A Multiobjective Artificial Bee Colony Algorithm based on Decomposition

Guang Peng, Zhihao Shang and Katinka Wolter
Department of Mathematics and Computer Science, Free University of Berlin, Takustr. 9, Berlin, Germany

Keywords: Multiobjective Optimization, Evolutionary Computation, Decomposition, Artificial Bee Colony, Adaptive Normalization.

Abstract: This paper presents a multiobjective artificial bee colony (ABC) algorithm using the decomposition approach for improving the performance of MOEA/D (multiobjective evolutionary algorithm based on decomposition). Using a novel reproduction operator inspired by ABC, we propose MOEA/D-ABC, a new version of MOEA/D. Then, a modified Tchebycheff approach is adopted to achieve higher diversity of the solutions. Further, an adaptive normalization operator can be incorporated into MOEA/D-ABC to solve the differently scaled problems. The proposed MOEA/D-ABC is compared to several state-of-the-art algorithms on two well-known test suites. The experimental results show that MOEA/D-ABC exhibits better convergence and diversity than other MOEA/D algorithms on most instances.

1 INTRODUCTION

In many real-life applications, a decision maker needs to handle different conflicting objectives. Problems with more than one conflicting objectives are called multiobjective optimization problems (MOPs). Multiobjective evolutionary algorithms (MOEAs) have been developed for solving MOPs (Deb and Kalyanmoy, 2001). MOEA based on decomposition (MOEA/D) (Zhang and Li, 2007) is a novel MOEA framework, which decomposes a MOP into a series of scalar optimization problems. Recently, the MOEA/D framework has achieved great success and received much attention (Trivedi et al., 2017). We focus on the following three aspects of existing research studies about MOEA/D.

Research on other nature inspired meta-heuristics combined with MOEA/D is increasing. Based on the MOEA/D framework, Li and Landa-Silva (Li and Landa-Silva, 2011) incorporated simulated annealing to propose a MOEA for solving multiobjective knapsack problems. Moubayed et al. (Al Moubayed et al., 2014) adopted particle swarm optimization to develop decomposition-based multiobjective particle swarm optimizers. Ke et al. (Ke et al., 2013) proposed a MOEA using decomposition and ant colony optimization.

Decomposition approaches have also been widely studied. In the original MOEA/D (Zhang and Li, 2007) there are three decomposition methods including the weighted sum approach, the weighted Tchebycheff approach and the penalty-based boundary intersection (PBI) approach. To deal with the poor diversity control problem of the original Tchebycheff approach, Qi et al. (Qi et al., 2014) put forward a transformed Tchebycheff approach, which substitutes the weight vector by its respective "normalization inverse". The transformed Tchebycheff approach can obtain better uniformly distributed solutions compared with the original Tchebycheff approach. Hiroyuki et al. (Sato, 2015) used a nadir point to propose an inverted PBI approach for solving multiobjective maximization problems. Zhang et al. (Zhang et al., 2018) developed a modified PBI approach for MOPs with complex Pareto fronts.

Since the original MOEA/D is sensitive to the scales of objectives, some normalization operators need to be incorporated into the MOEA/D framework. In MOEA/D (Zhang and Li, 2007), a simple normalization method is used to replace the original objectives. NSGA-III (Deb and Jain, 2014) designs an achievement scalarizing function to get the extreme points to constitute a hyperplane, and uses the intercepts on each axis to normalize the objectives. Unlike NSGA-III, I-DBEA (Asafuddoula et al., 2014) adopts a corner-sort-ranking procedure to calculate the extreme points to build the hyperplane, and also uses the intercepts to normalize the objectives. Compared

Peng, G., Shang, Z. and Wolter, K.
A Multiobjective Artificial Bee Colony Algorithm based on Decomposition.
DOI: 10.5220/0008167801880195
In *Proceedings of the 11th International Joint Conference on Computational Intelligence (IJCCI 2019)*, pages 188-195
ISBN: 978-989-758-384-1

with the usual normalization method in MOEA/D, both procedures based on hyperplanes (Deb and Jain, 2014)(Asafuddoula et al., 2014) are more computationally expensive for solving the linear system of equations. Moreover, these two procedures are not used for biobjectives.

Following the above ideas, three objectives are followed in this paper: first, we want to develop other nature inspired meta-heuristics so as to adopt an artificial bee colony algorithm as the reproduction operator to improve the performance of MOEA/D. Second, we substitute the original Tchebycheff approach with a modified Tchebycheff approach for improved diversity. Then, in terms of differently scaled problems, an adaptive normalization mechanism is incorporated into the proposed algorithm. Finally, we propose a multiobjective artificial bee colony algorithm based decomposition for the different MOPs.

