Multiobjective Optimization Problems With Complicated Pareto Sets, MOEA/D and NSGA-II

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

***Abstract*—Partly due to lack of test problems, the impact of the Pareto set (PS) shapes on the performance of evolutionary algorithms has not yet attracted much attention. This paper introduces a general class of continuous multiobjective optimization test instances with arbitrary prescribed PS shapes, which could be used for studying the ability of multiobjective evolutionary algorithms for dealing with complicated PS shapes. It also proposes a new version of MOEA/D based on differential evolution (DE), i.e., MOEA/D-DE, and compares the proposed algorithm with NSGA-II with the same reproduction operators on the test instances introduced in this paper. The experimental results indicate that MOEA/D could significantly outperform NSGA-II on these test instances. It suggests that decomposition based multiobjective evolutionary algorithms are very promising in dealing with complicated PS shapes.**

***Index Terms*—Aggregation, decomposition, differential evolution, evolutionary algorithms, multiobjective optimization, Pareto optimality, test problems.**

I. Introduction

A *multiobjective optimization problem* (MOP) can be stated as follows:

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \ldots, f_m(x)) \\ \text{subject to} \quad & x \in \Omega \end{aligned} \tag{1}$$

where Ω is the *decision (variable) space*, R^m is the *objective space*, and $F : \Omega \to R^m$ consists of m real-valued objective functions. If Ω is a closed and connected region in R^n and all the objectives are continuous of x, we call problem (1) a *continuous MOP*.

Let $u = (u_1, \ldots, u_m)$, $v = (v_1, \ldots, v_m) \in R^m$ be two vectors, u is said to *dominate* v if $u_i \le v_i$ for all $i = 1, \ldots, m$, and $u \ne v$.[1] A point $x^\star \in \Omega$ is called *(globally) Pareto optimal* if there is no $x \in \Omega$ such that $F(x)$ dominates $F(x^\star)$. The set of all the Pareto optimal points, denoted by PS, is called the *Pareto set*. The set of all the Pareto objective vectors, $PF = \{F(x) \in R^m | x \in PS\}$, is called the *Pareto front* [1].

Under certain smoothness assumptions, it can be induced from the Karush–Kuhn–Tucker condition that the PS of a continuous MOP defines a piecewise continuous $(m-1)$-dimensional manifold in the decision space [1], [2]. Therefore, the PS of a continuous bi-objective optimization problem is a piecewise continuous 1-D curve in R^n while the PS of a continuous MOP with three objectives is a piecewise continuous 2-D surface. This is the so-called regularity property of continuous MOPs.

Recent years have witnessed significant progress in the development of evolutionary algorithms (EAs) for multiobjective optimization problems [3]–[16]. Multiobjective evolutionary algorithms (MOEAs) aim at finding a set of representative Pareto optimal solutions in a single run. Due largely to the nature of MOEAs, their behaviors and performances are mainly studied experimentally. Continuous multiobjective test problems are most widely used for this purpose since they are easy to describe and understand. It has been well-understood (as reviewed in [17]) that the geometrical shapes of the PF, among other characteristics of an MOP, could affect the performance of MOEAs. In fact, a range of PF shapes such as convex, concave, and mixed PFs can be found in commonly used continuous test problems [18]–[21], which can be used for studying how MOEAs perform with different PF shapes. However, the PS shapes of most existing test problems are often strikingly simple. For example, the PSs of the ZDT test instances [19] with two objectives are part of a line segment defined by

$$\begin{aligned} &0 \le x_1 \le 1, \\ &x_2 = x_3 = \ldots = x_n = 0 \end{aligned}$$

and the PSs of the DTLZ test instances [20] with three objectives are subsets of

$$\begin{aligned} &0 \le x_1, x_2 \le 1, \\ &x_3 = \ldots = x_n = 0.5. \end{aligned}$$

There is no reason that real world problems have such simple PSs.[2] Observing these oversimplified PSs in the existing test instances, Okabe *et al.* first argued the necessity of constructing test instances with complicated PSs and provided a method for controlling PS shapes [24]. They have constructed several test instances with complicated PSs. However, their test instances have two objectives and two decision variables. Very recently, Huband *et al.* [17] and Deb *et al.* [25] emphasized that variable linkages (i.e., parameter dependencies) should be considered in constructing test instances and proposed using variable transformations for introducing variable linkages. Variable linkages could often complicate PS shapes. However, the PS shapes in the

Manuscript received October 18, 2007; revised January 28, 2008. First published September 26, 2008; current version published April 01, 2009.

H. Li was with the Department of Computing and Electronic Systems, University of Essex, Colchester CO4 3SQ, U.K. He is now with the School of Computer Science, University of Nottingham, Nottingham NG8 1BB, U.K. (e-mail: hzl@cs.nott.ac.uk).

Q. Zhang is with the Department of Computing and Electronic Systems, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qzhang@essex.ac.uk).

Digital Object Identifier 10.1109/TEVC.2008.925798

[1]This definition of domination is for minimization. All the inequalities should be reversed if the goal is to maximize the objectives in (1). "Dominate" means "be better than."

[2]Two examples with complicated PS shapes can be found in a vehicle dynamic design problem in [22] and a power plant design optimization in [23].

test instances constructed in [17] and [25] are not easy to be directly controlled and described. Moreover, variable linkages and PS shapes are different aspects of MOPs. Actually, PS shapes are not a focus in [17] and [25]. It is possible that a continuous MOP with complicated variable linkages has a very simple PS. Partially due to lack of test problems, the influence of PS shapes over the performance of MOEAs has attracted little attention in the evolutionary computation community.

Inspired by the strategies for constructing test problems in [24], [25], we have recently proposed several continuous test instances with variable linkages/complicated PSs [26], in which the PS shapes could be easily described. These test instances have also been modified and used in [27] for comparing RM-MEDA with several other MOEAs. The experimental results indicate that complicated PS shapes could cause difficulties for MOEAs. However, the PSs of these test instances are either linear or quadratic, which are not complicated enough for resembling some real-life problems. One of the major purposes in this paper is to propose a general class of continuous test problems with arbitrary prescribed PS shapes for facilitating the study of the ability of MOEAs to deal with the complication of PSs.

The majority of existing MOEAs are based on Pareto dominance [4]–[6], [8], [9], [11], [12]. In these algorithms, the utility of each individual solution is mainly determined by its Pareto dominance relations with other solutions visited in the previous search. Since using Pareto dominance alone could discourage the diversity of search, some techniques such as fitness sharing and crowding have often been used as compensation in these MOEAs [5], [9], [11], [28]. Arguably, NSGA-II [11] is one of the most popular Pareto dominance based MOEAs. The characteristic feature of NSGA-II is its fast nondominated sorting procedure for ranking solutions in its selection.

A Pareto optimal solution to an MOP could be an optimal solution of a single objective optimization problem in which the objective is an aggregation function of all the individual objectives. Therefore, approximation of the PF can be decomposed into a number of single objective optimization subproblems. This is a basic idea behind many traditional mathematical programming methods for approximating the PF [1]. A number of MOEAs adopt this idea for their fitness assignment to some extent [29]–[41]. MOGLS [29], [32] optimizes an aggregation of the objectives with randomly selected aggregation coefficients at each stage, which makes it very easy, at least in principle, to use single objective local search for improving individual solutions. MSOPS uses a number of scalar aggregation functions to guide its search and has demonstrated its advantage for tackling MOPs with many objectives [33], [37], [39]. By using aggregations of the objectives, a decision maker's preference has been incorporated into MOEAs in [41].

MOEA/D is a very recent one using decomposition (i.e., aggregations)[34]. MOEA/D simultaneously optimizes a number of single objective optimization subproblems. The objective in each of these problems is an aggregation of all the objectives. Neighborhood relations among these subproblems are defined based on the distances between their aggregation coefficient vectors. Each subproblem (i.e., scalar aggregation function) is optimized in MOEA/D by using information mainly from its neighboring subproblems.

We believe that comparison studies between MOEAs based on Pareto dominance and those using decomposition (i.e., aggregation) on test problems with various characteristics could be very useful for understanding strengths and weaknesses of these different methodologies and thus identifying important issues which should be addressed in MOEAs. The major contributions of this paper include the following.

- A general class of multiobjective continuous test instances with arbitrary prescribed PS shapes have been proposed.
- A new implementation of MOEA/D with a DE operator and polynomial mutation has been suggested, in which two extra measures have been introduced for maintaining the population diversity.
- Experiments have been conducted to compare MOEA/D and NSGA-II with the same DE and mutation operator on the test instances introduced in this paper.

