Biased Multiobjective Optimization and Decomposition Algorithm

Hui Li, Qingfu Zhang, *Senior Member, IEEE*, and Jingda Deng

***Abstract*—The bias feature is a major factor that makes a multiobjective optimization problem (MOP) difficult for multiobjective evolutionary algorithms (MOEAs). To deal with this problem feature, an algorithm should carefully balance between exploration and exploitation. The decomposition-based MOEA decomposes an MOP into a number of single objective subproblems and solves them in a collaborative manner. Single objective optimizers can be easily used in this algorithm framework. Covariance matrix adaptation evolution strategy (CMA-ES) has proven to be able to strike good balance between the exploration and the exploitation of search space. This paper proposes a scheme to use both differential evolution (DE) and covariance matrix adaptation in the MOEA based on decomposition. In this scheme, single objective optimization problems are clustered into several groups. To reduce the computational overhead, only one subproblem from each group is selected to optimize by CMA-ES while other subproblems are optimized by DE. When an evolution strategy procedure meets some stopping criteria, it will be reinitialized and used for solving another subproblem in the same group. A set of new multiobjective test problems with bias features are constructed in this paper. Extensive experimental studies show that our proposed algorithm is suitable for dealing with problems with biases.**

***Index Terms*—Bias feature, covariance matrix adaptation evolution strategy (CMA-ES), decomposition, multiobjective evolutionary algorithms (MOEAs).**

I. Introduction

With the effort of more than twenty years from researchers and practitioners in different fields, multiobjective evolutionary algorithms (MOEAs) have become a major methodology for dealing with multiobjective optimization problems (MOPs) [1]–[4]. An MOEA drives a population of candidate solutions toward the Pareto front (PF) of an MOP. To design an MOEA, one should consider two highly related issues. One is how to select solutions to become parents for reproduction or enter the next generation. The other is how to generate new solutions. Based on their selection operators, most MOEAs can be classified into three classes: 1) Pareto dominance-based algorithms [5]–[8]; 2) decomposition-based ones [9]–[12]; and 3) performance indicator-based ones [13]–[16]. Although not always very easy, it is a common practice to apply or generalize reproduction operators developed in single objective evolutionary algorithms in these frameworks for dealing with MOPs.

In recent years, an MOEA based on decomposition (MOEA/D) proposed in [17] has attracted much attention in the area of MOEAs. It decomposes an MOP into a number of subproblems and solves them in a collaborative manner. Each subproblem can be either single objective or multiobjective [18]. Compared with other MOEAs, a major advantage of MOEA/D is that it is very suitable to accommodate single objective optimizers when the subproblems are single objective. In fact, some well-known single objective algorithms, including differential evolution (DE) [19], particle swarm optimization [20], simulated annealing [21], ant colony optimization [22], and efficient global optimization [23], have been used in the MOEA/D framework. More recently, some extensive work on MOEA/D for many-objective optimization have been done [24]. The archiving strategy has also been studied in MOEA/D for combinatorial optimization [25].

To design an algorithm, one has to take problem features into consideration. Various MOP problem features have been identified and studied [8], [19], [26]–[32]. Most of them are related to the geometric shapes of PF in the objective space [26]–[29] or the geometric shapes of Pareto set (PS) in the decision space [19], [30]. In [26], [28], and [29], a class of MOPs with biases on its PS have been studied. The biases on the PS mean that a tiny change on the decision variables of some Pareto solutions may cause significant changes of their objective vectors in the objective space. To deal with bias, the search operator with very powerful ability in exploitation must be considered in MOEAs. The typical examples of the benchmark multiobjective test problems with bias include DTLZ6 [28] and WFG1 [29], of which the objective functions contain the power function of decision variables. The previously reported experimental results showed these two benchmark test problems cannot be solved by any existing MOEAs. We believe that the reason is twofold: 1) the hardness of two test problems is extremely high and 2) the search

Manuscript received June 5, 2015; revised September 30, 2015; accepted November 27, 2015. Date of publication February 3, 2016; date of current version December 14, 2016. This work was supported in part by the National Natural Science Foundation of China under Grant 61175063, Grant 61573279, Grant 61473241, Grant 11131006, and Grant 91330204, in part by the National Basic Research Program of China (973 Program) under Grant 2013CB329404, and in part by the State Key Laboratory of Intelligent Control and Decision of Complex Systems under Grant 20150458. This paper was recommended by Associate Editor K.-C. Tan.

H. Li is with the School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: lihui10@mail.xjtu.edu.cn).

Q. Zhang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, on leave from the School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qingfu.zhang@cityu.edu.hk).

J. Deng is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: jingddeng2-c@my.cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2507366

operators in most existing MOEAs lack the ability for exploiting solutions with high precision. To have better understanding on the bias difficulties for MOEAs, it is also quite important to study how to generate new but challenging test problems with bias.

Covariance matrix adaptation evolution strategy (CMA-ES) [33]–[36] is one of the most successful single objective evolutionary algorithms for continuous optimization. It generates new solutions by sampling a multivariate Gaussian distribution model. CMA-ES can adaptively balance the exploration and the exploitation of search space very well. The use of CMA-ES should also be a good option for solution reproduction in MOEAs. Some efforts along this line are summarized and commented as follows.

1) In [37], MO-CMA-ES, an extension of CMA-ES for multiobjective optimization, was proposed and studied under the framework of NSGA-II. This algorithm uses Pareto dominance and a performance indicator in its selection. Each individual solution is associated with a different Gaussian model. Due to the invariant properties of CMA-ES, MO-CMA-ES was proved to be advantageous when handling the MOPs with interacting variables. Note that the update of a covariance matrix involves high computational overheads, MO-CMA-ES is not very efficient when its population size is large.
2) In [38] and [39], two similar versions of MOEA/D with CMA-ES have been suggested. To optimize all subproblems, each solution is associated with a different Gaussian model. This is the same as that in MO-CMA-ES. The experimental results showed that MOEA/D with CMA-ES is promising for solving benchmark and real-world MOPs. But these two MOEA/D variants still suffer from low efficiency when the large population size is used.

Since CMA-ES considers the adaptive strategy for controlling step size in sampling new solutions, it has the potential to exploit the optimal solutions in a small area. For this reason, CMA-ES should be promising for solving the MOPs with biases. However, sampling new solutions with only CMA-ES will cause very high computational cost as in the previous work using CMA-ES in NSGA-II or MOEA/D. To ensure the efficiency, some commonly-used reproduction operators with good ability in exploration can also be used for generating new solutions. To this end, we consider DE as the other reproduction operator in this paper since it is simple but powerful. Under the framework of MOEA/D, this paper further extends CMA-ES into multiobjective optimization, where the balance between CMA-ES and DE is highly addressed. The proposed algorithm is denoted by MOEA/D-CMA. Our major contributions of this paper are as follows.

1) *Construction of MOP Test Instances With Biases:* It has been recognized that solution distribution bias is a major factor to determine the difficulty of an MOP. However, little effort has been made to investigate it. We propose a set of MOP test instances with position-related bias and distance-related bias.
2) *Clustering Strategy for Using CMA-ES:* To reduce the computational overhead, MOEA/D-CMA does not use CMA-ES to optimize every subproblem at each generation. Instead, each Gaussian model in CMA-ES is responsible for a group of subproblems, in which only one subproblem is optimized at a time. If a CMA-ES converges on its subproblem or its Gaussian model becomes ill-conditioned, it will select another subproblem from the same group to optimize.
3) *Elite Strategy for Updating CMA-ES Models:* The current best solutions of some subproblems optimized by CMA-ES are always involved in the update of distribution mean although they are not sampled by the corresponding Gaussian models. This is beneficial for speeding up the convergence of the proposed algorithm.
4) *Experimental Studies of MOEA/D-CMA on the Proposed MOP Test Instances:* Systematical experimental studies have been conducted to study the behavior of our proposed MOEA/D-CMA on a number of MOPs with bias.

The remainder of this paper is organized as follows. Section II introduces some basic definitions in multiobjective optimization. In Section III, some existing work on the bias in MOPs are discussed, and a set of new MOP test instances with controllable bias are suggested. Section IV gives the details of the proposed algorithm. Section V presents and analyzes the experimental results. The algorithmic behavior of the proposed algorithm is studied in Section VI. The final section concludes the paper.

II. Basic Definitions

This paper considers the following continuous MOP:

$$\text{minimize}_{x\in\Omega} \quad F(x) = (f_1(x), \ldots, f_m(x))^T \tag{1}$$

where $x = (x_1, \ldots, x_n)^T$ is a vector of n decision variables in $\Omega = \prod_{j=1}^{n} [a_j, b_j]$, and $F : \Omega \to R^m$ is a vector function of m individual objective functions $f_1, \ldots, f_m$. Ω and R^m are the decision space and the objective space, respectively. Due to the conflict among different objectives, solution optimality in multiobjective optimization is often defined in terms of Pareto dominance.

Definition 1 (Dominance): Let $x, y \in \Omega$, x is said to dominate y, denoted by $x \prec y$, if $f_i(x) \le f_i(y)$ for all $i = 1, \ldots, m$ and $F(x) \neq F(y)$. $F(x)$ is said to dominate $F(y)$, denoted by $F(x) \prec F(y)$, if $x \prec y$.

Definition 2 (Pareto Optimality): A solution $x^* \in \Omega$ is said to be Pareto-optimal if no solution in Ω dominates it. The set of all Pareto-optimal solutions in Ω is called the PS. The set of all the corresponding objective vectors is called the PF.

Definition 3 (Tchebycheff Subproblem): Given a weight vector $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ with $\lambda_i \ge 0$ and $\sum_{i=1}^{m} \lambda_i = 1$, and a reference point $z = (z_1, \ldots, z_m)^T$, the objective function in a Tchebycheff subproblem of (1) for minimization is

$$g^{(\text{tch})}(x|\lambda, z) = \max_{i\in\{1,\ldots,m\}} \lambda_i |f_i(x) - z_i|. \tag{2}$$

Under certain conditions, the optimal solution of (2) is a Pareto solution to (1). Solving a set of such subproblems with appropriate and different λ settings can produce a good approximation to the PF of (1).

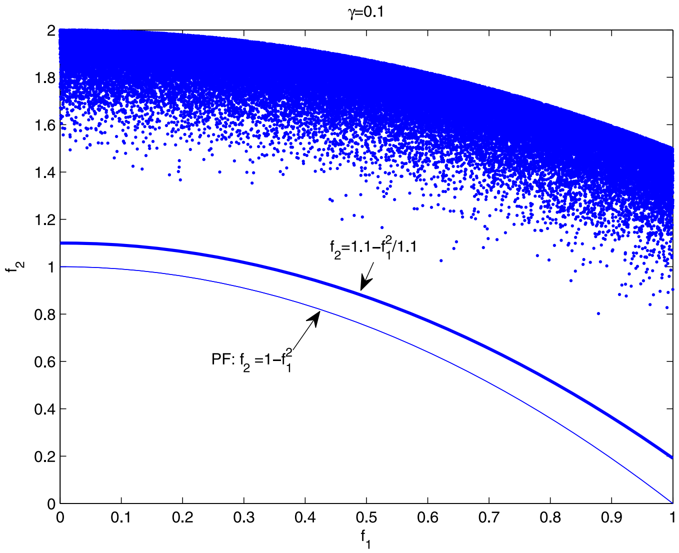


Fig. 1. Distributions of 20 000 random solutions for Deb's problem.