The rest of this paper is organized as follows. The technical background is presented in Section 2 and the details of the proposed algorithm are presented in Section 3. The performance of the proposed MOEA/D-ABC on two well-known test suites is presented and compared with other state-of-the-art MOEAs in Section 4. The final section summarizes the contributions and points to future research.

2 BACKGROUND

In this section, some basic concepts behind MOP are provided. Then, we briefly introduce the most widely used decomposition methods and the original artificial bee colony algorithm, respectively.

2.1 Multiobjective Optimization

A MOP can be defined as follows (Reyes-Sierra et al., 2006):

$$\min \ F(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T \quad \text{subject to } x \in \Omega \subseteq R^n \tag{1}$$

where Ω is the decision space and $x = (x_1, x_2, \cdots, x_n)$ is an n-dimensional decision vector; $F : \Omega \to \Theta \subseteq R^m$ denotes an m-dimensional objective vector and Θ is the objective space.

Definition 1 (Pareto Dominance). A decision vector $x^0 = \left(x_1^0, x_2^0, \cdots, x_n^0\right)$ is said to dominate another decision vector $x^1 = \left(x_1^1, x_2^1, \cdots, x_n^1\right)$, denoted by $x^0 \prec x^1$, if

$$\begin{cases} f_i\left(x^0\right) \le f_i\left(x^1\right), & \forall i \in \{1, 2, \cdots, m\} \\ f_j\left(x^0\right) < f_j\left(x^1\right), & \exists j \in \{1, 2, \cdots, m\} \end{cases} \tag{2}$$

Definition 2 (Pareto Optimal Solution). A solution vector $x^0 = \left(x_1^0, x_2^0, \cdots, x_n^0\right)$ is called a Pareto optimal solution, if $\neg \exists x^1 : x^1 \prec x^0$.

Definition 3 (Pareto Optimal Solution Set). The set of Pareto optimal solutions is defined as $PS = \left\{x^0 \middle| \neg \exists x^1 \prec x^0 \right\}$.

Definition 4 (Pareto Front). The Pareto optimal solution set in the objective space is called Pareto front, denoted $PF = \{F(x) = (f_1(x), f_2(x), \cdots, f_m(x)) \,|\, x \in PS\}$.

2.2 Decomposition Approach

MOEA/D is an efficient algorithm framework approaching the Pareto front. The weighted sum approach, the Tchebycheff approach and the PBI approach are three widely used decomposition methods in the framework. It has been proven that the weighted sum approach does not work well with nonconvex Pareto fronts.

In the Tchebycheff approach a scalar optimization problem can be stated as follows:

$$\min_{x \in \Omega} g^{te}(x | \lambda, z^*) = \min_{x \in \Omega} \max_{1 \le i \le m} \{\lambda_i \,|f_i(x) - z_i^*|\} \tag{3}$$

where $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)^T$ is a weight vector and $\sum_{i=1}^{m} \lambda_i = 1, \lambda_i \ge 0, i = 1, 2, \cdots, m$. $z^* = (z_1^*, z_2^*, \cdots, z_m^*)^T$ is the reference point. Because it is often time-consuming to compute the exact z_i^*, it is estimated by the minimum objective value f_i (i.e., $z_i^* = \min\{f_i(x) \,|\, x \in \Omega\}, i = 1, 2, \cdots, m$).

A scalar optimization problem of the PBI approach is defined as follows:

$$\min_{x \in \Omega} \ g^{pbi}(x | \lambda, z^*) = \min_{x \in \Omega} \ (d_1 + \theta d_2) \tag{4}$$

where

$$\begin{cases} d_1 = \frac{\left\|(f(x) - z^*)^T \lambda\right\|}{\|\lambda\|} \\ d_2 = \left\| f(x) - \left(z^* + d_1 \frac{\lambda}{\|\lambda\|}\right) \right\| \end{cases} \tag{5}$$

Here θ is a user-predefined penalty parameter. d_1 denotes the distance of the projection of vector $(f(x) - z^*)$ along the weight vector. d_2 denotes the perpendicular distance from $f(x)$ to λ.

2.3 The Artificial Bee Colony Algorithm

The Artificial bee colony (ABC) algorithm is a population based algorithm, which is motivated by the intelligent foraging behavior of a honey bee swarm (Karaboga, 2005). The honey bee colony swarm contains three types of bees: employed bees, onlooker bees, and scout bees.