The remainder of this paper is organized as follows. Section II introduces a class of continuous multiobjective optimization test instances with complicated PS shapes and provides a theorem about their properties. Section III proposes MOEA/D-DE, a new implementation of MOEA/D with a DE operator and polynomial mutation, and describes NSGA-II-DE, an implementation of NSGA-II with the same reproduction operators. Experiments and discussions are given in Section IV. Section V concludes the paper.

II. Multiobjective Test Problem With Prescribed Pareto Set

In our proposed generic continuous test problem, the decision space is

$$\Omega = \prod_{i=1}^{n}[a_i, b_i] \subset R^n \tag{2}$$

where $-\infty < a_i < b_i < +\infty$ for all $i = 1, \ldots, n$. Its m objectives to be minimized take the following form:

$$\begin{aligned} f_1(x) &= \alpha_1(x_I) + \beta_1(x_{II} - g(x_I)) \\ &\vdots \\ f_m(x) &= \alpha_m(x_I) + \beta_m(x_{II} - g(x_I)) \end{aligned} \tag{3}$$

where

- $x = (x_1, \ldots, x_n) \in \Omega$, $x_I = (x_1, \ldots, x_{m-1})$ and $x_{II} = (x_m, \ldots, x_n)$ are two subvectors of x;
- α_i $(i = 1, \ldots, m)$ are functions from $\prod_{i=1}^{m-1}[a_i, b_i]$ to R;
- β_i $(i = 1, \ldots, m)$ are functions from R^{n-m+1} to R^+;
- g is a function from $\prod_{i=1}^{m-1}[a_i, b_i]$ to $\prod_{i=m+1}^{n}[a_i, b_i]$

The following theorem is about the PS and PF of the generic test problem.

Theorem 1: Suppose that

[i] $\beta_i(z) = 0$ for all $i = 1, \ldots, m$ if and only if $z = 0$.

[ii] The PS of the following m-objective optimization problem:

$$\begin{aligned} &\text{minimize } (\alpha_1(x_I), \ldots, \alpha_m(x_I)) \\ &\text{subject to } x_I \in \prod_{i=1}^{m-1}[a_i, b_i] \end{aligned} \tag{4}$$

is $E \subset \prod_{i=1}^{m-1}[a_i, b_i]$.

Then the PS of the generic continuous test problem defined by (2) and (3) is

$$x_{II} = g(x_I), \quad x_I \in E$$

and its PF is the same as that of (4), i.e.,

$$\{(\alpha_1(x_I), \dots, \alpha_m(x_I)) \,|x_I \in E\}.$$

Proof: It suffices to show that $x = (x_I, x_{II})$ is Pareto optimal to the test problem defined by (2) and (3) if and only if $x_I \in E$ and $x_{II} = g(x_I)$.

We first prove the "if" part. Let $x = (x_I, x_{II})$, $y = (y_I, y_{II}) \in \prod_{i=1}^{n}[a_i, b_i]$. If $x_I \in E$ and $x_{II} = g(x_I)$, then, by [i]

$$F(x) = (\alpha_1(x_I), \dots, \alpha_m(x_I)).$$

It is from [ii] that $F(x)$ cannot be dominated by $(\alpha_1(y_I), \dots, \alpha_m(y_I))$, the latter dominates or equals to $F(y)$ since $\beta_i \geq 0$ for all $i = 1, \dots, m$. Therefore, $F(x)$ cannot be dominated by $F(y)$, which implies that x is Pareto optimal and completes the proof of the "if" part.

Now we prove the "only if" part. Let $x = (x_I, x_{II})$. If $x_I \notin E$, then by [ii], x_I is not Pareto optimal to (4). Therefore, there exists y_I such that $(\alpha_1(y_I), \dots, \alpha_m(y_I))$ dominates $(\alpha_1(x_I), \dots, \alpha_m(x_I))$. Noting that the latter dominates $F(x)$ and

$$F(y_I, g(y_I)) = (\alpha_1(y_I), \dots, \alpha_m(y_I))$$

we have that $F(y_I, g(y_I))$ dominates $F(x)$. Thus, x is not Pareto optimal. If $x_{II} \neq g(x_I)$, then, by [i] and non-negativeness of β_i, $F(x)$ is dominated by $F(x_I, g(x_{II}))$. Therefore, x is not Pareto optimal. This completes the proof of the "only if" part.

In the following, we would like to make several comments on the generic test problem defined by (2) and (3).

- If the dimensionality of E is $m-1$ as in all the test instances constructed in this paper (in fact, $E = \prod_{i=1}^{m-1}[a_i, b_i]$ in all of them), the PS and PF of the test problem will be $(m-1)$-D and then exhibit the regularity property.
- The PF and PS of the test problem are determined by α_i and g, respectively. In principle, one can obtain arbitrary $(m-1)$-D PSs and PFs by setting appropriate α_i and g.
- Functions β_i control the difficulty of convergence. If $\sum_{i=1}^{m}\beta_i$ has many local minima, then the test problem may have many local Pareto optimal solutions.

Like DTLZ and WFG test problems [17], [20], the proposed test problem uses component functions for defining its PF and introducing multimodality. Its major advantage over others is that the PS can be easily prescribed. To generate a test instance, one needs to construct α_i, β_i, and g which satisfy conditions [i] and [ii] in Theorem 1. Table I lists nine test instances generated in such a way, and Fig. 1 plots the projections of their PSs onto the space of x_1, x_2, and x_3. In the following, taking three test instances in Table I for example, we explain how test instances could be generated.

F2:

- The decision space $\Omega = [0,1] \times [-1,1]^{n-1}$, $x_I = x_1$, and $x_{II} = (x_2, \dots, x_n)$;
- $\alpha_1(x_1) = x_1$ and $\alpha_2(x_1) = 1 - \sqrt{x_1}$;
- $g(x_1) = (g_2(x_1), \dots, g_n(x_1))$ and

$$g_j(x_1) = \sin\left(6\pi x_1 + \frac{j\pi}{n}\right), \quad j = 2, \dots, n;$$

- β_1 and β_2 are functions from R^{n-1} to R^+. Let $y_{2:n} = (y_2, \dots, y_n) \in R^{n-1}$,

$$\beta_1(y_{2:n}) = \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2,$$
$$\beta_2(y_{2:n}) = \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2,$$

where $J_1 = \{j|j \text{ is odd and } 2 \leq j \leq n\}$ and $J_2 = \{j|j \text{ is even and } 2 \leq j \leq n\}$.

By Theorem 1, the PF of F2 in the objective space is

$$\left\{(x_1, 1 - \sqrt{x_1}) \in R^2 | 0 \leq x_1 \leq 1\right\}$$

and its PS is

$$x_{II} = g(x_1), \quad 0 \leq x_1 \leq 1.$$

F6:

- The decision space $\Omega = [0,1]^2 \times [-2,2]^{n-2}$, $x_I = (x_1, x_2)$ and $x_{II} = (x_3, \dots, x_n)$

$$\alpha_1(x_I) = \cos(0.5\pi x_1)\cos(0.5\pi x_2),$$
$$\alpha_2(x_I) = \cos(0.5\pi x_1)\sin(0.5\pi x_2),$$
$$\alpha_3(x_I) = \sin(0.5\pi x_1)$$

- $g(x_I) = (g_3(x_I), \dots, g_n(x_I))$ and

$$g_j(x_I) = 2x_2 \sin\left(2\pi x_1 + \frac{j\pi}{n}\right), \quad j = 3, \dots, n$$

- β_1, β_2, and β_3 are functions from R^{n-2} to R^+. Let $y_{3:n} = (y_3, \dots, y_n) \in R^{n-2}$

$$\beta_1(y_{3:n}) = \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2$$
$$\beta_2(y_{3:n}) = \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2$$
$$\beta_3(y_{3:n}) = \frac{2}{|J_3|}\sum_{j\in J_3} y_j^2$$

where

$$J_1 = \{j|3 \leq j \leq n, \text{ and } j-1 \text{ is a multiplication of } 3\}$$
$$J_2 = \{j|3 \leq j \leq n, \text{ and } j-2 \text{ is a multiplication of } 3\}$$
$$J_3 = \{j|3 \leq j \leq n, \text{ and } j \text{ is a multiplication of } 3\}.$$