III. Multiobjective Optimization Problems With Bias Features

In this section, some existing MOP test problems with bias features are reviewed and discussed. Then, two methods are proposed for constructing test problems with bias features.

A. Existing MOP Test Instances With Bias Feature

As pointed in [26] and [29], the bias feature is a major factor that makes MOPs difficult for MOEAs. Bias means that the slopes of objective functions are large in the vicinities of some Pareto solutions [29]. Many real world MOPs should have this property. The position-related bias means that a small change on the position-related variable of one Pareto solution will cause significant change along PF in the objective space. On the other hand, the distance-related bias means that a small variation on the distance-related variables of a Pareto solution will cause significant deterioration on its closeness toward the PF.

To study how to deal with the bias, one needs a set of test instances with controllable biases for conducting experimental studies. The following bi-objective problem was studied in [26]:

$$f_1(x) = x_1$$
$$f_2(x) = \left(1.0 + x_2^{\gamma}\right)\left[1 - \left(\frac{f_1(x)}{1.0 + x_2^{\gamma}}\right)^2\right] \qquad (3)$$

where $x_1 \in [0, 1]$ is the position-related variable and $x_2 \in [0, 1]$ is the distance-related variable. Its PF is $f_2 = 1 - f_1^2$. The difficulty of (3) in terms of convergence is mainly caused by the term x_2^{γ}, i.e., the power function of x_2. To visualize its solution distribution in the objective space, the objective vectors of 20 000 uniformly randomly selected solutions in the search space (i.e., $[0, 1]^2$) are plotted in Fig. 1 for $\gamma = 0.1$. It is obvious that these solutions are far away from the PF. This is due to the fact that a solution close to the PF must have an extremely small x_2 value. For example, if we want $f_2 < 1.1 - f_1^2/1.1$, then x_2 must be smaller than 10^{-10}. Most crossover and mutation operators are unable to work in such precision. The well-known benchmark test problem with distance-related bias constructed by using power function is DTLZ6 [28].

In [29], the similar transformations are applied to generate bias on both distance-related variables and position-related variables in the WFG test suite. Taking WFG1 as an example, its position or distance function contains the term $\bar{x}_j^{0.02}$, where $\bar{x}_j$ is the underlying non-negative variable. When the value of $\bar{x}_j$ changes from the optimal value 0 to 10^{-50}, the corresponding distance function value is significantly increased by 0.1. It should be pointed out that most crossover and mutation operators used in MOEAs lack the ability of refining a solution with such high precision.

As far as we know, both DTLZ6 and WFG1 have not be well optimized by any existing MOEAs. Due to the unsovable hardness of these two benchmark test problems, very little progress has been made on the development of MOEAs for the MOPs with biased search space. Note that the bias hardness in the formulation (3) can be controlled by the parameter γ. However, it is not easy to construct a test problem with suitable hardness to study the search behavior of MOEAs when dealing with biased search space. This is because the benchmark test problems generated by the method formulated in (3) are either too easy or extremely hard for MOEAs. Therefore, the design of new challenging but solvable test problems with bias becomes quite important.

B. New Multiobjective Test Problems With Bias Feature

To facilitate the study of the ability of MOEAs for dealing with biases, we generalize our work in [19] and propose the following way for constructing multiobjective test problems:

$$f_k(x) = \alpha_k\left(\hat{x}_{1:m-1}\right) + \sum_{j\in I_k\subset\{m:n\}} \hat{x}_j, \; k = 1, \ldots, m \qquad (4)$$

where:

1) $\hat{x}_{1:m-1} = (\hat{x}_1, \ldots, \hat{x}_{m-1})^T \in R^{m-1}$ is an intermediate position-related vector, and $\alpha_k, k = 1, \ldots, m-1$ are the position functions, which define the PF geometric shape;
2) $\hat{x}_{m:n} = (\hat{x}_m, \ldots, \hat{x}_n)^T \in R^{n-m+1}$ is an intermediate non-negative distance-related vector, and $\beta_k(\hat{x}_{I_k}) = \sum_{j\in I_k} \hat{x}_j, k = 1, \ldots, m$, is the distance function of $f_k(x)$ with $I_k = \{j \in \{m : n\} | \mathrm{mod}(j, m) = k - 1\}$;
3) Each component $\hat{x}_j$ is defined as

$$\hat{x}_j = \begin{cases} S\left(x_j|\gamma\right) & j = 1, \ldots, m-1 \\ D\left(g\left(x_j, x_{1:m-1}\right)|\theta\right) & j = m, \ldots, n \end{cases} \qquad (5)$$

with:

a) the position-related transformation S is a one-to-one mapping from $[a_j, b_j]$ into itself. The parameter γ is used to control position-related bias;
b) the distance-related transformation D maps R into $[0, +\infty)$. The function value of D equals to zero if and only if $g(x_j, x_{1:m-1}) = 0$. The parameter θ is used to control distance-related bias.

To construct an MOP instance with bias difficulties, the position functions $\alpha_k, k = 1, \ldots, m$, the function g for defining

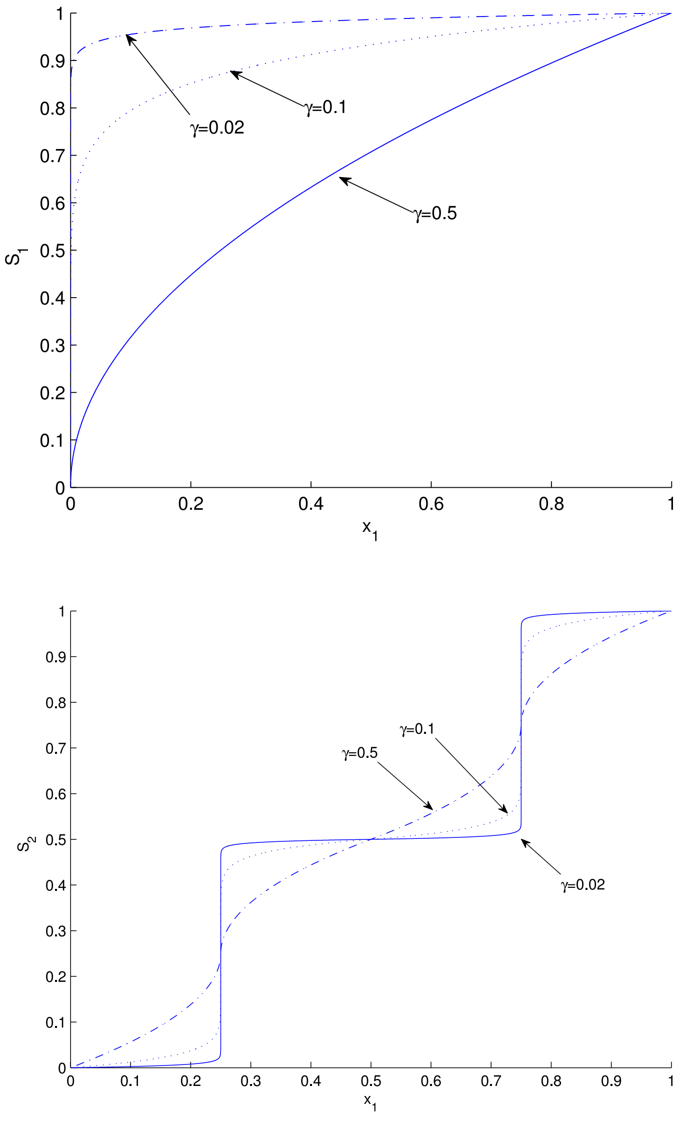


Fig. 2. Plots of S_1 (top) and S_2 (bottom) with $\gamma = 0.02, 0.1, 0.5$.

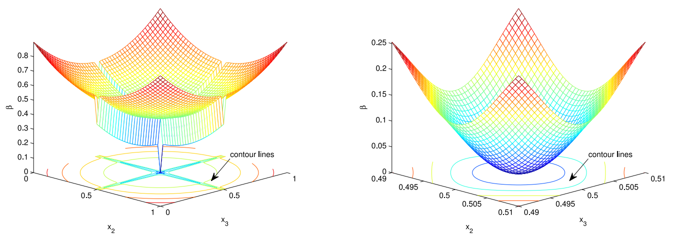


Fig. 3. Contour lines of $\beta_1(x_2, x_3|D_1)$ in $[0, 1]^2$ (left) and in $[0.5 - 10^{-2}, 0.5 + 10^{-2}]^2$ (right).

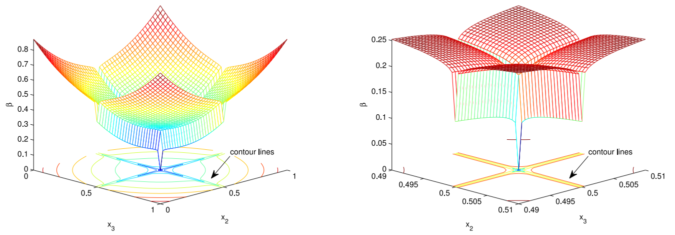


Fig. 4. Contour lines of $\beta_2(x_2, x_3|D_2)$ in $[0, 1]^2$ (left) and in $[0.5 - 10^{-2}, 0.5 + 10^{-2}]^2$ (right).

the PS geometric shape, and two bias mappings S and D should be specified. In the following, we suggest two methods for specifying these functions.

1) Position-Related Bias: Similar to [29], the following two polynomial transformations are proposed to map all position-related variables x_j from [0, 1] to [0, 1], $j = 1, \dots, m-1$:

$$\hat{x}_j = S_1(x_j|\gamma) = |x_j|^\gamma \tag{6}$$

or

$$\hat{x}_j = S_2(x_j|\gamma) = \begin{cases} \dfrac{1-(1-4x_j)^\gamma}{4} & \text{if } x_j \in [0, 0.25) \\ \dfrac{1+(4x_j-1)^\gamma}{4} & \text{if } x_j \in [0.25, 0.5) \\ \dfrac{3-(3-4x_j)^\gamma}{4} & \text{if } x_j \in [0.5, 0.75) \\ \dfrac{3+(4x_j-3)^\gamma}{4} & \text{if } x_j \in [0.75, 1]. \end{cases} \tag{7}$$

Fig. 2 plots the shapes of S_1 and S_2 with various settings of γ. When $\gamma = 0.02$, a very small change on x_1 may lead to a large change on $\hat{x}_1$. For example, in the case when $\alpha_1(\hat{x}_1) = \hat{x}_1$ and $\alpha_2(\hat{x}_1) = 1 - \sqrt{\hat{x}_1}$, the approximation of the part of PF near $f_1 = 0$ will become very difficult due to the bias of S_1. More precisely, $f_1 \in [0, 0.9]$ in the PF means $x_1 \in [0, 0.0052]$, a very small subset of [0, 1]. The PF near $f_1 = 0.25$ or $f_1 = 0.75$ has position-related bias caused by S_2. Fig. 2 shows that the position-related bias decreases as γ increases.