In the ABC algorithm, the number of employees and onlookers is equal to the number of food sources. The ABC algorithm first generates a randomly distributed initial population of N solutions. Then, the employed bees search the new solutions within the neighborhood in their memory. Let $X_i = \{x_{i,1}, x_{i,2}, \cdots, x_{i,n}\}$ represent the i-th solution in the swarm, where n is the dimension. Each employed bee X_i generates a new position V_i using the following formula:

$$V_{i,k} = X_{i,k} + \Phi_{i,k} \times \left(X_{i,k} - X_{j,k}\right) \tag{6}$$

where X_j is a randomly selected solution $(i \neq j), k$ is a random dimension index from the set $\{1, 2, \cdots, n\}$, and $\Phi_{i,k}$ is a random number within the range $[-1, 1]$. After generating a new candidate solution V_i, a greedy selection between V_i and X_i is used. Comparing the fitness value between V_i and X_i, the better one is adopted to update the population. Once the searching phase of the employed bees is completed, the employed bees share the food source information with the onlooker bees through waggle dances. An onlooker bee chooses a food source with a probability based on a roulette wheel selection mechanism. The probability P_i for the maximization problem is defined as follows:

$$P_i = \frac{fit_i}{\sum_j^N fit_j} \tag{7}$$

where fit_i is the fitness value of the i-th solution. The better solution often has higher probability to be chosen to reproduce the new solution using Eq. 6. If a position X_i cannot be improved through a predefined number of cycles, then it is replaced by the new solution X_i^{new} discovered by the scout bee using the following equation:

$$X_{i,k}^{new} = lb_i + rand(0,1) \times (ub_i - lb_i) \tag{8}$$

where $rand(0,1)$ is a random number in $[0,1]$. The upper and lower boundaries of the i-th dimension are lb_i and ub_i, respectively.

3 THE PROPOSED ALGORITHM

In this section we will present the details of the new algorithm proposed in this paper.

3.1 Overview

The general framework of the proposed MOEA/D-ABC is given in Algorithm 17. First, a set of uniformly distributed weight vectors $\Lambda = \left\{\lambda^1, \lambda^2, \cdots, \lambda^N\right\}$ is generated (Zhang and Li, 2007). Then, a population of N solutions $P = \{x_1, x_2, \cdots, x_N\}$ is initialized randomly, after that the reference point $z^* = (z_1^*, z_2^*, \cdots, z_m^*)^T$ is initialized. According to the generated weight vectors, the neighborhood range T of subproblem i as $B(i) = \{i_1, \cdots, i_T\}$ can be obtained by computing the Euclidean distance between all the weight vectors and finding the T closest weight vectors. Steps 7-17 are iterated until the termination criterion is met. At each iteration, for the solution x_i, the mating solutions x_k and x_l are chosen from the neighborhood $B(i)$. In MOEA/D-ABC, we use the ABC operator and polynomial mutation operator to reproduce the offspring y, which will be introduced in detail in Section 3.3. Then the new offspring is used to update the reference point and neighboring solutions. In addition, we use the modified Tchebycheff approach to determine the search direction for updating the neighboring solutions.

Algorithm 1: Framework of MOEA/D-ABC.

1 Generate a set of weight vector $\Lambda \leftarrow \left\{\lambda^1, \lambda^2, \cdots, \lambda^N\right\}$;
2 Initialize the population $P \leftarrow \{x_1, x_2, \cdots, x_N\}$;
3 Initialize the reference point $z^* \leftarrow (z_1^*, z_2^*, \cdots, z_m^*)^T$;
4 **for** $i = 1 : N$ **do**
5     $B(i) \leftarrow \{i_1, i_2, \cdots, i_T\}$, where $\lambda^{i_1}, \lambda^{i_2}, \cdots, \lambda^{i_T}$ are T closest weight vectors to λ^i ;
6 **end**
7 **while** *the termination criterion is not satisfied* **do**
8   **for** $i = 1 : N$ **do**
9         $E \leftarrow B(i)$;
10   Select an index $k \in E$ based on roulette wheel selection ;
11   Randomly select an index $l \in E$ and $l \neq k$;
12         $\bar{y} \leftarrow$ ABCOperator(x_k, x_l);
13         $y \leftarrow$ PolynomialMutationOperator$(\bar{y})$;
14   UpdateIdealPoint(y, z^*);
15   UpdateNeighborhood$(y, z^*, \Lambda, B(i))$;
16   **end**
17 **end**

3.2 Modified Tchebycheff Approach

In MOEA/D-ABC, we adopt the modified Tchebycheff approach, which is defined as follows:

$$\min_{x \in \Omega} g^{mte}(x|\lambda, z^*) = \min_{x \in \Omega} \max_{1 \le i \le m} \left\{ \frac{1}{\lambda_i} |f_i(x) - z_i{}^*| \right\} \tag{9}$$

where $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)^T$ is a weight vector and $\sum_{i=1}^{m} \lambda_i = 1, \lambda_i \geq 0, i = 1, 2, \cdots, m$. $z^* = (z_1^*, z_2^*, \cdots, z_m^*)^T$ is the reference point. It is worth noting that the modified Tchebycheff approach has two advantages (Yuan et al., 2015) over the original one in MOEA/D (Zhang and Li, 2007). First, the modified form can produce more uniformly distributed solutions with a set of uniformly spread weight vectors. Second, each weight vector corresponds to a unique solution on the Pareto front (PF). The proof can be found in Theorem 1.