TABLE I
TEST INSTANCES WITH COMPLICATED PS SHAPES

Instance	Objectives and PSs	Variable Bounds
F1	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$ where $J_1 = \{j \vert j$ is odd and $2 \le j \le n\}$ and $J_2 = \{j \vert j$ is even and $2 \le j \le n\}$. Its PS is $x_j = x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}, j = 2,\ldots,n.$	$[0,1]^n$
F2	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - \sin(6\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - \sin(6\pi x_1 + \frac{j\pi}{n}))^2$ where J_1 and J_2 are the same as those of F1. Its PS is $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2,\ldots,n.$	$[0,1]\times[-1,1]^{n-1}$
F3	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - 0.8x_1\cos(6\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n}))^2$ where J_1 and J_2 are the same as those of F1. Its PS is $x_j = \begin{cases} 0.8x_1\cos(6\pi x_1 + \frac{j\pi}{n}) & j \in J_1 \\ 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n}) & j \in J_2 \end{cases}$	$[0,1]\times[-1,1]^{n-1}$
F4	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - 0.8x_1\cos(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}))^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n}))^2$ where J_1 and J_2 are the same as those of F1. Its PS is $x_j = \begin{cases} 0.8x_1\cos(\frac{6\pi x_1 + \frac{j\pi}{n}}{3}) & j \in J_1 \\ 0.8x_1\sin(6\pi x_1 + \frac{j\pi}{n}) & j \in J_2 \end{cases}$	$[0,1]\times[-1,1]^{n-1}$
F5	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}\{x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\cos(6\pi x_1 + \frac{j\pi}{n})\}^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}\{x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\sin(6\pi x_1 + \frac{j\pi}{n})\}^2$ where J_1 and J_2 are the same as those of F1. Its PS is $x_j = \begin{cases} [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\cos(6\pi x_1 + \frac{j\pi}{n}) & j \in J_1 \\ [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + 0.6x_1]\sin(6\pi x_1 + \frac{j\pi}{n}) & j \in J_2 \end{cases}$	$[0,1]\times[-1,1]^{n-1}$
F6	$f_1 = \cos(0.5x_1\pi)\cos(0.5x_2\pi) + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = \cos(0.5x_1\pi)\sin(0.5x_2\pi) + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2$ $f_3 = \sin(0.5x_1\pi) + \frac{2}{\lvert J_3\rvert}\sum_{j\in J_3}(x_j - 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}))^2$ where $J_1 = \{j \vert 3 \le j \le n$, and $j-1$ is a multiplication of $3\}$, $J_2 = \{j \vert 3 \le j \le n$, and $j-2$ is a multiplication of $3\}$, $J_3 = \{j \vert 3 \le j \le n$, and j is a multiplication of $3\}$ Its PS is $x_j = 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}), j = 3,\ldots,n.$	$[0,1]^2\times[-2,2]^{n-2}$
F7	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(4y_j^2 - \cos(8y_j\pi) + 1.0)$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(4y_j^2 - \cos(8y_j\pi) + 1.0)$ where J_1 and J_2 are the same as those of F1 and $y_j = x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}, j = 2,\ldots,n.$ Its PS is $x_j = x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}, j = 2,\ldots,n.$	$[0,1]^n$
F8	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}(4\sum_{j\in J_1}y_j^2 - 2\prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}(4\sum_{j\in J_2}y_j^2 - 2\prod_{j\in J_2}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$ where J_1 and J_2 are the same as those of F1 and $y_j = x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}, j = 2,\ldots,n.$ Its PS is $x_j = x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}, j = 2,\ldots,n.$	$[0,1]^n$
F9	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - \sin(6\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = 1 - x_1^2 + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - \sin(6\pi x_1 + \frac{j\pi}{n}))^2$ where J_1 and J_2 are the same as those of F1. Its PS is $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2,\ldots,n.$	$[0,1]\times[-1,1]^{n-1}$

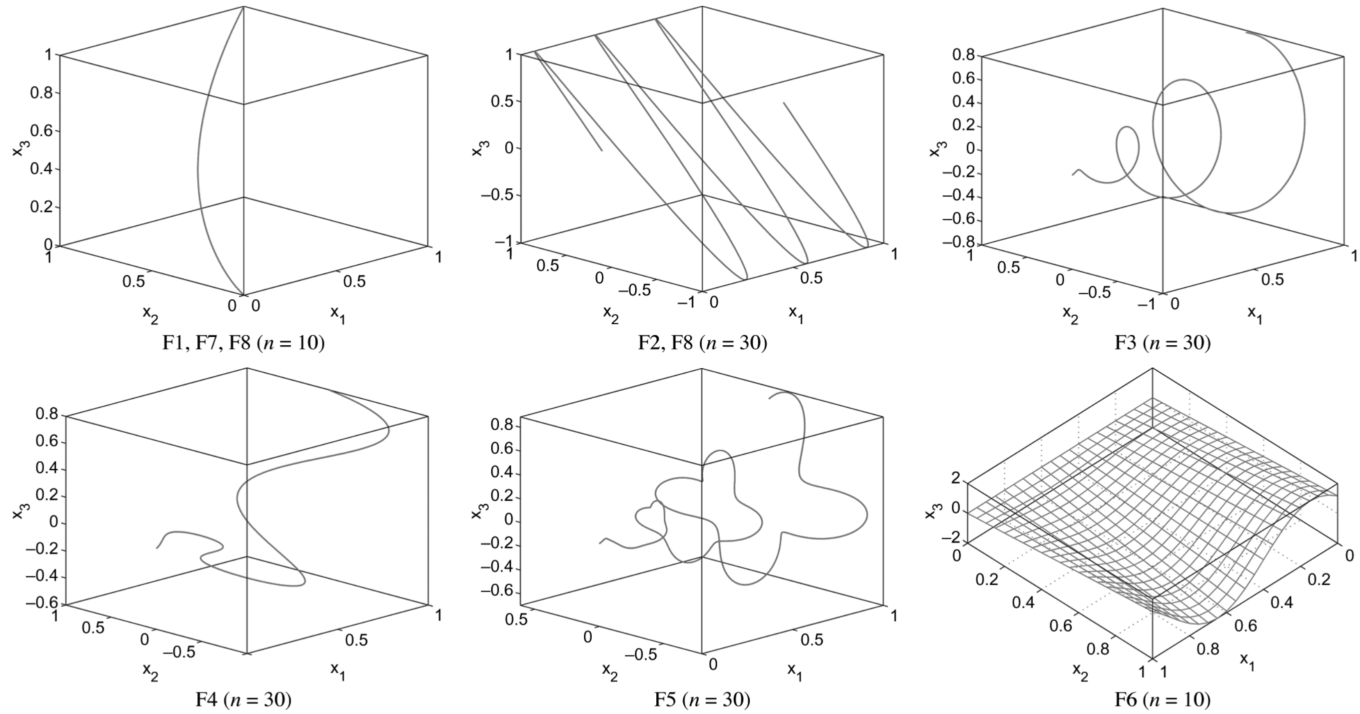


Fig. 1. Projections of PSs of F1–F9 onto the $x_1 - x_2 - x_3$ space.

By Theorem 1, the PF of F6 is

$$\left\{(\alpha_1(x_I), \alpha_2(x_I), \alpha_3(x_I)) \in R^3 | x_I \in [0,1]^2\right\}$$

and its PS is

$$x_{II} = g(x_I), \quad 0 \le x_1, x_2 \le 1.$$

F8:

- The decision space $\Omega = [0,1]^n$, $x_I = x_1$ and $x_{II} = (x_2, \ldots, x_n)$;
- $\alpha_1(x_1) = x_1$ and $\alpha_2(x_1) = 1 - \sqrt{x_1}$;
- $g(x_1) = (g_2(x_1), \ldots, g_n(x_1))$ and

$$g_j(x_1) = x_1^{0.5\left[1.0 + \frac{3(j-2)}{n-2}\right]}, \quad j = 2, \ldots, n;$$

- β_1 and β_2 are functions from R^{n-1} to R^+. Let $y_{2:n} = (y_2, \ldots, y_n) \in R^{n-1}$,

$$\beta_1(y_{2:n}) = \frac{2}{|J_1|}\left(4\sum_{j \in J_1} y_j^2 - 2\prod_{j \in J_1}\cos\left(\frac{20\pi y_i}{\sqrt{j}}\right) + 2\right)$$

$$\beta_2(y_{2:n}) = \frac{2}{|J_2|}\left(4\sum_{j \in J_2} y_j^2 - 2\prod_{j \in J_2}\cos\left(\frac{20\pi y_i}{\sqrt{j}}\right) + 2\right)$$

where $J_1 = \{j | j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j | j \text{ is even and } 2 \le j \le n\}$.