2) Distance-Related Bias: For every component of $y = (g(x_m, x_{1:m-1}), \dots, g(x_n, x_{1:m-1}))^T$, the following two nonlinear transformations are used to generate the distance-related bias:

$$\hat{x}_j = D_1\big(g(x_j, x_{1:m-1})|\theta\big) = g^2 + \frac{1-\exp(-g^2/\theta)}{5} \tag{8}$$

$$\hat{x}_j = D_2\big(g(x_j, x_{1:m-1})|\theta\big) = g^2 + \frac{|g|^\theta}{5}. \tag{9}$$

The minimal values of D_1 and D_2 are zero when $g = 0$. To understand the distance-related bias determined by D_1 and D_2 when $m = 2$, we study the landscapes of the following two distance functions with only two distance variables x_2 and x_3:

$$\beta_1(x_2, x_3|D_1) = D_1(g(x_2)|0.001) + D_1(g(x_3)|0.001) \tag{10}$$

$$\beta_2(x_2, x_3|D_2) = D_2(g(x_2)|0.1) + D_2(g(x_3)|0.1) \tag{11}$$

where $g(x_j) = x_j - 0.5, j = 2, 3$. The minimal solutions of the distance functions in (10) and (11) are (0.5, 0.5). Fig. 3 plots the landscape of β_1 within two regions, i.e., a large area $[0, 1]^2$ on the left and a small one $[0.5 - 10^{-2}, 0.5 + 10^{-2}]^2$ on the right. In the large area, the overall landscape of β is similar to that of a quadratic function, and there is a narrow trap near (0.5, 0.5). In the small area shown on the right of Fig. 3, the local landscape of β is still similar to a convex quadratic function. Fig. 4 shows the landscape of β_2 in two regions. It can be observed that the landscape in the large area is quite similar to that of β_1 based on D_1. But the difference between them in the small area is significant. The local landscape of β_2 based on D_2 is nonconvex, and still has a trap near (0.5, 0.5). Therefore, β_2 should be more difficult to optimize than β_1. This will be verified in our experimental results in Section V.

Using (4), a set of nine new multiobjective test instances with bias, BT1–BT9, presented in Appendix A, are designed. The features of BT1–BT9 are summarized as follows.

1) BT1–BT8 are bi-objective while BT9 is a three-objective test instance. The number of decision variables is 30 in all test problems.
2) BT3 and BT4 are two test instances with position-related bias, while all other instances have no position-related bias.
3) BT2 is a test problem with distance-related bias defined by D_2 while all others have the distance-related bias defined by D_1.
4) The PF of BT5 is disconnected. BT6 and BT8 have simple nonlinear PSs while BT7 has a complicated nonlinear PS. BT8 is multimodal.

IV. MOEA/D-CMA

This section first briefly introduces CMA-ES and its extensions in multiobjective optimization. Then, the details of MOEA/D-CMA are presented.

A. Related Work

CMA-ES is one of the most successful EAs for global optimization. It does not use crossover and mutation for reproducing new solutions. Instead, a Gaussian model is maintained for sampling new solutions in the search space. Therefore, it belongs to a class of estimation of distribution algorithms [40]. In CMA-ES, the distribution mean vector, covariance matrix, and step size are updated iteratively by collecting information on a number of selected solutions with high quality.[1] As mentioned earlier, CMA-ES has an adaptive mechanism for exploring and exploiting search space. Many computational experiences have demonstrated that CMA-ES is not only a powerful global optimizer but also an excellent local optimizer for searching solutions with satisfactory precision. This is mainly due to its mechanism on step size control.

Treating CMA-ES as a solution reproduction operator, its extension into MOEAs is quite straightforward. The pioneering MOEAs integrated with CMA-ES include MO-CMA-ES [37] and MO-CMA-D [38], which are based on the frameworks of NSGA-II and MOEA/D, respectively. Both MO-CMA-ES and MO-CMA-D associate each individual with a Gaussian model. It should be pointed out that the sorting of solutions in the latter is much easier than that in the former since each subproblem in MO-CMA-D is a single objective problem. MO-CMA-D matches individuals and Gaussian models in a fixed way. In MO-CMA-ES, only surviving solutions after nondominated sorting have the chance to update the Gaussian model of its parent solution. In fact, neither MO-CMA-ES nor MO-CMA-D is efficient when the population size is large since the update of large number of covariance matrices involves much higher computational complexity than those of commonly-used crossover and mutation operators. To overcome this weakness, the number of Gaussian models should be limited in MOEAs.

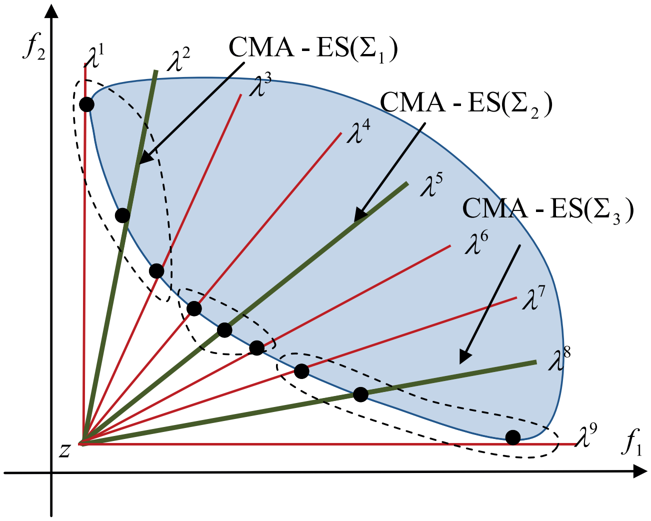


Fig. 5. Graphical illustration on the groups of subproblems in MOEA/D-CMA. In this example, nine subproblems are clustered into three groups $G_1 = \{1, 2, 3\}$, $G_2 = \{4, 5, 6\}$, and $G_3 = \{7, 8, 9\}$. Only three subproblems $s_1 = 2 \in G_1$, $s_2 = 5 \in G_2$, and $s_3 = 8 \in G_3$ are optimized by CMA-ES at a time.

B. Algorithmic Framework

MOEA/D-CMA optimizes N single optimization subproblems in a collaborative manner. In this paper, the objective function in subproblem s is defined by (2) with weight vector λ^s. Note that N Tchebycheff subproblems defined by a set of N weight vectors with even spread may not lead to N Pareto solutions with even spread along PF. To ensure good diversity of final solutions, as suggested in [41], any weight vector λ^s without zero components should be replaced by $\bar{\lambda}$, where $\bar{\lambda}_j = (1.0/\lambda_j^s)/\sum_{k=1}^{m}(1.0/\lambda_k^s)$. Compared with the weighting scheme used in the original MOEA/D, the weighting scheme adopted in this paper is clearly superior in finding the approximation of PF with good diversity when solving the MOPs with more than two objectives. Like other MOEA/D algorithms, a subproblem neighborhood concept is needed in MOEA/D-CMA. Subproblem s' is a T-neighbor of subproblem s if $\lambda^{s'}$ is among the T closest weight vectors to λ^s. MOEA/D-CMA also clusters all the subproblems into K groups. Each group of subproblems have close weight vectors. In our implementation, we employ the K-means clustering algorithm on the set of weight vectors for clustering all the subproblems. An example of clustering subproblems into three groups in MOEA/D-CMA is depicted in Fig. 5. In each group, only one subproblem is optimized by CMA-ES at every generation.

MOEA/D-CMA needs the following settings before the search.

1) *N:* The number of subproblems.
2) *T:* The neighborhood size.
3) *K:* The number of groups.
4) λ^s*:* The weight vector for subproblem s, $s = 1, \ldots, N$.
5) B_s*:* The set of indexes of neighboring subproblems to subproblem s, $s = 1, \ldots, N$.
6) G_k*:* The set of indexes of subproblem group k, $k = 1, \ldots, K$.
7) Stopping condition.

[1] The details of three major steps of a standard CMA-ES algorithm are illustrated in Appendix B.

Algorithm 1: Framework of MOEA/D-CMA

```
Input: $\lambda^1, \dots, \lambda^N$ - a set of $N$ weight vectors
Output: $WP = \{x^1, \dots, x^N\}$ - working population
Step 1: Initialization
  1.1 Initialize $WP = \{x^1, \dots, x^N\}$.
  1.2 Initialize $z = (z_1, \dots, z_m)$ by setting:
        $z_i = min_{x \in WP} f_i(x), i = 1, \dots, m.$
  1.3 Initialize $\Sigma_k$, $k = 1, \dots, K$.
Step 2: Reproduction and Update
foreach $s \in \{1, \dots, N\}$ do
    2.1 New solution reproduction and update:
    if $\exists \Sigma_k$ such that $s_k = s$ then
        $B \leftarrow B_s, X \leftarrow CMA(\Sigma_k)$
        $\Sigma_k \leftarrow UpdateCMA(x^{s_k}, X)$
    else
        $\{B, \tilde{x}\} \leftarrow DE(s)$, $X \leftarrow \{\tilde{x}\}$
    end
    2.2 Update $z$ and $WP$:
        $\{z, WP\} \leftarrow UpdatePopulation(X, B)$
end
Step 3: Termination:
if The stopping condition is met. then stop and return $WP$;
otherwise, go to Step 2
```

Algorithm 2: DE(s)

```
Step 1. Set the working neighborhood:
    $B = \begin{cases} B_s & \text{if } rand < p_s \\ \{1, \dots, N\} & \text{otherwise.} \end{cases}$
  where $rand$ is a uniform random number in [0, 1].
Step 2. Select two indexes $r_1, r_2$ from $B$ randomly and produce
  a trial solution $y \in R^n$:
    $y_j = \begin{cases} x_j^s + SF \times (x_j^{r_1} - x_j^{r_2}) & \text{if } rand < CR \\ x_j^s & \text{otherwise.} \end{cases}$
  where $SF$ is the scaling factor in DE, and $CR$ is the crossover rate.
Step 3. Mutate $y$ using the polynomial mutation [1]. For $j = 1, \dots, n$,
    $y_j \leftarrow \begin{cases} y_j + \tau_j(b_j - a_j) & \text{with probability } p_m \\ y_j & \text{with probability } 1 - p_m \end{cases}$
  with
    $\tau_j = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1 & \text{if } rand < 0.5 \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}} & \text{otherwise.} \end{cases}$
  where $p_m \in (0, 1)$ is the mutation rate, $\eta \in \mathbb{Z}^+$ is the distribution index.
Step 4. Do repairs:
    $\tilde{x}_j = \begin{cases} y_j & \text{if } a_j \leq y_j \leq b_j \\ a_j & \text{if } y_j < a_j \\ b_j & \text{if } y_j > b_j \end{cases} \quad j = 1, \dots, n.$
Step 5. Return $B$ and $\tilde{x}$.
```

Algorithm 3: CMA(Σ_k)

```
Step 1. Independently sample $\mu$ new solutions:
    $\tilde{x}^i \sim \mathcal{N}(\bar{x}^k, \sigma_k^2 C_k), i = 1, \dots, \mu$
Step 2. Repair the above solutions in the same way as in
    Algorithm 2 if necessary.
Step 3. Return $X = \{\tilde{x}^1, \dots, \tilde{x}^\mu\}$.
```

It maintains and updates the following data at each generation.