Theorem 1. *Assume the straight line passing through reference point z^* with the direction vector $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T$ has a intersection with the PF, then the intersection point is the optimal solution to $\Gamma(x)$ (i.e., $\Gamma(x) = \max_{1 \leq i \leq m} \left\{ \frac{1}{\lambda_i} |f_i(x) - z_i^*| \right\}$).*

Proof. Let $F(x)$ be the intersection point with the PF, then we can have the following equality

$$\frac{f_1(x) - z_1^*}{\lambda_1} = \frac{f_2(x) - z_2^*}{\lambda_2} = \cdots = \frac{f_m(x) - z_m^*}{\lambda_m} = C \tag{10}$$

where C is a constant. Suppose $F(x)$ that is not the optimal solution to $\Gamma(x)$, then $\exists F(y)$ satisfies $\Gamma(y) < \Gamma(x)$. According to Eq. 10, $\Gamma(x) = C$. Then $\forall k \in \{1, 2, \cdots, m\}$, we have

$$\frac{f_k(y) - z_k^*}{\lambda_k} \leq \Gamma(y) < C = \frac{f_k(x) - z_k^*}{\lambda_k} \tag{11}$$

Hence, $f_k(y) < f_k(x)$. This is in contradiction with the condition that $F(x)$ is the intersection point on the PF and the supposition is invalid. □

3.3 The ABC Operator

Inspired by the ABC algorithm, we adopt the ABC operator to reproduce the offspring. For each solution x_i, one mating solution x_k is chosen based on the roulette wheel selection mechanism and another $x_l\,(l \neq k)$ is randomly selected from the neighborhood $B(i)$. To get the mating solution x_k, assuming there is a solution x_i and its associated weight vector λ^i. First, the fitness value of the solution x_i can be calculated using the following equation:

$$\Gamma(x_i) = \max_{1 \leq j \leq m} \left\{ \frac{1}{\lambda_j^i} \left| f_j(x_i) - z_j^* \right| \right\} \tag{12}$$

In this way we can obtain T fitness values $\Gamma(B(i))$ of the neighboring solutions with the same weight vector λ^i. Then the fitness value of the solution x_i can be converted in the following way:

$$\Gamma^*(x_i) = \exp\left(\frac{-\Gamma(x_i)}{\sum \Gamma(B(i))/T} \right) \tag{13}$$

According to the converted T fitness values the mating solution x_k can be determined using the roulette wheel selection mechanism. For each solution x_i, the new solution $\bar{y}$ is computed as follows:

$$\bar{y} = x_k + \Phi_i \times (x_k - x_l) \tag{14}$$

where Φ_i is a n-dimensional random vector within the range $[-1, 1]$. After using the ABC operator to obtain the new solution $\bar{y}$, we apply a polynomial mutation operator (Deb and Kalyanmoy, 2001) on $\bar{y}$ to produce a new offspring y.

3.4 Adaptive Normalization

For disparately scaled objectives the original MOEA/D sometimes cannot provide satisfying results. The normalization operators are by default incorporated into the MOEA/D framework. In recent research there are three typical normalization approaches proposed in MOEA/D (Zhang and Li, 2007), NSGA-III (Deb and Jain, 2014), and I-DBEA (Asafuddoula et al., 2014). The normalization procedures in NSGA-III and I-DBEA are similar to some extent, as both aim to find the extreme points to constitute a hyperplane. However, these two algorithms are more computationally expensive for solving the linear system of equations and sometimes result in abnormal normalization results (Yuan et al., 2014). Therefore, in this paper we select a simple and efficient way to normalize the objectives.

For a solution x_i the objective value $f_j(x_i)$ $(j = 1, 2, \cdots, m)$ can be replaced with the normalized objective value $\bar{f}_j(x_i)$ as follows:

$$\bar{f}_j(x_i) = \frac{f_j(x_i) - z_j^*}{z_j^{\max} - z_j^*} \tag{15}$$

where $z_j^{\max}$ is the maximum value of objective f_j in the current population.

3.5 Computational Complexity

For MOEA/D-ABC, the major computational costs are the iteration process in the Algorithm 17. Step 9-13 mainly need $O(mT)$ operations to calculate the modified Tchebycheff values for choosing the mating solutions based on the roulette wheel selection mechanism. Step 14 performs $O(m)$ comparisons to update the reference point. Step 15 requires $O(mT)$ computations to update the neighborhood. Thus, the overall computational complexity of MOEA/D-ABC is $O(mNT)$ in one generation. Considering the adaptive normalization operator incorporated into the MOEA/D-ABC for solving the scaled optimization problems, the computational complexity of MOEA/D-ABC will be $O\left(mN^2\right)$ in one generation since T is smaller than N.