By Theorem 1, the PF of F8 in the objective space is

$$\left\{(x_1, 1 - \sqrt{x_1}) \in R^2 | 0 \le x_1 \le 1\right\}$$

and its PS is

$$x_{II} = g(x_1), \quad 0 \le x_1 \le 1.$$

Since $\beta_1 + \beta_2$ has many local minima, this instance has many local Pareto optimal solutions.

III. Algorithms in Comparison

A. MOEA/D With DE

MOEA/D requires a decomposition approach for converting approximation of the PF of problem (1) into a number of single objective optimization problems. In principle, any decomposition approach can serve for this purpose. Three different decomposition approaches have been used in [34]. In this paper, we use the Tchebycheff approach [1]. A single objective optimization subproblem in this approach is

$$\begin{aligned} &\text{minimize} \quad g(x|\lambda, z^*) = \max_{1 \le i \le m} \{\lambda_i |f_i(x) - z_i^*|\} \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{5}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is a weight vector, i.e., $\lambda_i \ge 0$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^m \lambda_i = 1$. $z^* = (z_1^*, \ldots, z_m^*)$ is the reference point, i.e., $z_i^* = \min\{f_i(x) | x \in \Omega\}$[3] for each $i = 1, \ldots, m$.

It is well known that, under mild conditions, for each Pareto optimal point there exists a weight vector λ such that it is the optimal solution of (5) and each optimal solution of (5) is a Pareto optimal solution of problem (1). Let $\lambda^1, \ldots, \lambda^N$ be a set of weight vectors. Correspondingly, we have N single objective optimization subproblems where the ith subproblem is (5) with $\lambda = \lambda^i$. If N is reasonably large and $\lambda^1, \ldots, \lambda^N$ are properly

[3]In the case when the goal of (1) is maximization, $z_i^* = \max\{f_i(x) | x \in \Omega\}$.

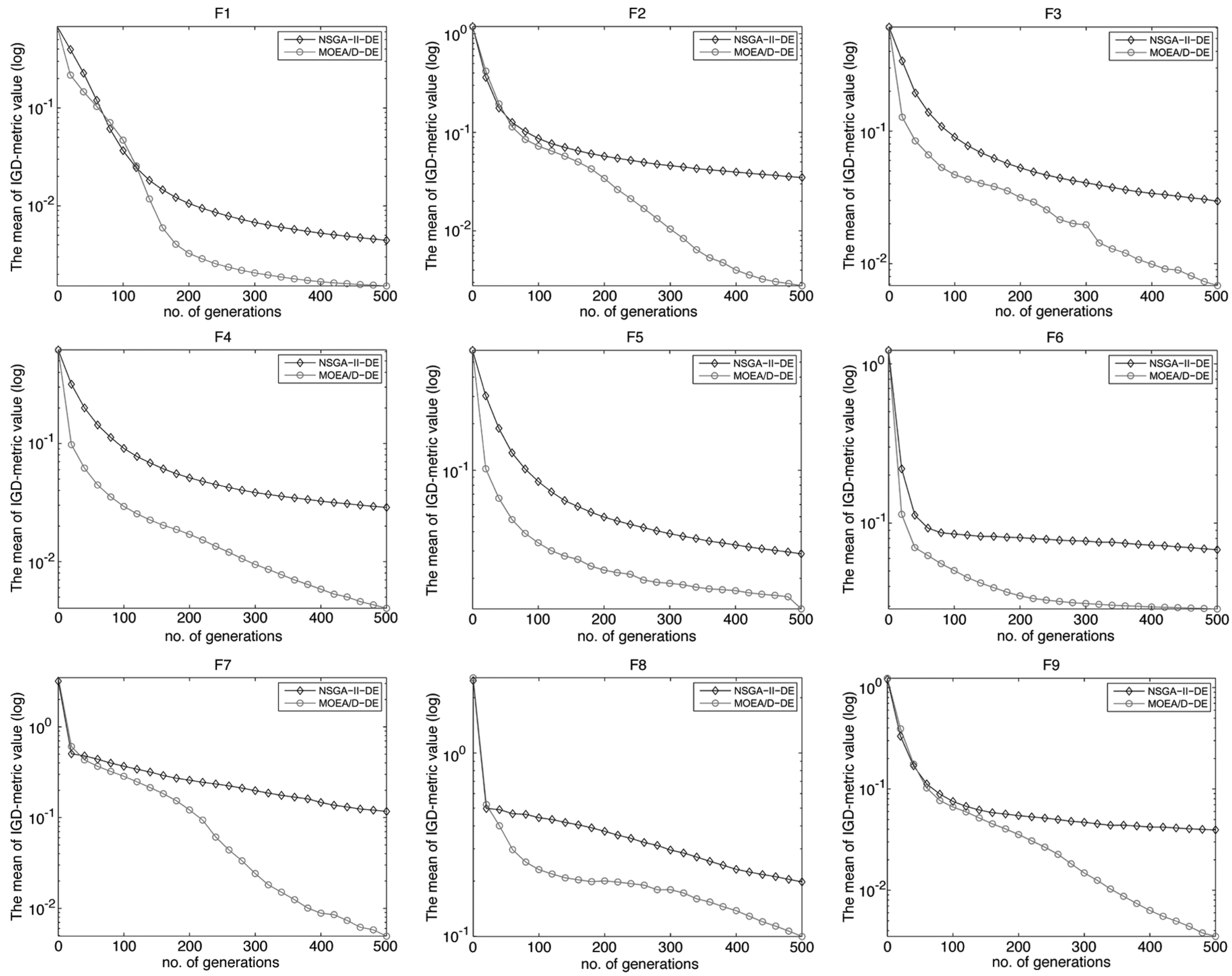


Fig. 2. Evolution of the mean of IGD-metric values versus the number of generations.

TABLE II
IGD-METRIC VALUES OF THE NONDOMINATED SOLUTIONS FOUND BY MOEA/D-DE AND NSGA-II-DE ON F1–F9

IGD-value	MOEA/D-DE			NSGA-II-DE		
Instance	mean	min	std	mean	min	std
F1	0.0015	0.0015	0	0.0044	0.0044	0
F2	0.0028	0.0023	0.0004	0.0349	0.0203	0.0066
F3	0.0068	0.0022	0.0099	0.0296	0.0228	0.0030
F4	0.0040	0.0025	0.0014	0.0288	0.0251	0.0021
F5	0.0127	0.0073	0.0069	0.0288	0.0244	0.0031
F6	0.0289	0.0276	0.0014	0.0680	0.0522	0.0072
F7	0.0049	0.0015	0.0063	0.1171	0.0270	0.0716
F8	0.0998	0.0487	0.0429	0.1981	0.1191	0.0494
F9	0.0035	0.0025	0.0008	0.0395	0.0303	0.0061

selected, then the optimal solutions to these subproblems will provide a good approximation to the PF or PS of problem (1).

MOEA/D attempts to optimize these N single objective optimization subproblems simultaneously instead of solving problem (1) directly. In MOEA/D, the T closest weight vectors in $\{\lambda^1, \ldots, \lambda^N\}$ to a weight vector λ^i constitute the neighborhood of λ^i. The neighborhood of the ith subproblem consists of all the subproblems with the weight vectors from the neighborhood of λ^i. Since $g(x|\lambda, z^*)$ is continuous of λ, the optimal solutions of neighboring subproblems should be close in the decision space. MOEA/D exploits the neighborhood relationship among the subproblems for making its search effectively and efficiently.

