$C(x) = -\min \left\{ \min_{i=1}^m \left[ (f_i(x) - 1)^2 + \sum_{j=1, j \neq i}^m \frac{f_j^2(x)}{j} \right. \right. \\ \left. \left. - r^2 \right], \times \left[ \sum_{i=1}^m \left( f_i(x) - \frac{1}{\sqrt{m}} \right)^2 - r^2 \right] \right\} \geq 0 \quad (14)$$

C3-DTLZ4 is a MaOPs problem; the problem is formulated as

$$C_i(x) = \frac{f_i^2(x)}{4} + \sum_{j=1, j \neq i}^m f_j^2(x) - 1 \geq 0 \quad (15)$$



TABLE 2: The average and standard deviation of GD values obtained for CF1–CF4, C1\_DTLZ1, C2\_DTLZ2, and C3\_DTLZ4 with four algorithms and different number of objectives. Best performance is in bold face.

	M	MOEA/D-ACO-PBI	MOEA/D-ACO <sup>W</sup>	MOEA/D	MACS
CF1	2	<b>1.0013e-4 (1.14e-4)</b>	1.1945e-4 (2.08e-4)	1.2524e-4 (2.12e-2)	3.4214e-2 (2.92e-2)
CF2	2	2.3325e-2 (1.05e-2)	<b>2.0382e-2 (2.59e-2)</b>	2.9194e-2 (1.24e-2)	4.5986e-2 (4.28e-2)
CF3	2	<b>2.2667e-1 (1.68e-1)</b>	3.7510e-1 (2.41e-1)	3.3477e-1 (3.19e-1)	4.2804e-1 (1.32e-1)
CF4	2	4.4754e-2 (1.29e-1)	5.4257e-2 (1.38e-1)	<b>4.2232e-2 (7.19e-2)</b>	2.1410e-1 (2.78e-1)
	3	<b>3.0152e-3 (2.13e-4)</b>	3.2859e-3 (2.36e-4)	3.4831e-3 (3.32e-4)	4.4339e-3 (1.33e-4)
C1_DTLZ1	4	<b>3.0221-3 (2.23e-4)</b>	2.5532e-2 (2.23e-3)	2.1234e-2 (1.34e-3)	2.4452e-2 (2.12e-2)
	5	<b>4.4264e-3 (3.25e-4)</b>	3.3354e-2 (2.23e-2)	5.4261e-2 (2.34e-3)	5.3634e-2 (2.84e-2)
	3	<b>7.4664e-4 (5.31e-5)</b>	2.2772e-3 (4.29e-4)	9.5259e-3 (1.67e-3)	7.9976e-3 (5.29e-3)
C2_DTLZ2	4	<b>2.7056e-3 (1.15e-4)</b>	1.1709e-2 (4.81e-3)	1.1867e-2 (1.28e-3)	2.7389e-2 (1.08e-2)
	5	<b>5.2944e-3 (2.99e-4)</b>	2.5550e-2 (1.10e-2)	2.7631e-2 (6.95e-3)	5.9090e-2 (2.84e-4)
	3	<b>4.0236e-4 (2.23e-4)</b>	6.9914e-4 (3.28e-4)	4.5365e-4 (2.65e-4)	3.514e-3 (3.26e-3)
C3_DTLZ4	4	<b>3.0885e-3 (2.32e-3)</b>	2.2035e-2 (3.82e-2)	1.3564e-2 (1.36e-2)	2.5658e-2 (1.25e-4)
	5	<b>3.0624e-3 (3.25e-4)</b>	3.6412e-2 (2.21e-2)	3.7225e-2 (5.62e-3)	5.3361e-2 (3.26e-4)

4.2. *Quality Indicators.* In our empirical study, the following two widely used quality indicators were considered. The first one reflects the convergence of an algorithm, while the second one reflects the convergence and diversity of the solutions simultaneously.

4.2.1. *Generational Distance (GD) Indicator.* For any kind of algorithm, let  $P$  be the set of final nondominated points obtained from the objective space. Let  $P^*$  be a set of points uniformly spread over the true PF. The GD can indicate only the convergence of an algorithm [21], and a smaller value indicates better quality. The GD is computed as

$$GD(P, P^*) = \frac{\sqrt{\sum_{u \in P} d(u, P^*)}}{|P|} \quad (16)$$

4.2.2. *Inverted Generational Distance (IGD) Indicator.* For any kind of algorithm, let  $P$  be the set of final nondominated points obtained from the objective space and  $P^*$  be a set of points uniformly spread over the true PF. The IGD can indicate both the convergence and diversity [21], and a smaller value indicates better quality. The IGD is computed as

$$IGD(P, P^*) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|} \quad (17)$$

4.3. *Parameter Settings.* The parameters for the four MOEAs considered in this paper are listed as follows.