1) Working population WP $= \{x^1, \dots, x^N\}$, where x^s is the current solution to subproblem s.
2) z: The estimated reference point.
3) Data in CMA-ES for each group k

$$\Sigma_k = \left\{ s_k, \bar{x}^k, \sigma_k, C_k, p_c^k, p_\sigma^k \right\}$$

where

a) s_k is the index of the subproblem in G_k that CMA-ES optimizes;
b) $\bar{x}^k \in R^n$ is the distribution mean;
c) σ_k is the step size;
d) $C_k \in R^{n \times n}$ is the covariance matrix;
e) p_c^k and p_σ^k are evolution path vectors.

The framework of MOEA/D-CMA is given in Algorithm 1. In initialization, the working population WP is initialized with N random solutions in Ω in step 1.1, the reference point z, which is needed in computing the objective function value of each subproblem, is initialized in step 1.2 and the data for K CMA-ES procedures are initialized in step 1.3. In our implementation, $\Sigma_k = \{s_k, \bar{x}^k, \sigma_k, C_k, p_c^k, p_\sigma^k\}$ is initialized as follows.

1) s_k is a randomly selected index from G_k.
2) $\bar{x}^k = x^{s_k}$.
3) $C_k = I_{n \times n}$ (identity matrix).
4) $\sigma_k = 0.5, p_c^k = 0$, and $p_\sigma^k = 0$.

At each pass of the for-loop of step 2, one subproblem s is considered. If a CMA-ES procedure k is for subproblem s, then a set of new solutions X are generated by CMA-ES and Σ_k is updated. Otherwise, X containing only one solution is produced by DE. B produced in step 2 is the working neighborhood which will be used in step 2.2 for updating z and WP by X. In the following, we give the details of these steps.

1) *DE(s):* It is given in Algorithm 2. In step 1, the working neighborhood B is set to B_s with probability p_s or the set of all the indexes $\{1, \dots, N\}$ with probability $1 - p_s$. As suggested in [19], p_s is set to 0.9 in this paper. In step 2, two parent solutions are selected from $\{x^j | j \in B\}$ and then a DE operator is used for generating a trial solution y. Following [19], the scaling factor SF and the crossover rate CR are set to 0.5 and 1.0, respectively. To promote solution diversity, the polynomial mutation with the distribution index 20 is used with probability $1/n$ for mutating y in step 3. If y is out of the search space, step 4 will repair it.
2) *CMA(Σ_k):* It is presented in Algorithm 3. It samples μ new solutions from the Gaussian model and repairs them if necessary. The setting of μ is given in Appendix C.
3) *UpdateCMA(x^{s_k}, X):* It is given in Algorithm 4. Step 1 adds the current solution x^{s_k} into X. Step 2 uses X to update Σ_k. Step 3 updates the data in CMA-ES models.

Algorithm 4: UpdateCMA(x^{s_k}, X)

```
Step 1. Set $X = X \cup \{x^{s_k}\}$.
Step 2. Use $X$ to update $\bar{x}^k$, $p_c^k$, $p_\sigma^k$, $\sigma_k$, and $C_k$ in the same
way as the standard CMA-ES detailed in Appendix B.
Step 3. If one of the stopping criteria presented in [35] is met,
then reset $\Sigma_k$ as follows:
    3.1 Set $s_k$ to be another index randomly selected from $G_k$;
    3.2 $\bar{x}^k = x^{s_k}$;
    3.3 The other parameters in $\Sigma_k$ are reset to their initial
    values.
Step 4. Return $\Sigma_k$.
```

Algorithm 5: UpdatePopulation(X, B)

```
foreach $x \in X$ do
    Step 1. Update the reference point $z$:
    foreach $i \in \{1, \ldots, m\}$ do
        if $f_i(x) < z_i$ then $z_i = f_i(x)$.
    end
    Step 2. Update the current working population $WP$:
    Set $n_r = 0$;
    foreach $s \in B$ do
        if $g^{(tch)}(x|\lambda^s, z) < g^{(tch)}(x^s|\lambda^s, z)$ then
            $x^s = x$ and $n_r \leftarrow n_r + 1$.
        end
        if $n_r \geq 2$ then break the for-loop.
    end
end
Return $z$, $WP$
```

If one of the stopping criteria proposed in [35] for the Gaussian model is met, it implies that subproblem s_k has been well optimized by CMA-ES or the data in CMA-ES has become ill-conditioned. Then Σ_k is reinitialized and another subproblem is randomly selected from the rest of other subproblems in G_k for CMA-ES to optimize. This is very helpful for searching different parts of PF. Note that some subproblems in the same group might be selected repeatedly. But such a possibility is relatively small and has very little influence on the performance of MOEA/D-CMA.

4) *UpdatePopulation(X, B):* It is implemented in Algorithm 5. Each solution in X is used to update z and WP in steps 1–2 within the for-loop. As in [19], the maximal number of solutions in WP replaced by the same new solution is 2, which is good for population diversity.

We would like to make the following remarks on MOEA/D-CMA.

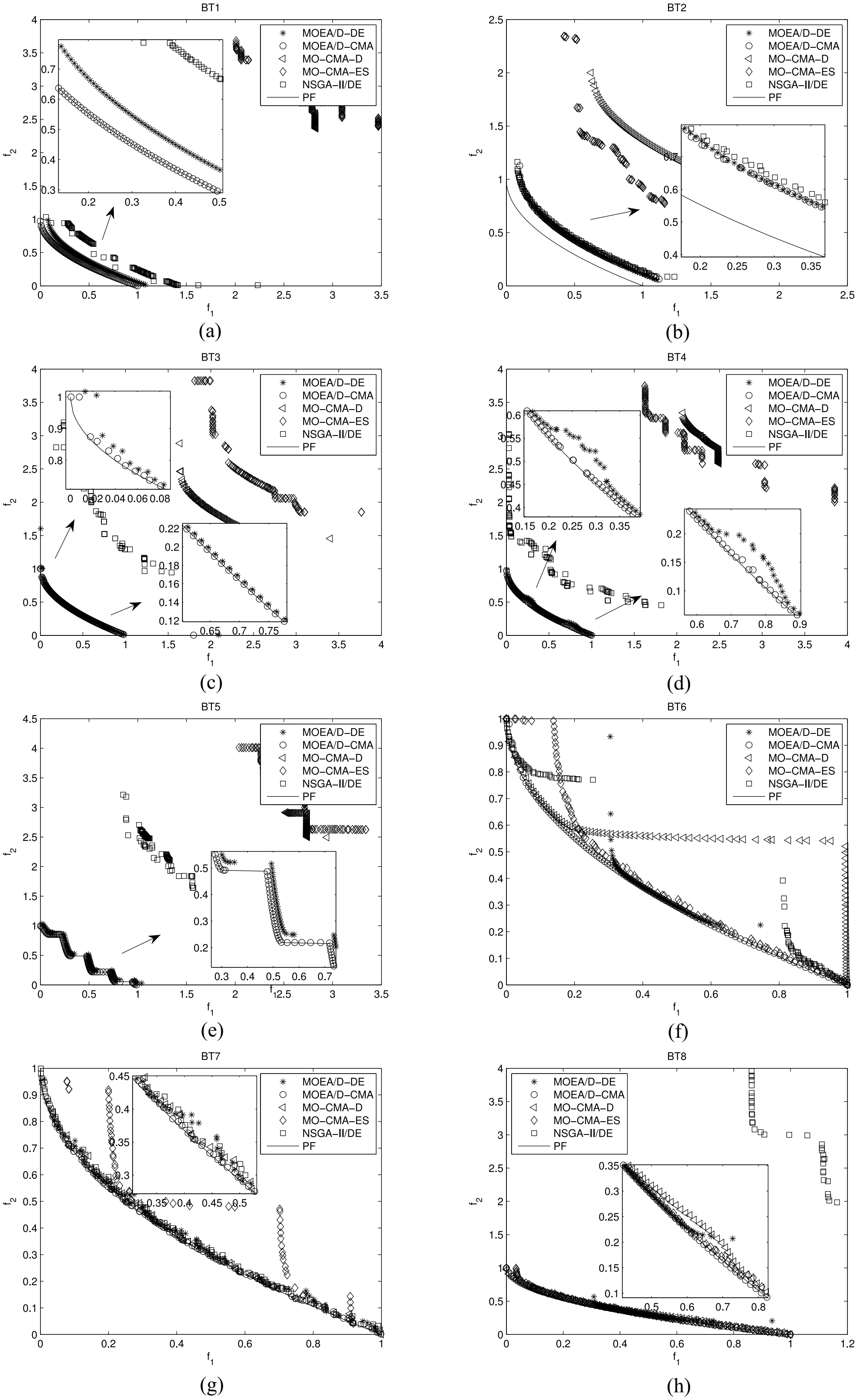
1) MOEA/D-CMA generates new solutions using two methods, i.e., DE and CMA-ES. The former is computationally cheap and fast while the latter is of high computational complexity. On one hand, DE is a very powerful tool for exploration, which is very effective for exploring the search space when the current population is far from the PF. On the other hand, CMA-ES is able to do exploitation in an efficient way. In particular, it is good at intensifying the search in the area close to PF. Thus, using both of them can make MOEA/D-CMA balance the exploration and the exploitation of search space.
2) To reduce the computational overhead, only one subproblem from each group is optimized by CMA-ES. In each generation, the number of new solutions sampled by CMA-ES models is $\mu \times K$ while $N - K$ other new solutions are produced by DE operators. When a CMA-ES meets some stopping criteria, it implies that the CMA-ES has lost its search ability. Then, it will be reinitialized for optimizing another subproblems selected from the same group. When $K = 0$, MOEA/D-CMA is the same as MOEA/D-DE. In this case, no CMA-ES is used for solution reproduction. When $K = N$, all subproblems are optimized by CMA-ES in MOEA/D-CMA. This is similar to MO-CMA-D, which involves very high computational complexity. To ensure the efficiency, a small value of K is recommended in this paper.
3) In Algorithm 4—UpdateCMA(x^{s_k}, X), both the current solution x^{s_k} of subproblem s_k and all new solutions in X produced by CMA-ES are used to update Σ_k. This is different from a standard implementation of CMA-ES, which only uses the solutions in X for updating its data. In fact, it is an elitism strategy for accelerating the convergence speed of MOEA/D-CMA. Due to the high-quality of x^{s_k} regarding subproblem s_k, it will be assigned with a large weight for the computation of distribution mean $\bar{x}_k$. As a result, $\bar{x}_k$ will move toward x^{s_k} with preference. In the following generations, the kth Gaussian model may sample new solutions close to x^{s_k} with high probability. In contrast, the other two MOEAs based on CMA-ES, i.e., MO-CMA-ES and MO-CMA-D, do not use any elitism strategy. They update the data of Gaussian model in the same way as in CMA-ES for single objective optimization, which only uses half newly-sampled solutions with the best quality for computing the distribution mean.