A general framework of MOEA/D has been proposed in [34]. A simple implementation of MOEA/D with simulated binary crossover (SBX) [14], called hereafter MOEA/D-SBX, in which the SBX operator was used for generating new solutions, has been tested on ZDT and DTLZ test instances. The experimental results have shown that MOEA/D-SBX performs well on these test instances [34]. However, our pilot experiments have indicated that MOEA/D-SBX is not suitable for dealing with the test instances constructed in this paper. The major reasons could be that 1) the population in MOEA/D-SBX may lose diversity, which is needed for exploring the search space effectively, particularly at the early stage of the search when ap-

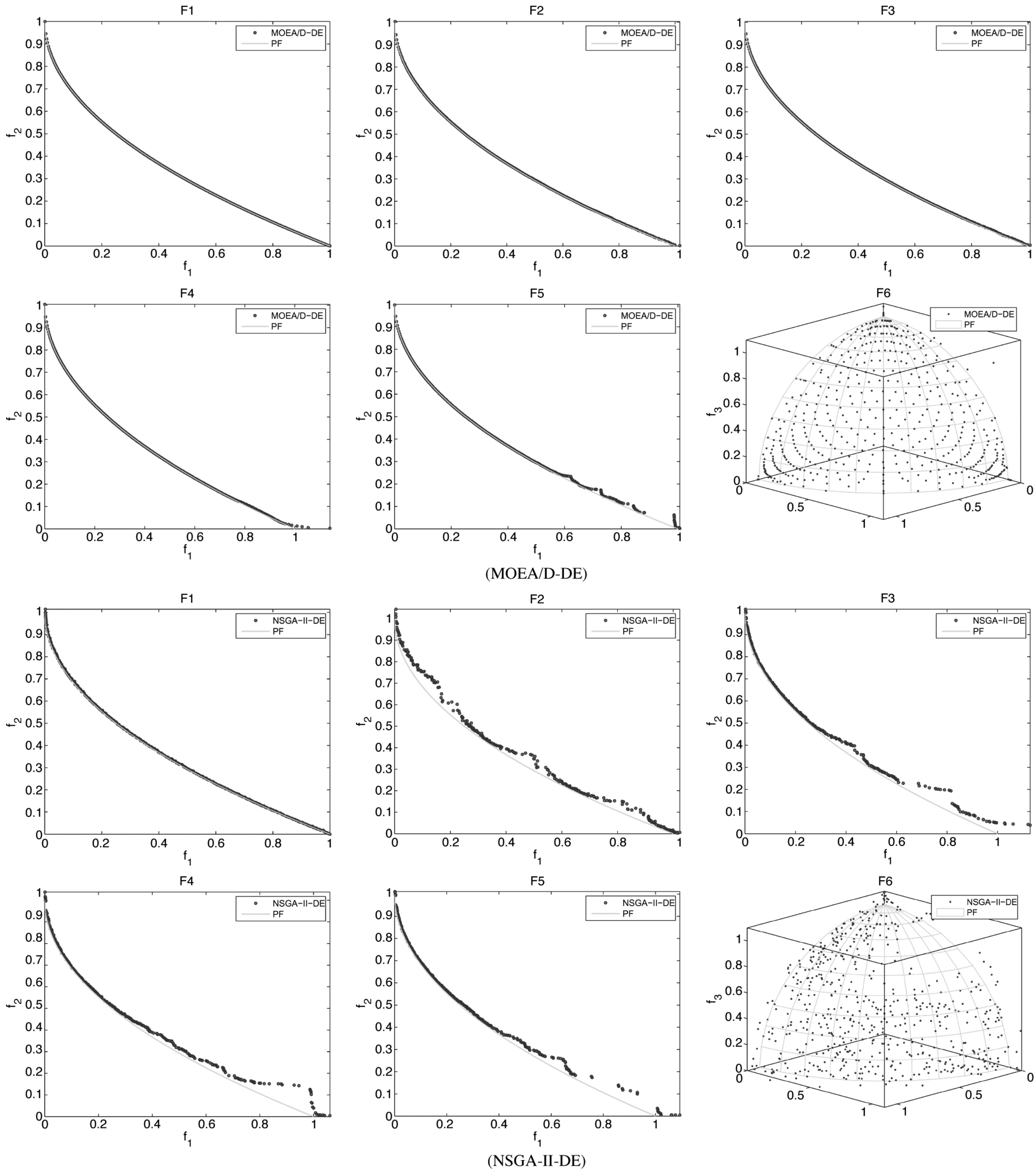


Fig. 3. Plots of the final populations with the lowest IGD-metric values found by MOEA/D-DE and NSGA-II-DE in 20 runs in the objective space on F1–F6.

plied to MOPs with complicated PSs, and 2) the SBX operator often generates inferior solutions in MOEAs, as shown in the recent experiments in [25], [42]. To overcome these shortcomings, this paper proposes a new implementation of MOEA/D, called MOEA/D-DE for dealing with continuous MOPs with complicated PSs. MOEA/D-DE uses a differential evolution (DE) [43] operator and a polynomial mutation operator [14] for producing new solutions, and it has two extra measures for maintaining the population diversity.

At each generation, MOEA/D-DE maintains the following:

- a population of N points $x^1, \ldots, x^N \in \Omega$, where x^i is the current solution to the ith subproblem;
- $FV^1, \ldots, FV^N$, where FV^i is the F-value of x^i, i.e., $FV^i = F(x^i)$ for each $i = 1, \ldots, N$;

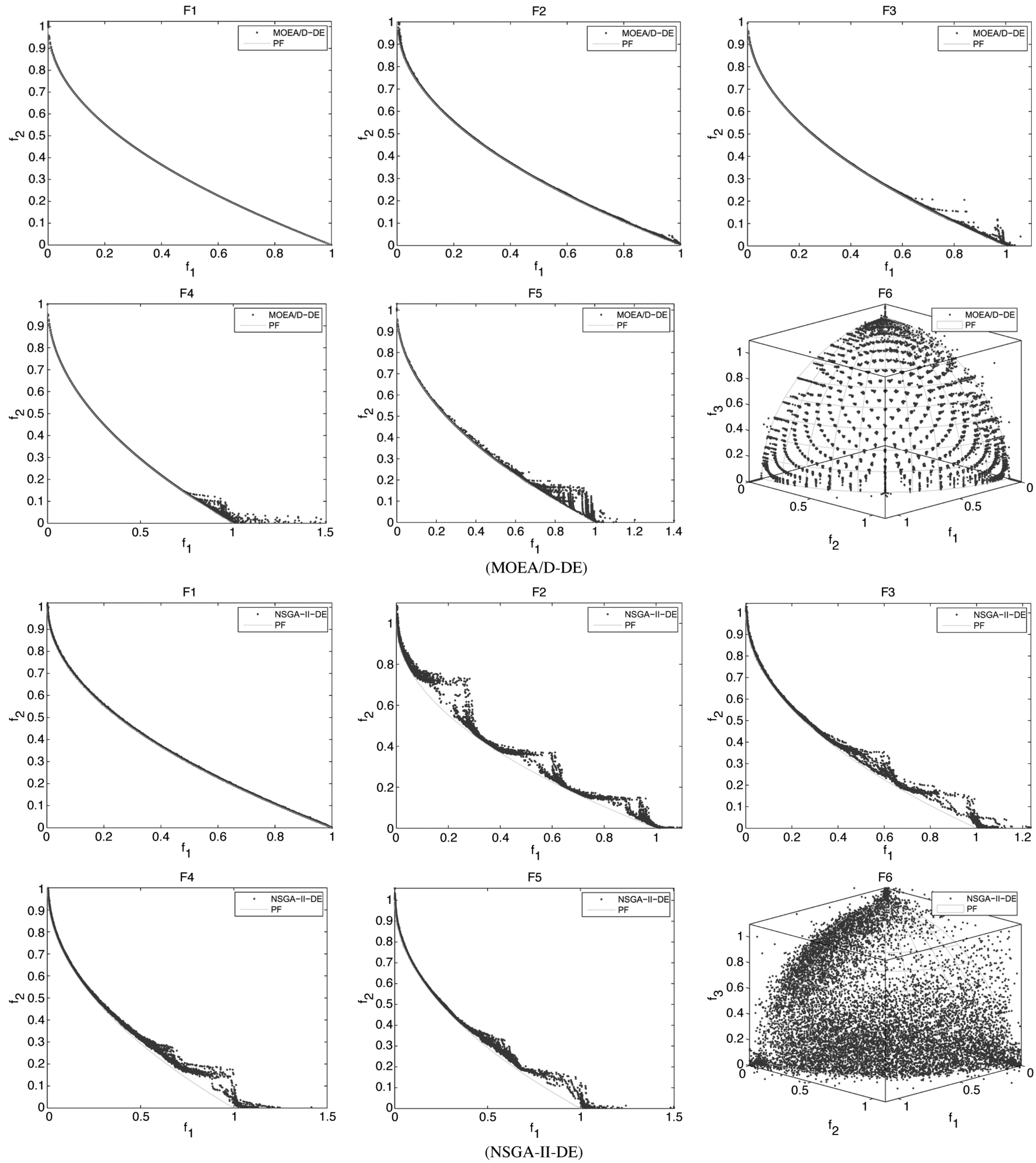


Fig. 4. Plots of all the 20 final populations produced by MOEA/D-DE and NSGA-II-DE in the objective space on F1–F6.