4 EXPERIMENTAL STUDIES

In this section we compare the performance of the proposed algorithm with other state-of-the-art MOEAs for solving different MOPs.

4.1 Experiment Settings

The proposed MOEA/D-ABC is implemented in the PlatEMO framework (Tian et al., 2017). For better comparison the other algorithms are also chosen from the PlatEMO. Two well-known ZDT (Zitzler et al., 2000) and DTLZ (Deb et al., 2001) test suites are used as test instances.

In order to evaluate the performance of the proposed algorithm, we have chosen the inverse generational distance (IGD) (Veldhuizen and Lamont, 1998) as a performance metric which can reflect both convergence and diversity. Since the exact Pareto front of the test problems is known we can easily locate some uniformly targeted points in the optimal surface. Let P^* be a set of these uniformly targeted points. Let A be a set of final non-dominated solutions in the objective space, which can be obtained for each algorithm. The IGD metric is computed as follows:

$$\mathrm{IGD}\left(A, P^*\right) = \frac{1}{|P^*|}\sqrt{\sum_{i=1}^{|P^*|} \tilde{d}_i^2} \tag{16}$$

where $\tilde{d_i}$ is the Euclidean distance between the i-th member of the set P^* and its nearest member in the set A. As for the IGD metric the smaller value means the obtained solutions have better quality. For each test instance 30 independent runs are performed and mean and standard deviation of the IGD values are recorded. For all algorithms we use the solutions from the final generation to compute the performance metrics.

In the experiment the performance of MOEA/D-ABC is compared with NSGA-II (Deb et al., 2002), MOEA/D (Zhang and Li, 2007) and MOEA/D-DE (Li and Zhang, 2009). The original MOEA/D study proposes two procedures MOEA/D-TCH using the Tchebycheff and MOEA/D-PBI using the PBI approach. Table 1 presents some parameters for crossover and mutation operators used in MOEA/D-ABC, NSGA-II, MOEA/D-TCH and MOEA/D-DE.

The other additional parameters are set according to suggestions given by their original papers. The neighborhood size T is set to be 20 and the penalty parameter θ is set to 5 for MOEA/D-PBI. In MOEA/D-DE the probability δ of choosing the parent solution form the whole population is set to 0.9 and the maximum number of replaced solutions n_r is set to 2. As analyzed above MOEA/D-ABC has the obvious advantage of having less parameters.

Table 1: Parameters for crossover and mutation.

Parameters	MOEA/D-ABC	NSGA-II	MOEA/D-TCH	MOEA/D-DE
SBX probability (p_c)	-	1	1	-
Polynomial mutation probability (p_m)	$1/n$	$1/n$	$1/n$	$1/n$
Distribution index for crossover (η_c)	-	20	20	-
Distribution index for mutation (η_m)	20	20	20	20
DE operator control parameter (CR)	-	-	-	1
DE operator control parameter (F)	-	-	-	0.5

4.2 Normalized Test Problems

Initially we use the ZDT problems and the DTLZ problems (DTLZ1, DTLZ2, DTLZ3, DTLZ4) to test the performance of the respectively used algorithms. The number of variables D is set according to the original papers. Since the test problems have similar range of values for each objective they are called "normalized test problems". For all 2-objective ($m = 2$) ZDT test problems the population size N in NSGA-II and other variants of MOEA/D is set to be 100 and the number of function evaluations (FEs) is set as 30000. For all 3-objective ($m = 3$) DTLZ test problems N is set to 200 and FEs is set to 100000.

Fig. 1 shows the obtained fronts with the median value of IGD performance metric of all algorithms for ZDT4 and DTLZ1. From Fig. 1 we can observe that the proposed algorithm MOEA/D-ABC can determine the Pareto optimal solutions with better convergence and diversity. Compared with the other three algorithms MOEA/D-DE has the worst convergence for the ZDT4 problem with its many local optima. NSGA-II can determine the random non-dominated solutions in the Pareto front. Both MOEA/D-TCH and MOEA/D-DE use the Tchebycheff approach as decomposition method to obtain the similar Pareto front. MOEA/D-ABC performs much better than MOEA/D-TCH and MOEA/D-DE with regard to the diversity which illustrates that the modified Tchebycheff approach improves the diversity of MOEA/D compared with the original Tchebycheff approach. Table 2 shows that MOEA/D-ABC outperforms the other three algorithms with respect to the IGD performance metric.