(1) Some parameter settings in MOEA/D-ACO-PBI: set  $T=8$ ,  $\alpha=1$ ,  $\beta=8$ ,  $\rho=0.95$ ,  $r=0.8$ , and  $\theta=5.0$ .

(2) Some parameter settings in MOEA/D-ACO<sup>W</sup> and MOEA/D:  $T=8$ ,  $\alpha=1$ ,  $\beta=8$ ,  $\rho=0.95$ , and  $r=0.8$ . The other parameters can be obtained from [15].

(3) Some parameter settings in MACS:  $\alpha=1$ ,  $\beta=2$ ,  $\rho=0.9$ ,  $\max_{\text{Factor}}=5$ , and  $\text{Ant}_{\text{Num}}=10$ .

(4) Number of runs and termination condition: all algorithms were programmed in PlatEMO [22], and each

algorithm was independently run 30 times for each test instance and stopped after 300 generations. All the algorithms were executed on a desktop PC with a 2.1-GHz CPU, 8-GB RAM, and Windows 10.

## 5. Results and Discussion

The GD indicator was utilized to compare the convergence capacity between the proposed MOEA/D-ACO-PBI algorithm and three other algorithms (Figure 4). Table 2 shows the average and standard deviation of the GD values obtained by the four algorithms for test suits with different number of objectives. Table 2 shows that the MOEA/D-ACO-PBI algorithm could perform well on all the test functions, especially on CF1 and CF3 problems. When the number of objective test functions is higher, the performance of MOEA/D-ACO-PBI algorithm is better. The results of comparison of the MOEA/D-ACO-PBI algorithm with the other three MOEAs in terms of IGD values, using the test suite, are provided in Table 3. It shows the average IGD values for the four MOEAs over 30 independent runs and the best average and standard deviation values. From the experimental results of the test problem, the MOEA/D-ACO-PBI algorithm performs on CF2, CF4, C1-DTLZ1, C2-DTLZ2, and C3-DTLZ4 test problems. It is shown that the MOEA/D-ACO-PBI algorithm has better performance of convergence and diversity. In addition, performance of MOEA/D-ACO<sup>W</sup> algorithm is also better, and MACS performance is the worst.

The conclusions are given below.

(1) MOEA/D-ACO-PBI has the best convergence performance, as its approximate fronts cover most of those returned by the other algorithms. MOEA/D-ACO<sup>W</sup> can obtain well-spread approximately fronts; however, some of its solutions were dominated by MOEA/D-ACO-PBI.

(2) It can be observed in Figure 3 that MOEA/D-ACO-PBI dominates in most solutions and that the advantage of MOEA/D-ACO-PBI is more pronounced as the number of objectives increases.