- $z = (z_1, \ldots, z_m)$, where z_i is the best value found so far for objective f_i.

The algorithm works as follows.

Input:
- Problem (1);
- a stopping criterion;
- N: the number of the subproblems considered in MOEA/D-DE;
- $\lambda^1, \ldots, \lambda^N$: a set of N weight vectors;
- T: the number of the weight vectors in the neighborhood of each weight vector;
- δ: the probability that parent solutions are selected from the neighborhood;

- n_r: the maximal number of solutions replaced by each child solution.

Output:

- Approximation to the PS: $\{x^1, \ldots, x^N\}$;
- Approximation to the PF: $\{F(x^1), \ldots, F(x^N)\}$.

Step 1 Initialization

Step 1.1. Compute the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each weight vector. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$ where $\lambda^{i_1}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.

Step 1.2 Generate an initial population $x^1, \ldots, x^N$ by uniformly randomly sampling from Ω. Set $FV^i = F(x^i)$.

Step 1.3 Initialize $z = (z_1, \ldots, z_m)$ by setting $z_j = \min_{1 \le i \le N} f_j(x^i)$.

Step 2 Update

For $i = 1, \ldots, N$, do

Step 2.1 Selection of Mating/Update Range: Uniformly randomly generate a number $rand$ from [0, 1].

Then set

$$P = \begin{cases} B(i) & \text{if } rand < \delta, \\ \{1, \ldots, N\} & \text{otherwise.} \end{cases}$$

Step 2.2 Reproduction: Set $r_1 = i$ and randomly select two indexes r_2 and r_3 from P, and then generate a solution $\bar{y}$ from x^{r_1}, x^{r_2} and x^{r_3} by a DE operator, and then perform a mutation operator on $\bar{y}$ with probability p_m to produce a new solution y.

Step 2.3 Repair: If an element of y is out of the boundary of Ω, its value is reset to be a randomly selected value inside the boundary.

Step 2.4 Update of z: For each $j = 1, \ldots, m$, if $z_j > f_j(y)$, then set $z_j = f_j(y)$.

Step 2.5 Update of Solutions: Set $c = 0$ and then do the following:

1) If $c = n_r$ or P is empty, go to **Step 3**. Otherwise, randomly pick an index j from P.
2) If $g(y|\lambda^j, z) \le g(x^j|\lambda^j, z)$, then set $x^j = y$, $FV^j = F(y)$ and $c = c + 1$.
3) Remove j from P and go to 1).

Step 3 Stopping Criterion If the stopping criterion is satisfied, then stop and output $\{x^1, \ldots, x^N\}$ and $\{F(x^1), \ldots, F(x^N)\}$. Otherwise go to **Step 2**.

In the DE operator used in Step 2.2, each element $\bar{y}_k$ in $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_n)$ is generated as follows:

$$\bar{y}_k = \begin{cases} x_k^{r_1} + F \times (x_k^{r_2} - x_k^{r_3}) & \text{with probability } CR, \\ x_k^{r_1}, & \text{with probability } 1 - CR \end{cases} \tag{6}$$

where CR and F are two control parameters.

The polynomial mutation in Step 2.2 generates $y = (y_1, \ldots, y_n)$ from $\bar{y}$ in the following way:

$$y_k = \begin{cases} \bar{y}_k + \sigma_k \times (b_k - a_k) & \text{with probability } p_m, \\ \bar{y}_k & \text{with probability } 1 - p_m \end{cases} \tag{7}$$

with

$$\sigma_k = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1 & \text{if } rand < 0.5, \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases}$$

where $rand$ is a uniform random number from [0, 1]. The distribution index η and the mutation rate p_m are two control parameters. a_k and b_k are the lower and upper bounds of the kth decision variable, respectively.

We would like to make the following remarks on similarities and differences between MOEA/D-DE and its predecessor, MOEA/D-SBX.

- Since it is often very computationally expensive to find the exact reference point z^*, we use z, which is initialized in Step 1.3 and updated in Step 2.4, as a substitute for z^* in g. The same strategy has been in MOEA/D-SBX.
- The first extra measure for maintaining the population diversity in MOEA/D-DE, which is not used on MOEA/D-SBX, allows three parent solutions to be selected from the whole population with a low probability $1 - \delta$ (Step 2.1 and Step 2.2). In such a way, a very wide range of child solutions could be generated due to the dissimilarity among these parent solutions. Therefore, the exploration ability of the search could be enhanced.
- In MOEA/D-SBX, the maximal number of solutions replaced by a child solution could be as large as T, the neighborhood size. If a child solution is of high quality, it may replace most of current solutions to its neighboring subproblems. As a result, diversity among the population could be reduced significantly. To overcome this shortcoming, the maximal number of solutions replaced by a child solution in MOEA/D-DE is bounded by n_r, which should be set to be much smaller than T in the implementation. Therefore, there is little chance that a solution has many copies in the population. This is the second extra measure in MOEA/D-DE for maintaining its population diversity.
- DE operators often outperform other genetic operators in single objective optimization. Furthermore, if CR is set to be 1.0 as in our simulation studies, the DE operator defined in (6) will be invariant of any orthogonal coordinate rotation, which is desirable for dealing with complicated PSs. For these reasons, we use a DE operator in MOEA/D-DE.

B. NSGA-II With DE

Comparing qualities of different solutions in NSGA-II is based on their nondomination ranks and crowded distances, which can be obtained by a fast sorting algorithm proposed in [11]. The lower nondomination rank of a solution is, the better it is. If two solutions have the same nondomination rank, NSGA-II prefers the solution with the larger crowded distance. NSGA-II-DE used in our experimental studies is the same as NSGA-II-SBX in [11] except that it replaces the SBX operator in NSGA-II-SBX by the DE operator.

NSGA-II-DE maintains a population P_t of size N at generation t and generates P_{t+1} from P_t in the following way.

Step 1) Do the following independently N times to generate N new solutions.

Step 1.1 Select three solutions x^{r_1}, x^{r_2} and x^{r_3} from P_t by using binary tournament selection.