C. Connections Between MOEA/D-CMA and Other MOEA/D Variants With CMA-ES

Both MOEA/D-CMA and MO-CMA-D integrate CMA-ES into the MOEA/D framework for solution reproduction. The differences between them mainly lie in two aspects. First, MOEA/D-CMA only optimizes a few subproblems selected from different groups by CMA-ES in each generation while MO-CMA-D needs to optimize all subproblems by CMA-ES. Second, the subproblems in MOEA/D-CMA are optimized by either CMA-ES or DE with a certain probability. In MO-CMA-D, these operators (called self mutation and social mutation) are used for optimizing every subproblem and produce two offspring solutions individually. The convex combination of these two solutions is used to update the distribution mean vector. Note that the other MOEA/D variant with CMA-ES studied in [39] can be regarded as a special case of MOEA/D-CMA when $K = N$, i.e., each subproblem belongs to a group only including itself.

V. Experimental Studies

This section studies the performance of MOEA/D-CMA on BT1–BT9. Some discussions are also made for understanding the behavior of the proposed algorithm.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 6. Plots of nondominated solutions found by MOEA/D-DE, MOEA/D-CMA, MO-CMA-D, MO-CMA-ES, and NSGA-II/DE in the run with the minimal IGD value on BT1–BT8. The subgraphs with labels (a)–(h) are the results found by five algorithms on BT1-BT8 respectively.

A. Experimental Setting

In our experiments, three state-of-the-art EMO algorithms—MOEA/D-DE, NSGA-II, and MO-CMA-ES, and the earlier version of MOEA/D with CMA-ES, i.e., MO-CMA-D, are considered in the comparison with MOEA/D-CMA. Both NSGA-II and MOEA/D-DE use the DE operators and polynomial mutation for generating new solutions, whereas MO-CMA-ES uses CMA-ES for generating new solutions and the nondominated sorting for selection. MO-CMA-D uses CMA-ES and DE operators to produce two solutions for the same subproblem separately. In the three algorithms based on MOEA/D, the scheme for generating weight vectors is the same as in [41]. All algorithms are implemented in C++ and executed in the PC with Intel Xeon CPU (3.3 GHZ×2) and 64 GB memory running Windows 7 operating system. The parameter settings are as follows.

1) The population size of all the five algorithms is set to 100 for BT1–BT8 and 300 for BT9.
2) In the three variants of MOEA/D, i.e., MOEA/D-DE, MOEA/D-CMA, and MO-CMA-D, the neighborhood size T is set to $0.1 \times N$. The number K of Gaussian models in MOEA/D-CMA is set to 5.
3) The stopping condition is determined by the total number of function evaluations, which is 10^6 for BT1–BT6, 2×10^6 for BT7-BT8, and 5×10^6 for BT9 in each run of five algorithms.

To quantitatively measure the quality of the set A of the nondominated solutions obtained by the above-mentioned algorithms, we use the inverted generational distance (IGD) [42] defined as follows:

$$\mathrm{IGD}(A, B) = \frac{1}{|B|}\sum_{u\in B}\mathrm{dist}(u, A) \tag{12}$$

where B is a set of reference solutions, and $\mathrm{dist}(u, A) = \min_{v\in A} \|u - v\|$. The smaller the IGD value is, the better the quality of A is. The IGD indicator can measure the quality of a set of solutions in convergence and diversity simultaneously.

B. Experimental Results

Figs. 6 and 7 visualize the nondominated solutions found by MOEA/D-DE, MOEA/D-CMA, MO-CMA-D, MO-CMA-ES, and NSGA-II/DE with the minimal IGD value in 20 independent runs on BT1–BT9. From these figures, it can be observed that MOEA/D-CMA clearly outperforms the other four algorithms on all test instances except BT2, which has extremely strong distance-related bias defined by D_2 transformation. On BT2, MOEA/D-CMA, MOEA/D-DE, and NSGA-II/DE have similar performance. Among three MOEAs with CMA-ES, MOEA/D-CMA performs much better than the other two on all test instances. More observations on MOEA/D-CMA can be obtained from these figures.

1) On two instances with only distance-related bias, i.e., BT1 and BT2, the performance of MOEA/D-CMA is quite different. MOEA/D-CMA is able to approximate the PF of BT1 well shown in Fig. 6(a) while it fails to find the PF of BT2 as shown in Fig. 6(b). The distance-related bias with D_2 in BT2 is more difficult to handle than that with D_1 since D_2 is nonconvex near its optimal solution, and has very narrow attraction region. For this reason, any MOP with the distance-related bias D_2 with the value of θ in (0, 1/3] is often unsolvable for MOEAs.

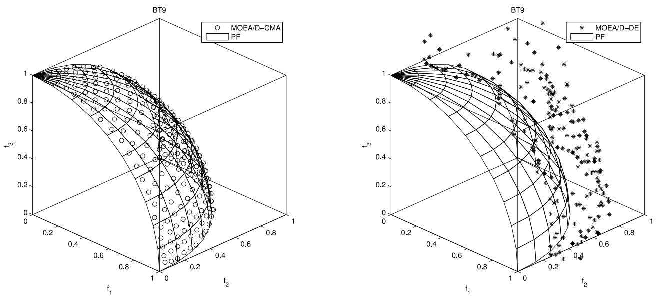


Fig. 7. Plots of nondominated solutions found by MOEA/D-CMA and MOEA/D-DE in the run with the minimal IGD value on BT9.

2) On two instances with the position-related bias, i.e., BT3 and BT4, the solutions found by MOEA/D-CMA and MOEA/D-DE shown in Fig. 6(c) and (d) are close to the PFs. But they fail to find parts of solutions of BT3 with f_1 close to 0 and those of BT4 with f_1 very close to 0.25 and 0.75. It is also evident from Fig. 6(c) and (d) that: a) MO-CMA-D, MO-CMA-ES, and NSGA-II/DE are outperformed by MOEA/D-DE and MOEA/D-CMA on BT3 and BT4 and b) MOEA/D-CMA still performs best while both MO-CMA-D and MO-CMA-ES perform worse than the other three algorithms.
3) Fig. 6(e) shows that MOEA/D-CMA is the best optimizer on BT5 with the distance-related bias D_1 and the disconnected PF. Both MO-CMA-D and MO-CMA-ES perform very badly on this instance since the nondominated solutions found by them are very far away from the PF of BT5. NSGA-II/DE performs better than MO-CMA-D and MO-CMA-ES but clearly worse than MOEA/D-DE.
4) Fig. 6(f)–(h) shows that the nondominated solutions in the best run of five algorithms on BT6, BT7, and BT8, which have the distance-related bias D_1 and the nonlinear PS. In Fig. 6(f), we can find that MOEA/D-CMA is the only algorithm that can approximate the PF of BT6 well in terms of both convergence and diversity while all other algorithms only find parts of PF on BT6. Interestingly, the performance of MO-CMA-ES is only worse than that of MOEA/D-CMA on BT6. This might be due to the good ability of CMA-ES for dealing with interacting variables. The results plotted in Fig. 6(g) indicate that all five algorithms except MO-CMA-ES can find the nondominated solutions close to the PF of BT7 with complicated nonlinear PS. Fig. 6(h) plots the results of five algorithms on BT8 with many local optima. MOEA/D-CMA still performs best. Similar to the results in Fig. 6(f), MOEA/D-DE only finds the nondominated solutions in the middle part of PF on BT8. Note that MO-CMA-D performs slightly worse than MOEA/D-CMA on BT8.
5) On the only instance with three objectives, i.e., BT9, MOEA/D-CMA, and MOEA/D-DE perform much better than the rest of other three algorithms. In Fig. 7, we plot the nondominated solutions found by these two algorithms in their best run. It is very clear that the performance of MOEA/D-CMA is superior to that of MOEA/D-DE in convergence and diversity on BT9.

TABLE I

Mean, Standard Deviation (in Round Brackets), Minimum, and Maximum (in Round Brackets) IGD Values of Nondominated Solutions in 20 Runs Found by MOEA/D-CMA, MO-CMA-D, MO-CMA-ES, MOEA/D-DE, and NSGA-II/DE on BT1–BT9. The Best Mean IGD Values Are Highlighted in Bold

Instance	MOEA/D-CMA	MO-CMA-D	MO-CMA-ES	MOEA/D-DE	NSGA-II/DE

BT1	**0.003830**(0.000001)
	0.003828(0.003833)
		3.321569(0.134307)
		3.109950(3.642563)
			3.388544(0.046863)
			3.304868(3.509714)
				0.122829(0.047713)
				0.061854(0.219144)
					0.803203(0.522896)
					0.199111(1.746078)

BT2	**0.135790**(0.017989)
	0.109831(0.170227)
		1.369935(0.102057)
		1.142027(1.510298)
			0.954360(0.074620)
			0.789589(1.124625)
				0.143712(0.014319)
				0.106601(0.166956)
					0.198683(0.052918)
					0.123319(0.339573)

BT3	**0.004814**(0.000184)
	0.004496(0.005294)
		2.643306(0.198979)
		2.157695(2.983070)
			3.066397(0.132054)
			2.813194(3.265165)
				0.012548(0.004080)
				0.006455(0.020010)
					1.638146(0.444165)
					1.002523(2.604810)

BT4	**0.007390**(0.001511)
	0.004936(0.011401)
		3.270050(0.130353)
		3.028229(3.462060)
			3.108080(0.096901)
			2.900353(3.295903)
				0.024321(0.005642)
				0.016668(0.041007)
					1.217553(0.401992)
					0.523832(1.978848)