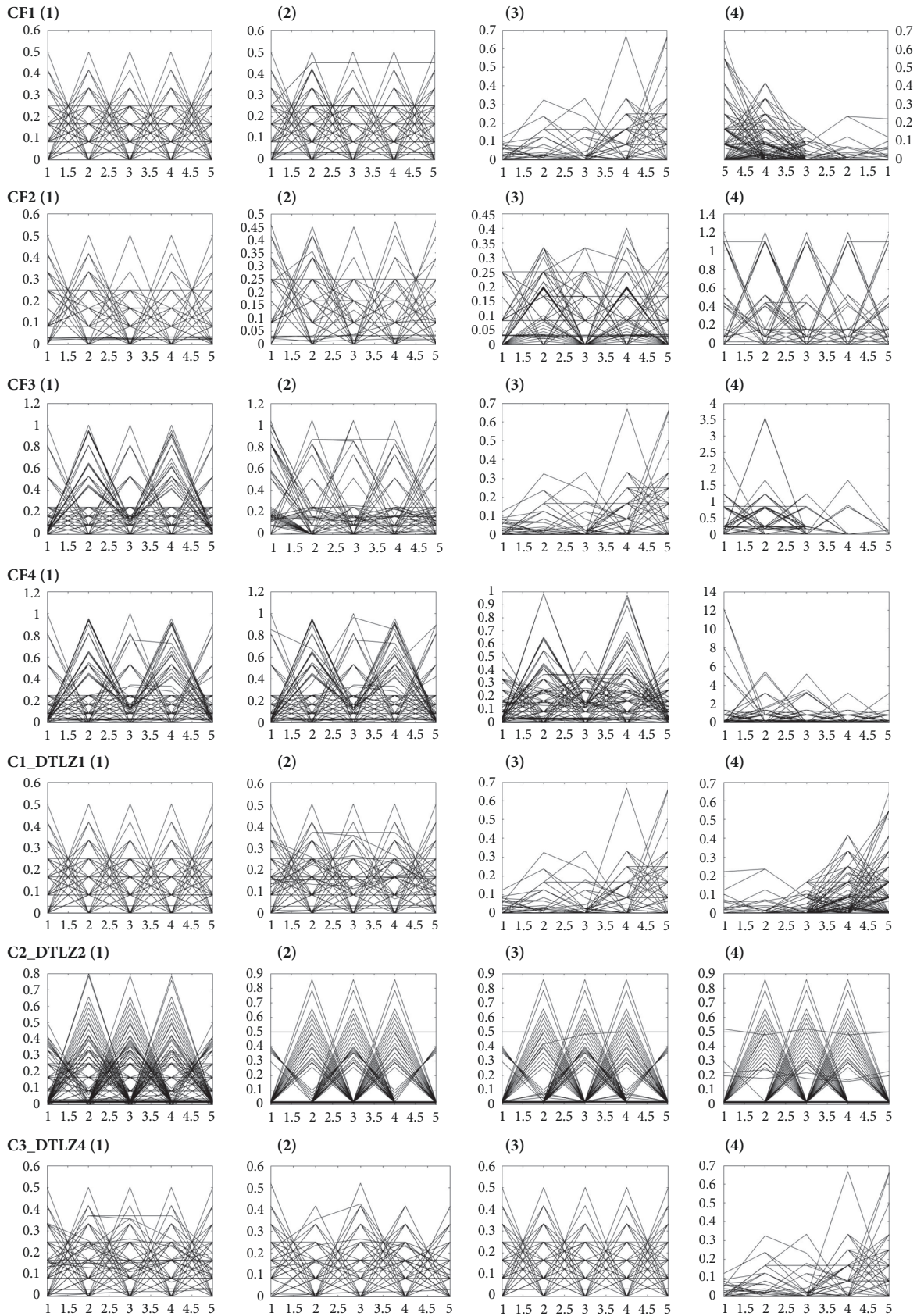


FIGURE 4: Parallel coordinates figure by four algorithms on the four-objective CF1, CF2, and CF3. (1) MOEA/D-ACO-PBI, (2) MOEA/D-ACO<sup>W</sup>, and (3) MOEA/D (4) MACS.

TABLE 3: The average and standard deviation of IGD values obtained for CF1–CF4, C1\_DTLZ1, C2\_DTLZ2, and C3\_DTLZ4 with four algorithms and different number of objectives. Best performance is in bold face.

	M	MOEA/D-ACO-PBI	MOEA/D-ACO <sup>W</sup>	MOEA/D	MACS
CF1	2	6.4371e-1 (1.36e-1)	<b>5.1532e-1 (1.27e-1)</b>	6.5734e-1 (1.08e-1)	7.8365e-1 (1.89e-1)
CF2	2	<b>4.5894e-2 (1.28e-2)</b>	4.8905e-2 (2.52e-2)	6.2428e2 (6.10e-2)	2.4116e-1 (3.61e-1)
CF3	2	3.7740e-1 (7.78e-2)	<b>3.1772e-1 (9.32e-2)</b>	3.7877e-1 (1.77e-1)	3.7581e-1 (2.54e-1)
CF4	2	<b>1.6256e-1 (6.34e-2)</b>	2.4364e-1 (1.18e-1)	1.9728e-1 (7.17e-2)	2.4654e-1 (1.32e-1)
C1_DTLZ1	4	<b>2.2653e-1 (3.25e-4)</b>	3.4629e-1 (2.36e-2)	2.5698e-1 3.26e-3)	3.7241e-1 (3.21e-4)
	5	<b>3.1651e-1 (4.13e-4)</b>	3.6234e-1 (1.12e-1)	3.7511e-1 (1.69e-1)	5.9283e-1 (3.87e-4)
	3	<b>5.1902e-2 (5.09e-4)</b>	6.5049e-2 (1.46e-3)	5.9751e-2 (4.85e-3)	9.3947e-2 (4.89e-4)
C2_DTLZ2	4	<b>1.5255e-1 (8.49e-4)</b>	2.3852e-1 (2.01e-2)	1.9833e-1 (5.52e-3)	4.7290e-1 (8.94e-4)
	5	<b>2.3283e-1 (4.29e-4)</b>	2.9947e-1 (1.28e-1)	2.9391e-1 (1.04e-1)	5.5282e-1 (4.94e-4)
	3	<b>5.1141e-2 (4.32e-4)</b>	7.3349e-2 (2.32e-3)	5.3365e-2 (3.32e-3)	6.5254e-2 (2.35e-4)
C3_DTLZ4	4	<b>1.1347e-1 (1.32e-4)</b>	2.5983e-1 (3.25e-2)	1.4651e-1 (2.22e-3)	2.4014e-1 (7.52e-4)
	5	<b>2.1698e-1 (3.24e-4)</b>	2.7321e-1 (2.36e-1)	2.6651e-1 (1.52e-1)	4.4652e-1 (5.32e-4)