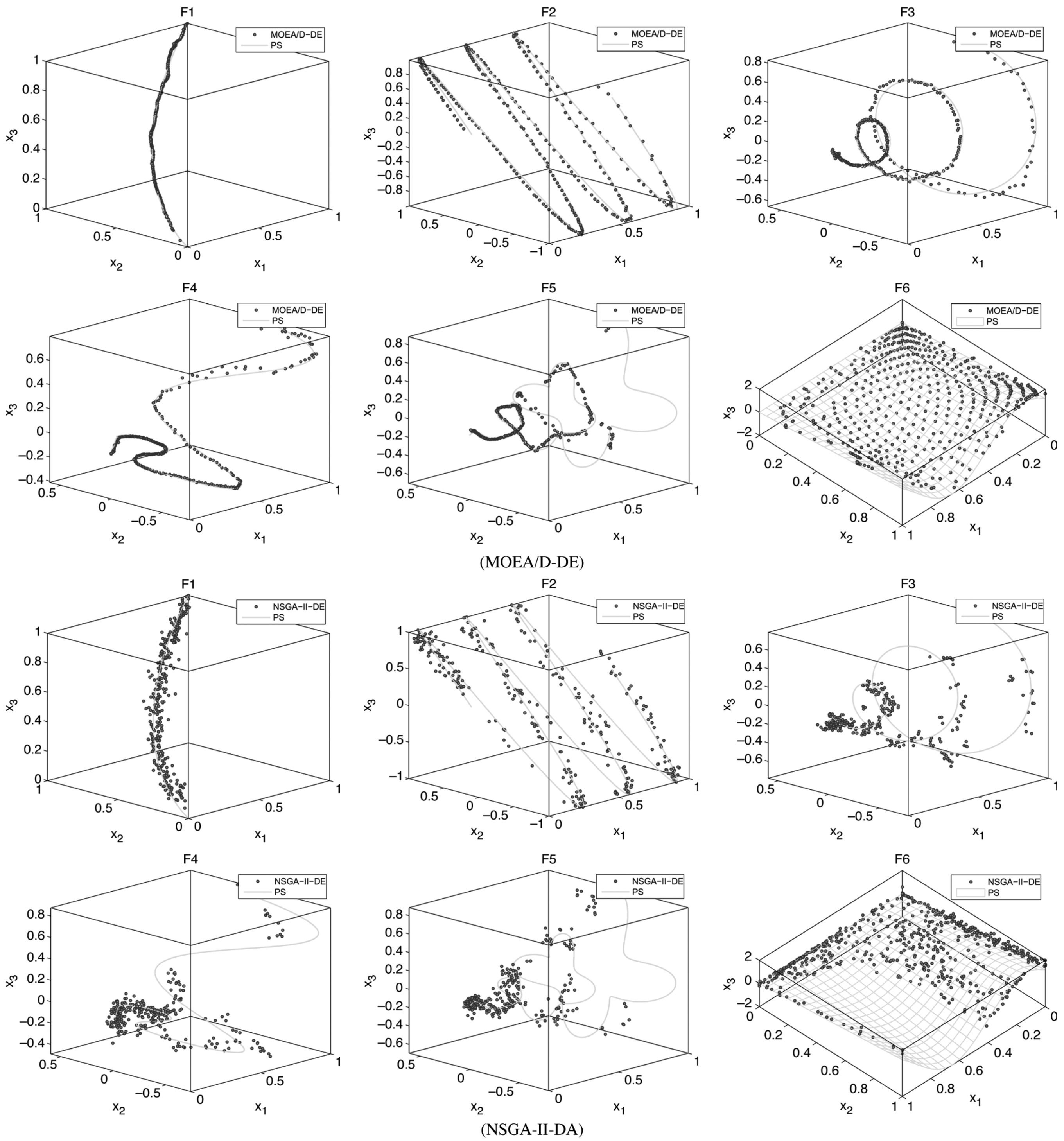


Fig. 5. Plots of the final populations with the lowest IGD-metric values found by MOEA/D-DE and NSGA-II-DE in 20 runs in the $x_1 - x_2 - x_3$ space on F1–F6.

Step 1.2 Generate a solution $\bar{y}$ from x^{r_1}, x^{r_2} and x^{r_3} by the DE operator defined in (6), and then perform a mutation operator (7) on $\bar{y}$ to produce a new solution y.

Step 1.3 If an element of y is out of the boundary of Ω, its value is reset to be a randomly selected value inside the boundary.

Step 2) Combine all the new solutions generated in **Step 1** and all the solutions in P_t together and form a combined population of size $2N$. Select the N best solutions from the combined population to constitute P_{t+1}.

The procedure for generating new solutions in NSGA-II-DE is exactly the same as in MOEA/D-DE. Several variants of NSGA-II with DE [42], [44]–[46] have been proposed for dealing with rotated MOPs or MOPs with nonlinear variable linkages. There is no big difference among these variants. None of them employs mutation after DE operators. Our pilot

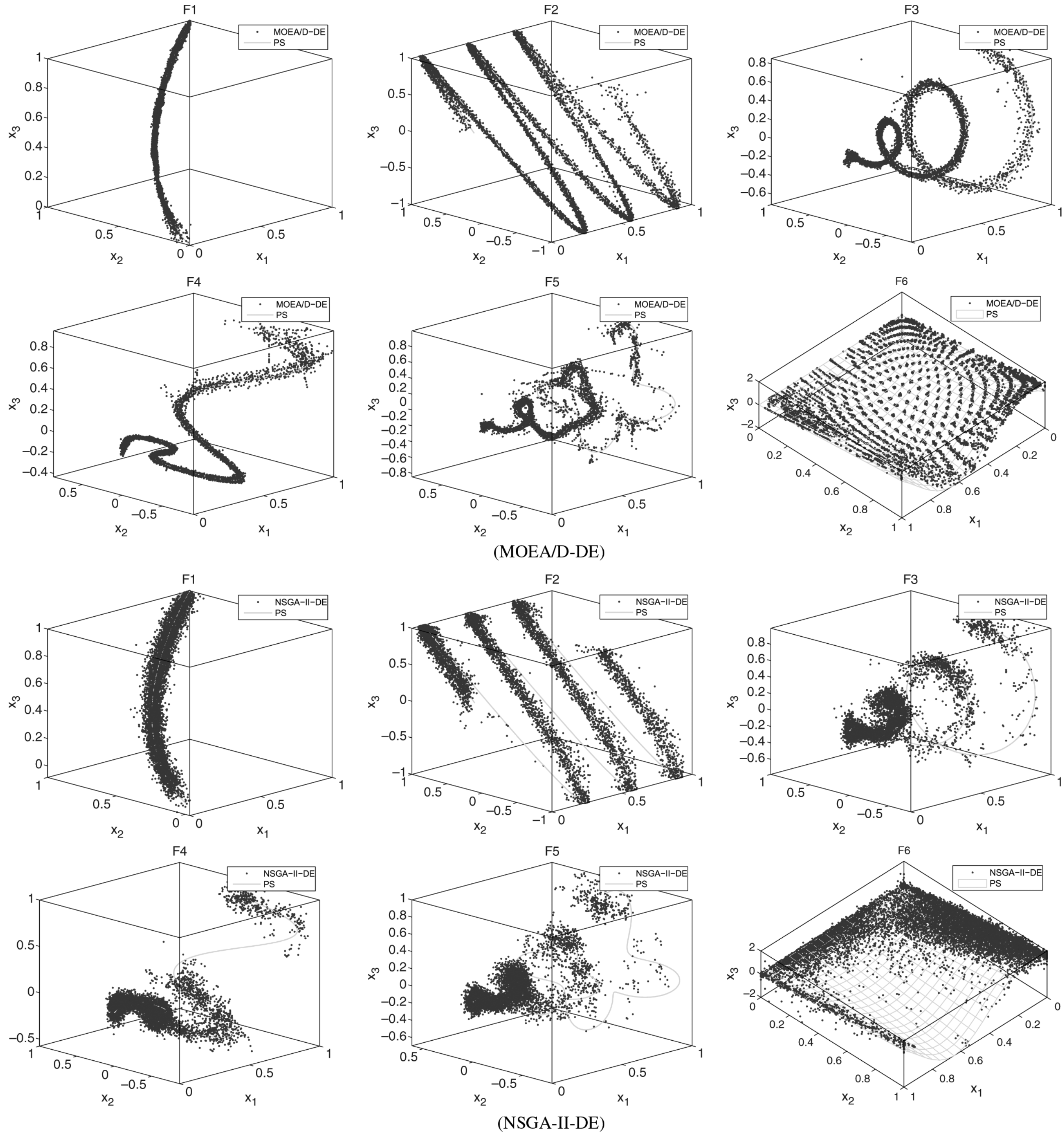


Fig. 6. Plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in the $x_1 - x_2 - x_3$ space on F1–F6.

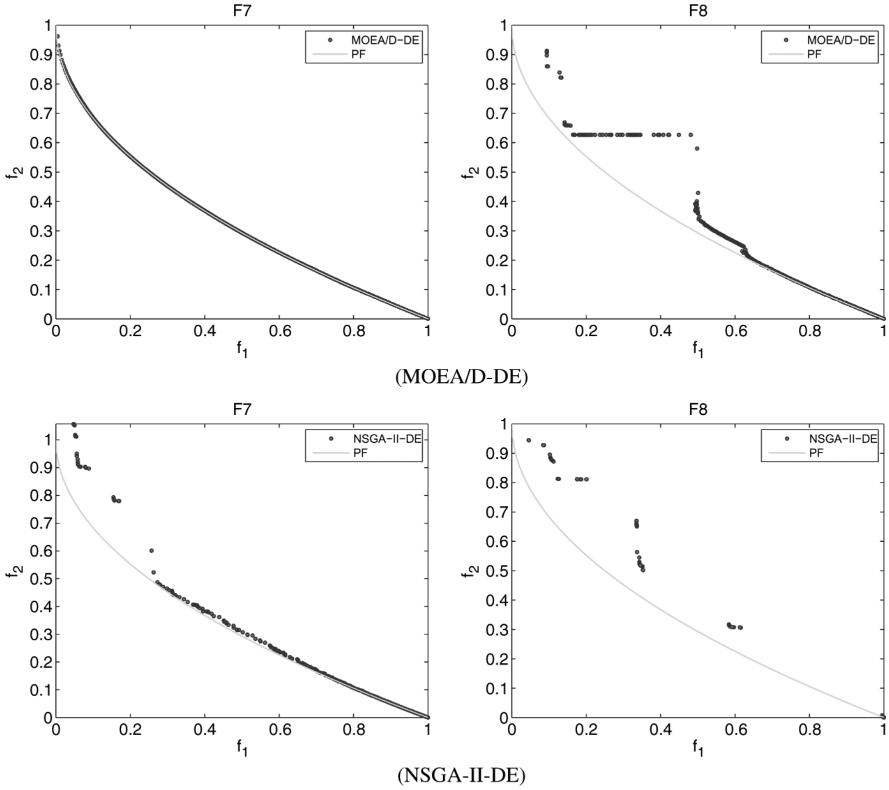
experimental studies have shown that the polynomial mutation operator does slightly improve the algorithm performance, particularly, on MOPs with complicated PSs.