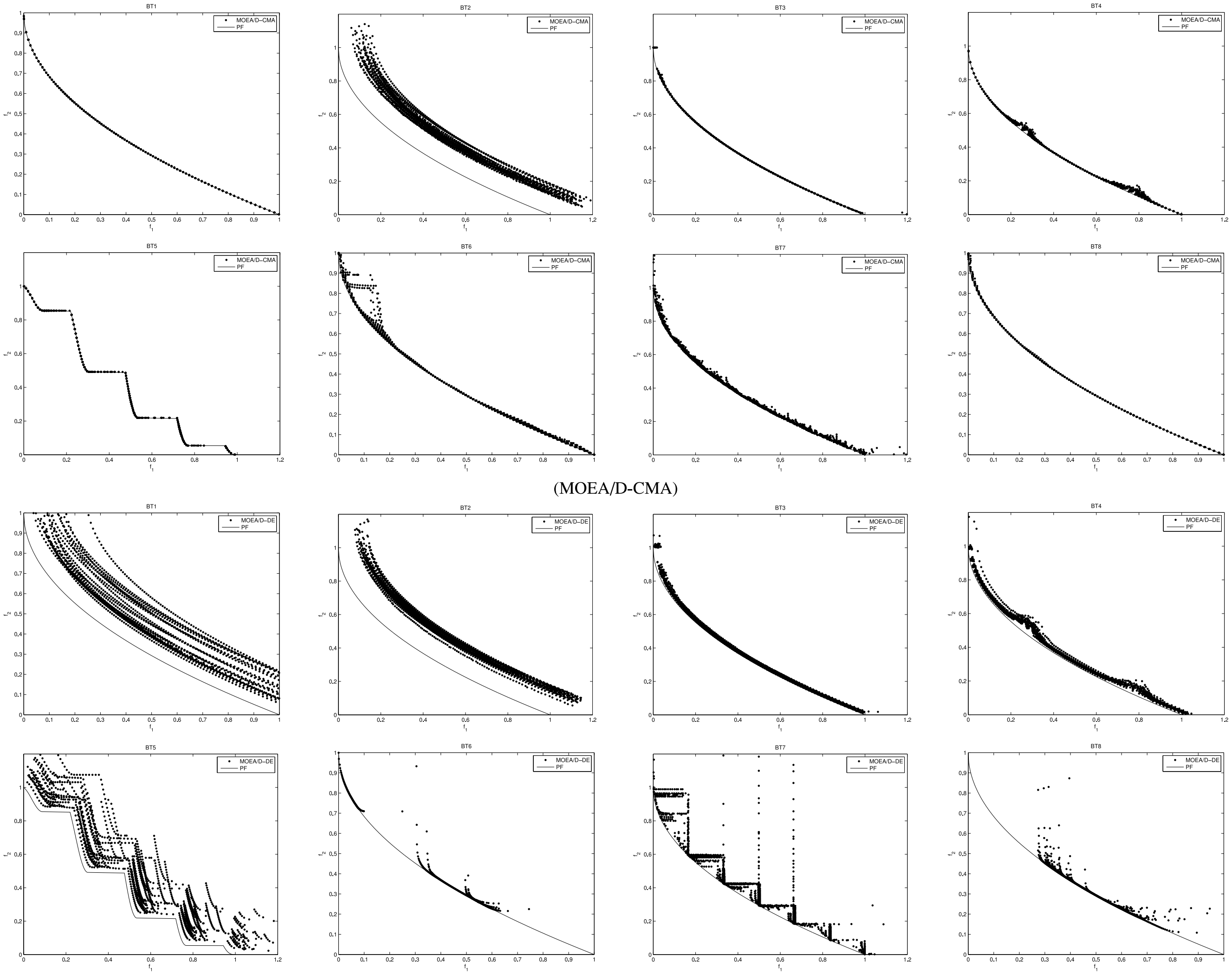
BT5	**0.004202**(0.000031)
	0.004162(0.004283)
		3.336688(0.143714)
		3.105070(3.561180)
			3.331303(0.079911)
			3.201033(3.476240)
				0.103935(0.053362)
				0.029672(0.234044)
					2.488861(0.332035)
					1.681817(2.924032)

BT6	**0.006959**(0.002934)
	0.004375(0.012361)
		0.269888(0.053371)
		0.154426(0.328650)
			0.037584(0.006942)
			0.026502(0.053674)
				0.303901(0.075625)
				0.111030(0.487422)
					0.394747(0.140035)
					0.176851(0.552407)

BT7	**0.010847**(0.003693)
	0.007059(0.019695)
		0.092205(0.090576)
		0.013060(0.339832)
			0.434812(0.290212)
			0.084045(1.049488)
				0.087937(0.062078)
				0.016291(0.242672)
					0.059789(0.064180)
					0.011967(0.325639)

BT8	**0.003907**(0.000086)
	0.003811(0.004102)
		0.072213(0.168038)
		0.010538(0.784870)
			0.480250(0.759062)
			0.009031(2.616295)
				0.127388(0.016496)
				0.101902(0.160795)
					2.835673(0.405145)
					1.816592(3.354514)

BT9	**0.793188**(1.793681)
	0.032659(5.800481)
		20.935781(1.715019)
		17.329203(23.352703)
			23.901462(1.158603)
			21.731814(25.824457)
				7.999624(4.561754)
				0.204013(12.890570)
					26.453359(0.941154)
					23.320937(27.684413)



(MOEA/D-CMA)

(MOEA/D-DE)

Fig. 8. All nondominated solutions found by MOEA/D-CMA and MOEA/D-DE in 20 runs on BT1–BT8.

6) Among three MOEAs based on CMA-ES, i.e., MOEA/D-CMA, MO-CMA-ES, and MO-CMA-D, it is quite clear that MOEA/D-CMA performs significantly better than the other two algorithms on all nine test problems. The experimental results in Fig. 6 also indicate both MO-CMA-ES and MO-CMA-D perform very poorly in convergence on the half of test problems. The success of MOEA/D-CMA on the test problems with bias is mainly

TABLE II
STATISTICS OF PERFORMANCE COMPARISONS USING *t*-TEST ON THE IGD-METRIC. "+," "≈," AND "−" DENOTE THAT THE PERFORMANCE OF CORRESPONDING ALGORITHM IS SIGNIFICANTLY BETTER THAN, SIMILAR TO, WORSE THAN THE REST OF OTHER FOUR ALGORITHMS

t-test	MOEA/D-CMA	MO-CMA-D	MO-CMA-ES	MOEA/D-DE	NSGA-II/DE
+	7	0	0	0	0
≈	2	1	0	1	0
−	0	8	9	8	9

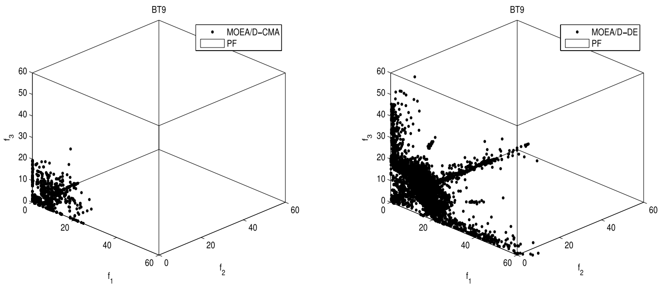


Fig. 9. All nondominated solutions found by MOEA/D-CMA (left) and MOEA/D-DE (right) in 20 runs on BT9.

due to two critical components, i.e., the use of DE for exploration and the elitism strategy.

To further analyze the performance of five algorithms in convergence and stability, we calculate the IGD values of the nondominated solutions found by each algorithm on all instances. The mean, standard deviation, minimal, and maximal IGD values of the results in 20 runs are provided in Table I. The IGD values shown in this table indicate that MOEA/D-CMA performs best among five algorithms on all instances regarding the mean, the standard deviation, the minimum, and the maximum of IGD values. In contrast, two other multiobjective algorithms with CMA-ES, i.e., MO-CMA-ES and MO-CMA-D, have the worst performance on all test instances except BT6, BT7, and BT8, which have nonlinear PSs. For BT2, the performances of MOEA/D-CMA and MOEA/D-DE are quite similar since the mean IGD values of them are about 0.14. This means that both algorithms fail to obtain a good approximation of the PF for BT2. As analyzed above, the reason is that the distance-related bias with D_2 is more challenging for MOEAs than that with D_1.

From the results in Table I, we can also notice that the overall performance of MOEA/D-DE is still better than those of MO-CMA-D, MO-CMA-ES, and NSGA-II/DE on the majority of instances. To compare the overall performance of MOEA/D-CMA and MOEA/D-DE in convergence, the nondominated solutions found by both algorithms in 20 runs are plotted in Figs. 8 and 9. These results show that MOEA/D-CMA is clearly better than MOEA/D-DE in convergence on all instance except BT2. The results on the 3-objective instance BT9 in Fig. 9 show that MOEA/D-CMA performs much more stably than MOEA/D-DE since the latter obtains quite many solutions distant to the PF of BT9. The statistics of performance comparison among five algorithms using *t*-test are provided in Table II. From this table, we can see that the performance of MOEA/D-CMA is significantly better on seven instances against all others (including MOEA/D-DE) in terms of the IGD-metric. On the other two instances, the performance of MOEA/D-CMA is similar to those of others. These results also indicate that MOEA/D-CMA is significantly worse in none of instances.

TABLE III
MEAN OF COMPUTATIONAL TIME (IN SECONDS) CONSUMED BY MOEA/D-DE(DE1), MOEA/D-CMA(CMA1), MO-CMA-D(CMA2), MO-CMA-ES(CMA3), AND NSGA-II/DE(DE2) ON BT1–BT9

Instance	DE1	CMA1	CMA2	CMA3	DE2
BT1	3.71	8.01	57.54	59.32	8.50
BT2	4.35	8.74	58.17	60.68	9.02
BT3	3.65	8.29	57.30	62.76	8.91
BT4	3.76	8.20	56.92	60.44	9.10
BT5	3.64	7.92	56.58	58.08	8.79
BT6	4.19	8.47	52.95	56.78	9.87
BT7	6.31	14.68	109.77	118.47	18.39
BT8	9.98	18.99	108.20	119.65	21.40
BT9	31.72	43.09	306.86	609.49	107.94

Moreover, the mean computational time consumed by five algorithms on BT1–BT9 in 20 runs are summarized in Table III. It can be observed that both MOEA/D-DE and NSGA-II/DE use less computational time than three algorithms based on CMA-ES. The computational time of MOEA/D-CMA is only about 2–3 times of that of MOEA/D-DE, while the computational time of MO-CMA-ES and MO-CMA-D is about 15–20 times of that of MOEA/D-DE. This result indicates that MOEA/D-CMA is more efficient than MO-CMA-D and MO-CMA-ES. The reason is that MOEA/D-CMA has the limitation on the number of Gaussian models in CMA-ES optimizer.

VI. MORE ANALYSIS ON THE ALGORITHMIC BEHAVIORS OF MOEA/D-CMA

In this section, we first conduct experiments to study the problem difficulties on the distance-related bias with various settings of θ. Then, the balance between Gaussian models and DE in MOEA/D-CMA is studied.

A. Problem Difficulties on the Distance-Related Bias

The distance-related bias of MOPs can be controlled by the parameter θ in D transformation. To study the search ability of MOEA/D-CMA on the MOPs with different levels of bias, we conducted some experiments to test MOEA/D-CMA on BT1, and BT7, with different values of θ. Note that BT1 and BT7 are the representative test problems, which have linear and nonlinear PSs, respectively. The biases in both test problems are introduced by the transformation D_1 suggested by us in this paper. All parameters used in MOEA/D-CMA remain the same as in the previous section.