4.3 Scaled Test Problems

To investigate the proposed algorithm's performance in the case of disparately scaled objectives we choose the modified ZDT1 and ZDT2 as two two-objective test instances and DTLZ1 and DTLZ2 as the two

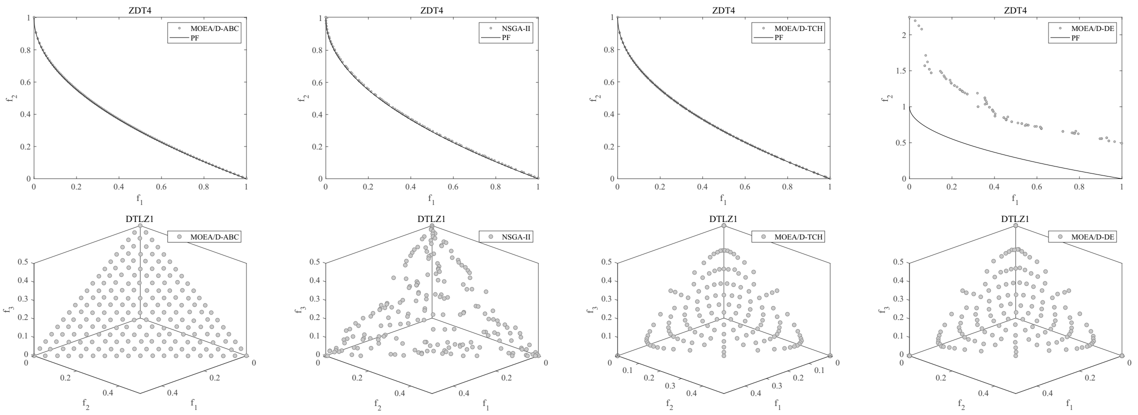

Figure 1: Obtained solutions by MOEA/D-ABC, NSGA-II, MOEA/D-TCH, and MOEA/D-DE for ZDT4 and DTLZ1.

Table 2: IGD values for MOEA/D-ABC, NSGA-II, MOEA/D-TCH, and MOEA/D-DE on ZDT and DTLZ.

Problem	N	m	D	FEs	MOEA/D-ABC	NSGA-II	MOEA/D-TCH	MOEA/D-DE
ZDT1	100	2	30	30000	4.0371e-3 (6.09e-5)	4.6043e-3 (1.84e-4)	5.8106e-3 (5.88e-3)	1.1626e-2 (5.42e-3)
ZDT2	100	2	30	30000	3.8379e-3 (2.29e-5)	4.7864e-3 (1.99e-4)	5.3845e-3 (4.66e-3)	9.4609e-3 (3.62e-3)
ZDT3	100	2	30	30000	1.0928e-2 (3.85e-2)	4.1278e-2 (5.03e-2)	1.9680e-2 (2.06e-2)	2.5511e-2 (1.52e-2)
ZDT4	100	2	10	30000	4.5511e-3 (9.93e-4)	5.4563e-3 (9.52e-4)	7.3588e-3 (4.00e-3)	1.8529e-1 (1.62e-1)
ZDT6	100	2	10	30000	3.1078e-3 (1.07e-5)	3.7673e-3 (1.14e-4)	3.1968e-3 (4.79e-5)	3.1125e-3 (1.63e-5)
DTLZ1	200	3	7	100000	1.4208e-2 (6.95e-4)	1.9097e-2 (9.01e-4)	1.9937e-2 (2.02e-5)	1.9716e-2 (5.19e-5)
DTLZ2	200	3	12	100000	3.7745e-2 (3.05e-4)	4.8807e-2 (1.49e-3)	4.9259e-2 (7.96e-5)	4.8923e-2 (2.25e-4)
DTLZ3	200	3	12	100000	4.3475e-2 (2.96e-3)	4.8337e-2 (1.22e-3)	4.8881e-2 (2.52e-4)	1.3899e-1 (4.83e-1)
DTLZ4	200	3	12	100000	4.1621e-2 (1.35e-3)	4.8543e-2 (1.33e-3)	2.7569e-1 (2.73e-1)	7.3774e-2 (6.23e-2)

three-objective test instances. The modified objective f_i is multiplied with a factor 10^{i-1}. For example, objectives f_1, f_2 and f_3 for the three-objective scaled DTLZ1 problem are multiplied with 10^0, 10^1 and 10^2, respectively.