## 6. Conclusions

The major contribution of this paper is the proposal of a MOEA/D-ACO-PBI algorithm for MaOPs. Due to the fact that the number of model optimization objectives increases, the PBI aggregate function was selected to decompose the objectives. MOEA/D-ACO-PBI algorithm can better ensure the convergence of algorithms and the balance of diversity, so as to obtain a better Pareto solution based on the simulation experiment; it could be seen that the results of the MOEA/D-ACO-PBI algorithm are significantly better than those of the other algorithms. Our future work aims at analysing further how to cope with constraint and dynamic constraint processing, and to optimize the number of objectives more effectively.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research work was supported by the national science and technology major projects (NO 2009ZX04001-111), subproject of national science and technology major (2013ZX04002-031), key laboratory of automotive power train and electronics (Hubei University of Automotive Technology, (No. ZDK1201703), and Hubei Provincial Education Department Youth Fund (No. Q20181801).

## References

- [1] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: a survey," *ACM Computing Surveys*, vol. 48, no. 1, article 13, 2015.
- [2] A. Zhou, B. Y. Qu, H. Li, S. Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: a survey of the state of the art," *Swarm Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.
- [3] C. Wang, Y. Wang, K. Wang, Y. Dong, and Y. Yang, "An Improved Hybrid Algorithm Based on Biogeography/Complex and Metropolis for Many-Objective Optimization," *Mathematical Problems in Engineering*, vol. 2017, 2017.
- [4] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.
- [5] C. Von Lüken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Computational Optimization and Applications*, vol. 58, no. 3, pp. 707–756, 2014.
- [6] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, Part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
- [7] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 269–285, 2014.
- [8] D. H. Phan and J. Suzuki, "R2 indicator based multiobjective memetic optimization for the pickup and delivery problem with time windows and demands (PDP-TW-D)," in *Proceedings of the Conference on Bioinspired Information and Communications Technologies*, vol. 22, pp. 43–50, 2014.
- [9] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.
- [10] 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.
- [11] H. Zhu, Z. He, and Y. Jia, "An improved reference point based multi-objective optimization by decomposition," *International Journal of Machine Learning and Cybernetics*, vol. 7, no. 4, pp. 581–595, 2016.
- [12] I. Alaya, C. Solnon, and K. Ghedira, "Ant Colony Optimization for Multi-Objective Optimization," in *Proceedings of the Problems*, vol. 1, pp. 450–457, 2017.
- [13] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and



- AntColony,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1845–1859, 2013.
- [14] M. Z. De Souza and A. T. R. Pozo, “A GPU implementation of MOEA/D-ACO for the multiobjective traveling salesman problem,” in *Proceedings of the 3rd Brazilian Conference on Intelligent Systems, BRACIS 2014*, pp. 324–329, Brazil, October 2014.
- [15] T. Lust and J. Teghem, “Two-phase pareto local search for the biobjective traveling salesman problem,” *Journal of Heuristics*, vol. 16, no. 3, pp. 475–510, 2010.
- [16] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [17] I. Das and J. E. Dennis, “Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems,” *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.
- [18] B. Barán and M. Schaerer, “A multiobjective ant colony system for vehicle routing problem with time windows,” in *Proceedings of the 21st IASTED International Multi-Conference on Applied Informatics*, pp. 97–102, Austria, February 2003.
- [19] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the CEC 2009 special session and competition,” Tech. Rep., University of Essex, Colchester, UK and Nanyang technological University, 2008.
- [20] M. Asafuddoula, T. Ray, and R. Sarker, “A decomposition-based evolutionary algorithm for many objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2015.
- [21] C. Wang, Y. Wang, K. Wang, Y. Yang, and Y. Tian, “An improved biogeography/complex algorithm based on decomposition for many-objective optimization,” in *Proceedings of the International Journal of Machine Learning Cybernetics (2)*, vol. 2, p. 17, 2017.
- [22] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, *Platemo, a matlab platform for evolutionary multi-objective optimization*. *arXiv preprint*, arXiv, 1701.00879, 2017.



**Hindawi**

Submit your manuscripts at  
[www.hindawi.com](http://www.hindawi.com)