The evolutions of the IGD values found by MOEA/D-CMA are shown in Fig. 10. It can be seen that the performance

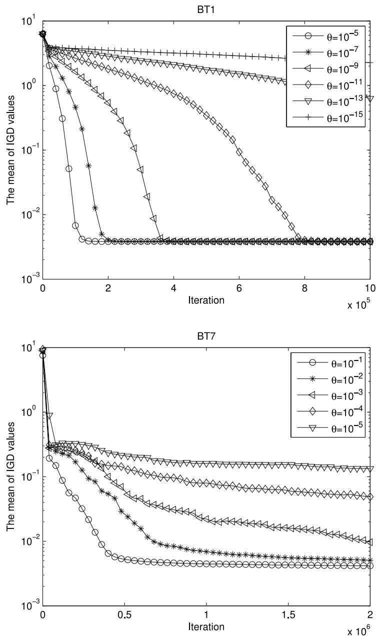


Fig. 10. IGD values found by MOEA/D-CMA with various settings of θ.

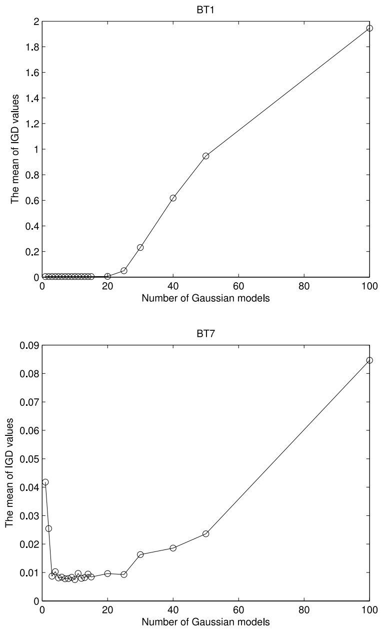


Fig. 11. IGD values found by MOEA/D-CMA with various settings of K.

of MOEA/D-CMA gets worse in convergence regarding IGD-metric on two instances as the parameter θ decreases. The results shown on the top of Fig. 10 show that MOEA/D-CMA can solve BT1 well when θ in D_1 is larger than 10^{-11}. The results shown on the bottom of Fig. 10 indicate that MOEA/D-CMA can solve BT7 well when θ in D_1 is larger than 10^{-3}. This means the instance with bias and nonlinear PS is more difficult than that with only bias.

B. Balance Between DE and CMA-ES

From the experimental results reported in Section V-B, we can see that using CMA-ES in MOEA/D as reproduction operator for exploitation is very crucial when handling MOPs with bias. In fact, the parameter K can be used to control the balance between CMA-ES and DE in MOEA/D-CMA. To study the effect of K in MOEA/D-CMA, we further conducted experiments to test the performance of MOEA/D-CMA with various settings of K on BT1 and BT7. All other parameters are the same as in the previous section. The mean IGD values obtained by MOEA/D-CMA in 20 runs are presented in Fig. 11. From this figure, it can be seen that the PF of BT1 can be well approximated by MOEA/D-CMA when K takes the integers between 1 and 20. This means that MOEA/D-CMA with only one Gaussian model can also solve BT1 well. This is because the PS of BT1 is linear and all its Pareto solutions have very similar structure in the decision space. In contrast, MOEA/D-CMA needs no less than five Gaussian models to find a good approximation of the PF of BT7, which has nonlinear PS. When the number of groups is larger than 25, the PFs of BT1 and BT7 are not well approximated by MOEA/D-CMA. In this case, the performance of MOEA/D-CMA gets worse as the value of K increases.

VII. Conclusion

In this paper, we have proposed a new variant of MOEA/D with CMA-ES, named MOEA/D-CMA, for solving MOPs with position-related bias and distance-related bias. This algorithm decomposes an MOP into a number of single objective optimization problems and solves them in a collaborative way. It divides these subproblems into several groups. At each generation, only one selected subproblem from each group is optimized by CMA-ES, while other subproblems are optimized by DE. When an evolution strategy procedure meets some stopping criteria, it will be reinitialized and used for dealing with another subproblem in the same group. To study the ability of MOEA/D-CMA, a set of new multiobjective test problems with bias features have been constructed. Extensive experimental studies have been conducted to compare MOEA/D-CMA and four other state-of-the-art multiobjective algorithms. Our experimental results have shown that MOEA/D-CMA outperforms others. This confirms that our proposed algorithm has inherited the ability for the exploration and the exploitation of search space from DE and CMA-ES. Our future work will include how to use other advanced single objective optimizers in evolutionary multiobjective optimization.

Appendix A
Multiobjective Test Instances With Bias Feature

The detailed formulations of nine multiobjective test problems with bias feature are as follows:

1) BT1

$$f_1(x) = x_1 + \sum_{j\in I_1} D_1\left(y_j \middle| 10^{-10}\right)$$

$$f_2(x) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_1\left(y_j \middle| 10^{-10}\right)$$

where $x \in [0, 1]^{30}$, $I_k = \{j| \bmod (j, 2) = k - 1, j = 2, \ldots, n\}, k = 1, 2$, $y_j = x_j - \sin(j\pi/(2n))$, and $j = 2, \ldots, n$. The PS is $\{x \in R^n | x_j = \sin(j\pi/(2n)), j = 2, \ldots, n, x_1 \in [0, 1]\}$. The PF is $\{(f_1, f_2)|, f_2 = 1 - \sqrt{f_1}, f_1 \in [0, 1]\}$.

2) BT2

$$f_1(x) = x_1 + \sum_{j\in I_1} D_2\left(y_j \Big| \frac{1}{5}\right)$$
$$f_2(x) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_2\left(y_j \Big| \frac{1}{5}\right)$$

where $x \in [0, 1]^{30}$, I_1, I_2, y_j, and $j = 2, \ldots, n$ are the same as in BT1. Its PS and PF are the same as those of BT1.

3) BT3

$$f_1(x) = S_1(x_1|0.02) + \sum_{j\in I_1} D_1\left(y_j \big| 10^{-8}\right)$$
$$f_2(x) = 1 - \sqrt{S_1(x_1|0.02)} + \sum_{j\in I_2} D_1\left(y_j \big| 10^{-8}\right)$$

where $x \in [0, 1]^{30}$, I_1, I_2, y_j, and $j = 2, \ldots, n$ are the same as in BT1. Its PS and PF are the same as those of BT1.

4) BT4

$$f_1(x) = S_2(x_1|0.06) + \sum_{j\in I_1} D_1\left(y_j \big| 10^{-8}\right)$$
$$f_2(x) = 1 - \sqrt{S_2(x_1|0.06)} + \sum_{j\in I_2} D_1\left(y_j \big| 10^{-8}\right)$$

where $x \in [0, 1]^{30}$, I_1, I_2, and $y_j, j = 2, \ldots, n$ are the same as in BT1. Its PS and PF are the same as those of BT1.

5) BT5

$$f_1(x) = x_1 + \sum_{j\in I_1} D_1\left(y_j \big| 10^{-10}\right)$$
$$f_2(x) = (1 - x_1)[1 - x_1 \sin(8.5\pi x_1)] + \sum_{j\in I_2} D_1\left(y_j \big| 10^{-10}\right)$$

where $x \in [0, 1]^{30}$, I_1, I_2, y_j, and $j = 2, \ldots, n$ are the same as in BT1. Its PS is the same as that of BT1. Its PF is the part of $\{(f_1, f_2)|f_2 = (1 - f_1)(1 - f_1 \sin(8.5\pi f_1)), f_1 \in [0, 1]\}$.

6) BT6

$$f_1(x) = x_1 + \sum_{j\in I_1} D_1\left(y_j \big| 10^{-4}\right)$$
$$f_2(x) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_1\left(y_j \big| 10^{-4}\right)$$

where $x \in [0, 1]^{30}$, I_1, and I_2 are the same as in BT1. $y_j = x_j - x_1^{0.5+1.5(j-1)/(n-1)}, j = 2, \ldots, n$. Its PS is $\{x \in R^n | x_j = x_1^{0.5+1.5(j-1)/(n-1)}, j = 2, \ldots, n, x_1 \in [0, 1]\}$. Its PF is the same as that of BT1.

7) BT7

$$f_1(x) = x_1 + \sum_{j\in I_1} D_1\left(y_j \big| 10^{-3}\right)$$
$$f_2(x) = 1 - \sqrt{x_1} + \sum_{j\in I_2} D_1\left(y_j \big| 10^{-3}\right)$$

where $x \in [0, 1] \times [-1, 1]^{29}$, I_1, and I_2 are the same as in BT1, $y_j = x_j - \sin(6\pi x_1), j = 2, \ldots, n$. Its PS is $\{x \in R^n | x_j = \sin(6\pi x_1), j = 2, \ldots, n, x_1 \in [0, 1]\}$. Its PF is the same as that of BT1.

8) BT8

$$f_1(x) = x_1 + \sum_{j\in I_1} Q\left(D_1\left(y_j \big| 10^{-3}\right)\right)$$
$$f_2(x) = 1 - \sqrt{x_1} + \sum_{j\in I_2} Q\left(D_1\left(y_j \big| 10^{-3}\right)\right)$$

where $x \in [0, 1]^{30}$, I_1, I_2 are the same as in BT1, $y_j, j = 2, \ldots, n$ is the same as in BT6, and $Q(z) = 4z^2 - \cos(8\pi z) + 1$ is a multimodal function. Its PS and PF are the same as that of BT6.

9) BT9

$$f_1(x) = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + 10\sum_{j\in I_1} D_1\left(y_j \big| 10^{-9}\right)$$
$$f_2(x) = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + 10\sum_{j\in I_2} D_1\left(y_j \big| 10^{-9}\right)$$
$$f_3(x) = \sin(0.5x_1\pi) + 10\sum_{j\in I_3} D_1\left(y_j \big| 10^{-9}\right)$$

where $x \in [0, 1]^{30}$, $I_k = \{j|\bmod(j, 3) = k - 1, j = 3, \ldots, n\}$, $k = 1, 2, 3$. $y_j = x_j - \sin(j\pi/(2n))$, and $j = 3, \ldots, n$. Its PS is $\{x \in R^n | x_j = \sin(j\pi/(2n)), j = 3, \ldots, n, x_1 \in [0, 1], x_2 \in [0, 1]\}$. Its PF is $\{(f_1, f_2, f_3)|, f_1^2 + f_2^2 + f_3^2 = 1, f_i \in [0, 1], i = 1, 2, 3\}$.