To handle the differently scaled test problems, we incorporate the adaptive normalization operator presented in Section 3.4 into the proposed MOEA/D-ABC. The original MOEA/D-TCH with and without normalization procedure is also used to compare the performance. For clarity, we denote the MOEA/D-ABC using the normalization procedure as MOEA/D-ABC-N, MOEA/D-TCH with normalization procedure as MOEA/D-TCH-N, respectively. Fig. 2 shows the distribution of obtained solutions for MOEA/D-ABC-N, MOEA/D-TCH-N and MOEA/D-TCH on scaled ZDT1 and DTLZ1. It is clear that the normalization operator can greatly improve the performance for handling the scaled problems. Both MOEA/D-ABC-N and MOEA/D-TCH-N can obtain better distributed solutions than MOEA/D-TCH with regard to two-objective test instances. MOEA/D-TCH is not able to handle the differently scaled DTLZ1 without normalization. It is interesting to observe that MOEA/D-ABC-N is superior to MOEA/D-TCH-N with respect to diversity for solving three-objective test instances. The IGD performance metric values of concerning algorithms are shown in Table 3 which also verifies the efficiency and reliability of MOEA/D-ABC with normalization for solving disparately scaled objective problems.

4.4 MOEA/D-ABC vs MOEA/D-PBI

In the original MOEA/D study (Zhang and Li, 2007), MOEA/D-PBI can obtain much better distribution of solutions than NSGA-II and MOEA/D-TCH on DTLZ1 and DTLZ2 instances when setting the penalty parameter as 5. According to the experiments on normalized test problems, MOEA/D-ABC can also get good results on three-objective instances. To further compare the performance of MOEA/D-ABC and MOEA/D-PBI, we choose DTLZ5 and DTLZ6 as the test instances.

Fig. 3 shows the obtained Pareto fronts with MOEA/D-ABC and MOEA/D-PBI on DTLZ5. It is clear that MOEA/D-PBI is unable to find the convergent front with the penalty factor 5. However, MOEA/D-ABC can determine the front approaching the true Pareto front. Table 4 shows the IGD metric of the obtained solutions with MOEA/D-ABC and MOEA/D-PBI for DTLZ5 and DTLZ6 instances.

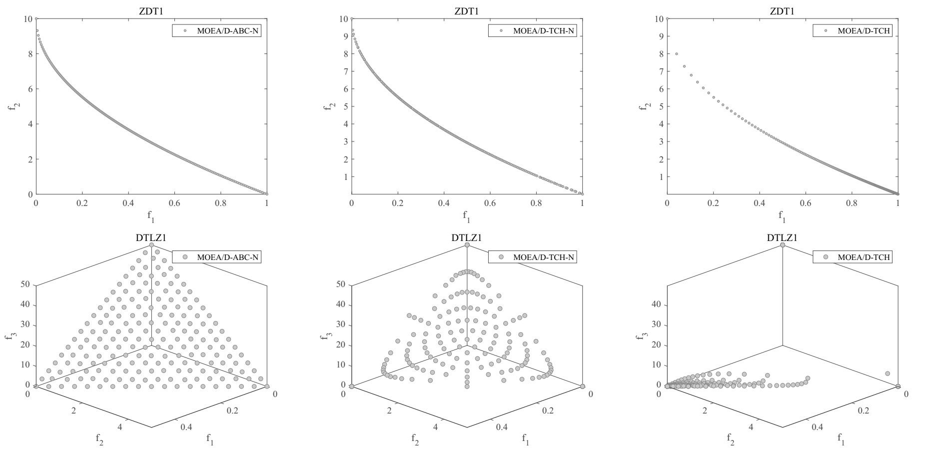

Figure 2: Obtained solutions by MOEA/D-ABC-N, MOEA/D-TCH-N, and MOEA/D-TCH for scaled ZDT1 and DTLZ1.

Table 3: IGD values for MOEA/D-ABC-N, MOEA/D-TCH-N, and MOEA/D-TCH on scaled ZDT1-2 and DTLZ1-2.

Problem	N	m	D	FEs	MOEA/D-ABC-N	MOEA/D-TCH-N	MOEA/D-TCH
ZDT1	200	2	30	100000	1.1069e-2 (3.68e-5)	1.1043e-2 (7.72e-6)	5.0087e-2(6.85e-5)
ZDT2	200	2	30	100000	1.1358e-2 (1.14e-5)	6.8675e-1 (1.42e+0)	4.0293e-2(8.80e-6)
DTLZ1	200	3	7	100000	1.2620e-1 (2.06e-2)	5.2850e-1 (5.72e-3)	9.1805e+0(8.39e-3)
DTLZ2	200	3	12	100000	3.1699e-1 (2.07e-2)	1.0600e+0 (2.27e-3)	1.5530e+1(9.49e-3)

Based on the above result analysis we see that the use of a penalty parameter cannot always obtain good results. MOEA/D-PBI requires an appropriate setting of the penalty parameter for different problems. MOEA/D-ABC is a more stable and efficient algorithm to solve different optimization problems.

5 CONCLUSIONS AND FUTURE WORK

In this paper we have developed a multiobjective artificial bee colony algorithm based on decomposition for solving multiobjective optimization problems. The proposed MOEA/D-ABC approach adopts a novel ABC operator as new reproduction operator and a modified Tchebycheff approach as new decomposition method, respectively. The above two operators are used to improve the convergence and diversity of the algorithm. Furthermore, the adaptive normalization operator is incorporated into the proposed MOEA/D-ABC for handling differently scaled problems.