Appendix B
Major Steps in Standard CMA-ES

The major steps in standard CMA-ES are illustrated in the following:

1) *Step 1:* Sampling and sorting

$$\tilde{x}^i \leftarrow \bar{x} + \sigma\mathcal{N}(0, C) \sim \mathcal{N}\left(\bar{x}, \sigma^2 C\right), i = 1, \ldots, \mu. \tag{13}$$

2) *Step 2:* Update of distribution mean

$$y^{(k)} = \left(\tilde{x}^{k:(\mu+1)} - \bar{x}\right)\Big/\sigma, k = 1, \ldots, \mu'$$
$$y_w = \sum_{k=1}^{\mu'}\left(w_k \times y^{(k)}\right), \bar{x} \leftarrow \bar{x} + \sigma y_w \tag{14}$$

where $\tilde{x}^{k:(\mu+1)}$ is the kth best solution among $\{x^{s_k}\} \cup \{\tilde{x}^i | i = 1, \ldots, \mu\}$, $w_k > 0, k = 1, \ldots, \mu'$ is the weighted coefficient and $\sum_{k=1}^{\mu'} w_k = 1$. The setting of w_k is given in Appendix C.

3) *Step 3:* Update of evolution paths

$$\begin{aligned} p_\sigma &\leftarrow (1-c_\sigma)p_\sigma + \sqrt{c_\sigma(2-c_\sigma)\mu_{\text{eff}}}\,C^{-\frac{1}{2}}y_w \\ p_c &\leftarrow (1-c_c)p_c + h_\sigma\sqrt{c_c(2-c_c)\mu_{\text{eff}}}\,y_w. \end{aligned} \tag{15}$$

4) *Step 4:* Update of step size and covariance matrix

$$\begin{aligned} \sigma &\leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma} \times \left(\frac{\|p_\sigma\|}{\mathbb{E}\|\mathcal{N}(0,\mathbf{I})\|} - 1\right)\right) \\ C &\leftarrow \left(1-c_1-c_\mu\right)C + c_1\left(p_c {p_c}^T + \delta(h_\sigma)C\right) \\ &\quad + c_\mu \sum_{k=1}^{\mu} w_k y^{(k)} {y^{(k)}}^T \end{aligned} \tag{16}$$

where $\mathbb{E}\|\mathcal{N}(0,\mathbf{I})\| = \sqrt{2}\Gamma(((n+1)/2))/\Gamma((n/2)) \approx \sqrt{n} + \mathcal{O}(1/n)$ is the expectation of Euclidean norm of a distributed random vector $\mathcal{N}(0,\mathbf{I})$. $\mathbf{I} \in R^{n\times n}$ is an identity matrix, and $\Gamma(\cdot)$ is a gamma distribution.

Appendix C
Parameter Settings in CMA-ES [35]

In MOEA/D-CMA, the parameter settings used in CMA-ES are the same as in [35], which consist of the following three parts:

1) Sampling and selection

$$\mu = 4 + \lfloor 3\ln n \rfloor,\ \mu' = \left\lfloor \frac{\mu}{2} \right\rfloor$$

$$w_i = \frac{\ln\frac{\mu+1}{2} - \ln i}{\sum_{j=1}^{\mu'}\left(\ln\frac{\mu+1}{2} - \ln i\right)} \text{ for } i = 1, \ldots, \mu'.$$

2) Control of step size

$$\begin{aligned} \mu_{\text{eff}} &= \frac{1}{\sum_{i=1}^{\mu'} w_i^2},\ c_\sigma = \frac{\mu_{\text{eff}}+2}{n+\mu_{\text{eff}}+5} \\ d_\sigma &= 1 + 2\max\left(0, \sqrt{\frac{\mu_{\text{eff}}-1}{n+1}} - 1\right) + c_\sigma. \end{aligned}$$

3) Covariance

$$\begin{aligned} c_c &= \frac{4+\mu_{\text{eff}}/n}{n+4+2\mu_{\text{eff}}/n},\ c_1 = \frac{2}{(n+1.3)^2+\mu_{\text{eff}}} \\ c_\mu &= \min\left(1-c_1, \frac{\mu_{\text{eff}}-2+1/\mu_{\text{eff}}}{(n+2)^2+\mu_{\text{eff}}}\right) \\ h_\sigma &= \begin{cases} 1 & \text{if } \dfrac{\|p_\sigma\|}{\sqrt{1-(1-c_\sigma)^{2(\text{gen}+1)}}} \\ & \qquad < \left(1.4+\frac{2}{n+1}\right)\mathbb{E}\|\mathcal{N}(0,\mathbf{I})\| \\ 0 & \text{otherwise} \end{cases} \\ \delta(h_\sigma) &= (1-h_\sigma c_c(2-c_c)) \le 1. \end{aligned}$$

References

[1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[2] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York, NY, USA: Kluwer Academic, 2007.

[3] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications* (Advanced Information and Knowledge Processing). London, U.K.: Springer, 2005.

[4] A. Zhou *et al.*, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[5] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

[6] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control With Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Périaux, K. D. Papailiou, and T. Fogarty, Eds. Athens, Greece: Int. Center Numer. Methods Eng., 2001, pp. 95–100.

[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[8] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.

[9] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0/1 knapsack problem—A comparative experiment," *IEEE Trans. Evol. Comput.*, vol. 6, no. 4, pp. 402–412, Aug. 2002.

[10] Y. Jin, T. Okabe, and B. Sendho, "Adapting weighted aggregation for multiobjective evolution strategies," in *Evolutionary Multi-Criterion Optimization*. Heidelberg, Germany: Springer, 2001, pp. 96–110.

[11] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. Congr. Evol. Comput.*, Canberra, ACT, Australia, 2003, pp. 2678–2684.

[12] H. Ishibuchi, T. Yoshida, and T. Murata, "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 204–223, Apr. 2003.

[13] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving From Nature (PPSN VIII)*, vol. 3242. Heidelberg, Germany: Springer, 2004, pp. 832–842.

[14] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[15] D. Brockhoff, T. Wagner, and H. Trautmann, "R2 indicator-based multiobjective search," *Evol. Comput.*, vol. 23, no. 3, pp. 369–395, 2015.

[16] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, Oct. 2015.

[17] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[18] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[19] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.

[20] N. A. Moubayed, A. Petrovski, and J. McCall, "D2MOPSO: MOPSO based on decomposition and dominance with archiving using crowding distance in objective and solution spaces," *Evol. Comput.*, vol. 22, no. 1, pp. 47–77, 2014.

[21] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, 2011.

[22] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and antcolony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.

[23] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010.

[24] M. Asafuddoula, H. K. Singh, and T. Ray, "Six-sigma robust design optimization using a many-objective decomposition-based evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 490–507, Aug. 2015.

[25] X. Cai, Y. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, Aug. 2015.

[26] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evol. Comput.*, vol. 7, no. 3, pp. 205–230, 1999.

[27] E. Zitzler, K. Deb, and L. Thiele, “Comparison of multiobjective evolutionary algorithms: Empirical results,” *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.

[28] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multi-objective optimization,” Comput. Eng. Netw. Lab. (TIK), ETH Zürich, Zürich, Switzerland, Tech. Rep. TIK 112, 2001.

[29] S. Huband, P. Hingston, L. Barone, and L. While, “A review of multiobjective test problems and a scalable test problem toolkit,” *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

[30] Q. Zhang *et al.*, “Multiobjective optimization test instances for the CEC 2009 special session and competition,” School Comput. Sci. Electr. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-487, 2009.

[31] H.-L. Liu, F. Gu, and Q. Zhang, “Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems,” *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[32] Á. Rubio-Largo, Q. Zhang, and M. A. Vega-Rodríguez, “Multiobjective evolutionary algorithm based on decomposition for 3-objective optimization problems with objectives in different scales,” *Soft Comput.*, vol. 19, no. 1, pp. 157–166, 2015.

[33] N. Hansen and S. Kern, “Evaluating the CMA evolution strategy on multimodal test functions,” in *Parallel Problem Solving From Nature—PPSN VIII*. Heidelberg, Germany: Springer, 2004, pp. 282–291.

[34] A. Auger and N. Hansen, “A restart CMA evolution strategy with increasing population size,” in *Proc. Congr. Evol. Comput.*, Edinburgh, U.K., 2005, pp. 1769–1776.

[35] N. Hansen. (Jan. 2009). *The CMA Evolution Strategy: A Tutorial*. [Online]. Available: http://www.lri.fr/~hansen/cmatutorial.pdf

[36] I. Loshchilov, “CMA-ES with restarts for solving CEC 2013 benchmark problems,” in *Proc. Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 369–376.

[37] C. Igel, N. Hansen, and S. Roth, “Covariance matrix adaptation for multi-objective optimization,” *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, 2007.

[38] S.-Y. Park and J.-J. Lee, “A multi-objective covariance matrix adaptation evolutionary strategy based on decomposition for analog circuit design,” in *Proc. IEEE 39th Annu. Conf. Ind. Electron. Soc. (IECON)*, Vienna, Austria, 2013, pp. 6365–6370.

[39] S. Zapotecas-Martínez *et al.*, “Injecting CMA-ES into MOEA/D,” in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Madrid, Spain, Jul. 2015, pp. 783–790.

[40] P. Larrañaga and J. A. Lozano, Eds., *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Boston, MA, USA: Kluwer Academic, 2002.

[41] Y. Qi *et al.*, “MOEA/D with adaptive weight adjustment,” *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.

[42] A. Zhou, Q. Zhang, Y. Jin, and B. Sendhoff, “Adaptive modelling strategy for continuous multi-objective optimization,” in *Proc. Congr. Evol. Comput.*, Singapore, 2007, pp. 431–437.

Hui Li received the B.Sc. and M.Sc. degrees in applied mathematics from the School of Mathematics and Statistics, Xi’an Jiaotong University, Xi’an, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K.

From 2007 to 2010, he was a Post-Doctoral Research Associate with the School of Computer Science, University of Nottingham, Nottingham, U.K. He is currently an Associate Professor with the School of Mathematics and Statistics, Xi’an Jiaotong University. His current research interests include evolutionary computation, multiobjective optimization, combinatorial optimization, and machine learning.

Dr. Li was a recipient of the 2010 IEEE Transactions on Evolutionary Computation Outstanding Paper Award as one of the inventors for MOEA/D.

Qingfu Zhang (M’01–SM’06) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, in 1984, the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xi’an, China, in 1991 and 1994, respectively.

He is a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong, a Professor on leave from the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., and a Changjiang Visiting Chair Professor with Xidian University. He is currently leading the Metaheuristic Optimization Research Group with the City University of Hong Kong. He holds two patents and has authored many research publications. His current research interests include evolutionary computation, optimization, neural networks, and data analysis and their applications.

Dr. Zhang was a recipient of the Unconstrained Multiobjective Optimization Algorithm Competition at the Congress of Evolutionary Computation 2009 for multiobjective optimization algorithm developed in his group and the 2010 IEEE Transactions on Evolutionary Computation Outstanding Paper Award. He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. He is also an Editorial Board Member of three other international journals.

Jingda Deng received the B.Sc. and M.Sc. degrees from the School of Mathematics and Statistics, Xi’an Jiaotong University, Xi’an, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree in computer science with the City University of Hong Kong, Hong Kong.

His current research interests include evolutionary computation and multiobjective optimization.