In the experiment two well-known test suites and some modified scaled test instances are applied to test the performance of proposed MOEA/D-ABC and compare them with other state-of-the-art MOEAs. The test problems involve fronts that have convex, concave, disjointed, non-uniformly distributed, differently scaled, and many local fronts where an optimization algorithm can get stuck in. The proposed MOEA/D-ABC can obtain a well-converging and well-diversified set of solutions repeatedly for all problems, which shows its obvious advantage over other state-of-the-art MOEAs. Moreover, there is another advantage of MOEA/D-ABC which is that it does not require any additional parameters with respect to the reproduction operator compared with other versions of MOEA/Ds.

In the future we will study the performance of the proposed MOEA/D-ABC for solving many-objective problems with more than three objectives. It would also be interesting to study how the proposed MOEA/D-ABC performs in practice.

REFERENCES

Al Moubayed, N., Petrovski, A., and McCall, J. (2014). D2mopso: Mopso based on decomposition and dominance with archiving using crowding distance in objective and solution spaces. *Evolutionary computation*, 22(1):47–77.

Asafuddoula, M., Ray, T., and Sarker, R. (2014). A decomposition-based evolutionary algorithm for many objective optimization. *IEEE Transactions on Evolutionary Computation*, 19(3):445–460.

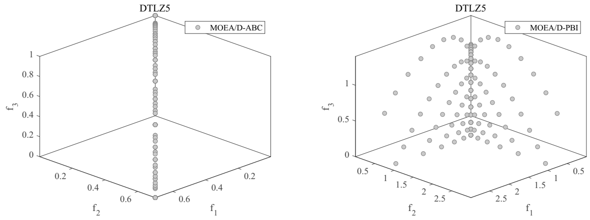

Figure 3: Obtained solutions by MOEA/D-ABC, and MOEA/D-PBI for DTLZ5.

Table 4: IGD values for MOEA/D-ABC, and MOEA/D-PBI on DTLZ5 and DTLZ6.

Problem	N	m	D	FEs	MOEA/D-ABC	MOEA/D-PBI
DTLZ5	200	3	12	100000	1.1250e-2 (2.06e-5)	2.2605e-2 (1.95e-5)
DTLZ6	200	3	12	100000	1.1319e-2 (7.43e-6)	2.2632e-2 (4.78e-6)

Deb, K. and Jain, H. (2014). 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*, 18(4):577–601.

Deb, K. and Kalyanmoy, D. (2001). *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2001). Scalable test problems for evolutionary multi-objective optimization. Technical report, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization.

Ke, L., Zhang, Q., and Battiti, R. (2013). Moea/d-aco: A multiobjective evolutionary algorithm using decomposition and antcolony. *IEEE transactions on cybernetics*, 43(6):1845–1859.

Li, H. and Landa-Silva, D. (2011). An adaptive evolutionary multi-objective approach based on simulated annealing. *Evolutionary Computation*, 19(4):561–595.

Li, H. and Zhang, Q. (2009). Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302.

Qi, Y., Ma, X., Liu, F., Jiao, L., Sun, J., and Wu, J. (2014). Moea/d with adaptive weight adjustment. *Evolutionary computation*, 22(2):231–264.

Reyes-Sierra, M., Coello, C. C., et al. (2006). Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3):287–308.

Sato, H. (2015). Analysis of inverted pbi and comparison with other scalarizing functions in decomposition based moeas. *Journal of Heuristics*, 21(6):819–849.

Tian, Y., Cheng, R., Zhang, X., and Jin, Y. (2017). Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]. *IEEE Computational Intelligence Magazine*, 12:73–87.

Trivedi, A., Srinivasan, D., Sanyal, K., and Ghosh, A. (2017). A survey of multiobjective evolutionary algorithms based on decomposition. *IEEE Transactions on Evolutionary Computation*, 21(3):440–462.

Veldhuizen, D. A. V. and Lamont, G. B. (1998). Multiobjective evolutionary algorithm research: A history and analysis.

Yuan, Y., Xu, H., and Wang, B. (2014). An improved nsga-iii procedure for evolutionary many-objective optimization. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 661–668, New York, NY, USA. ACM.

Yuan, Y., Xu, H., Wang, B., Zhang, B., and Yao, X. (2015). Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Transactions on Evolutionary Computation*, 20(2):180–198.

Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.

Zhang, Q., Zhu, W., Liao, B., Chen, X., and Cai, L. (2018). A modified pbi approach for multi-objective optimization with complex pareto fronts. *Swarm and Evolutionary Computation*, 40:216–237.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evol. Comput.*, 8(2):173–195.