IV. Experimental Results

A. Parameter Settings

Both MOEA/D-DE and NSGA-II-DE have been implemented in C++. The parameter setting in our experimental studies is as follows.

1) *Control parameters in DE and polynomial mutation*
   - $CR = 1.0$ and $F = 0.5$ in the DE operator
   - $\eta = 20$ and $p_m = 1/n$ in the polynomial mutation operator.
2) *The number of decision variables*: It is set to be 30 in F1–F5 and F9, and 10 in F6, F7 and F8.
3) *Number of runs and stopping condition*: Each algorithm is run 20 times independently for each test instance. The algorithms stop after a given number of generations. The maximal number of generations is set to 500 for all the test instances.



(MOEA/D-DE)

(NSGA-II-DE)

Fig. 7. Plots of the final populations with the lowest IGD-metric values found by MOEA/D-DE and NSGA-II-DE in 20 runs in the objective space on F7 and F8.

4) *The population size and weight vectors in MOEA/D-DE*: They are controlled by an integer H. More precisely, $\lambda^1, \ldots, \lambda^N$ are all the weight vectors in which each individual weight takes a value from

$$\left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}.$$

[4]Therefore, the population size (i.e., the number of weight vectors) is

$$N = C_{H+m-1}^{m-1}$$

where m is the number of objectives.

H is set to be 299 for all the 2-objective test instances, and 33 for the 3-objectives instances. Consequently, the population size N is 300 for the 2-objective instances and 595 for the 3-objective instances.

5) *The population size in NSGA-II-DE*: Its setting is the same as in MOEA/D-DE.

6) *Other control parameters in MOEA/D-DE*
   - $T = 20$;
   - $\delta = 0.9$;
   - $n_r = 2$.

Since the population size and the number of generations are the same in MOEA/D-DE and NSGA/D-DE for each test instance, both algorithms will make the same number of function evaluations.

[4]This method for specifying weight vectors with uniform spread has also been used in [32], [38] and [34].

B. Performance Metric

The inverted generational distance (IGD) [47] is used in assessing the performance of the algorithms in our experimental studies.

Let P^* be a set of uniformly distributed points in the objective space along the PF. Let P be an approximation to the PF, the inverted generational distance from P^* to P is defined as

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

where $d(v, P)$ is the minimum Euclidean distance between v and the points in P. If $|P^*|$ is large enough to represent the PF very well, IGD(P^*, P) could measure both the diversity and convergence of P in a sense. To have a low value of IGD(P^*, P), P must be very close to the PF and cannot miss any part of the whole PF.

In our experiments, we select 500 evenly distributed points in PF and let these points be P^* for each test instance with two objectives, and 990 points for each test instance with three objectives.

C. Instances With Various PS Shapes

We first compare MOEA/D-DE and NSGA-II-DE on six test instances: F1–F6. F1–F5 are two-objective instances, they have the same convex PF shape but their PS shapes are various nonlinear curves in the decision space. F6 has three objectives and its PS is a nonlinear 2-D surface.

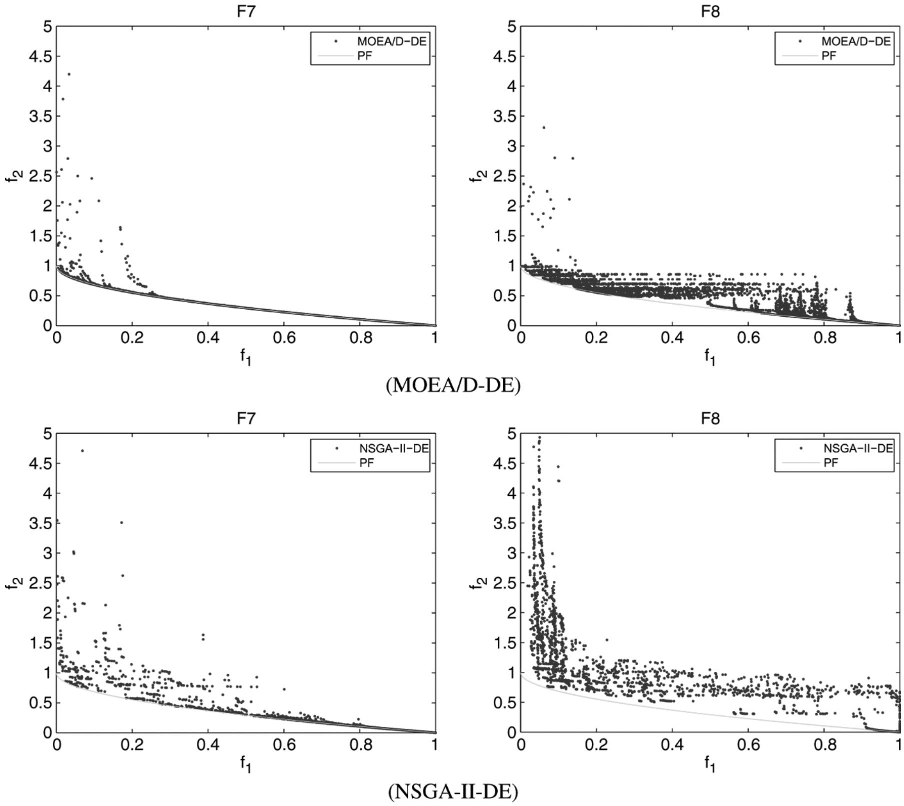


Fig. 8. Plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in the objective space on F7 and F8.

Fig. 2 shows the evolution of the average IGD-metric values of the population with the number of generations in two algorithms. Table II presents the minimum, mean, and standard deviation of the IGD-metric values of the 20 final populations. Since no single metric is always able to rank different MOEAs appropriately, Figs. 3 and 5 plot, in both the objective space and decision space, the distribution of the final populations with the lowest IGD-metric values obtained in 20 runs of each algorithm on each test instance. To show their distribution ranges, all the 20 final populations are also plotted together in Figs. 4 and 6.

It is clear from Fig. 2 that MOEA/D-DE is much more effective and efficient than NSGA-II-DE in reducing the IGD-metric values on all the test instances. Figs. 3 and 5 visually show that the final populations in MOEA/D-DE are significantly better than those in NSGA-II-DE in approximating both the PFs and PSs. NSGA-II-DE fails, within the given number of generations, in satisfactorily approximating the PFs and PSs in all the test instances except F1, which has the simplest PS shape. In contrast, MOEA/D-DE is able to find a good approximation in all the test instances except F5.

D. Instances With Many Local Pareto Fronts

There are no local PFs in the test instances used in Section IV-C. To study the global search ability of the two algorithms, we test both algorithms on F7 and F8. The only difference between these two instances and F1 lies in the setting of functions β_i. F7 and F8 have the same PF shape as F1 does. Their PS shapes are similar to that of F1. There are many local Pareto solutions in F7 and F8 since their $\sum_{i=1}^{m}\beta_i$'s have many local minima.

Fig. 2 shows that MOEA/D-DE performs much better than NSGA-II-DE in terms of IGD-metric. It can be observed from Figs. 7–10 that the final solutions obtained by MOEA/D-DE have better spread and convergence than those by NSGA-II-DE on these two instances. It is also clear that MOEA/D-DE fails in finding a good approximation on F8 and NSGA-II-DE could not approximate the PFs and PSs on both F7 and F8 within the given number of generations. F8 is harder than F7 for both algorithms, it could be because that $\sum_{i=1}^{m}\beta_i$ in F8 is harder than that in F7 for a DE to optimize.

These experimental results, in conjunction with the results on F1, suggest that the presence of many local Pareto optimal solutions would deteriorate the performance of both algorithms, and MOEA/D-DE could outperform NSGA-II-DE on problems with nonlinear PSs and many local Pareto optimal solutions.

E. Instance With Concave Parent Fronts

The PFs in all the test instances in the above two sections are convex. F9 used in this subsection has a concave PF and its PS is the same as that of F2.

Fig. 2 suggests that MOEA/D-DE outperforms NSGA-II-DE on this instance in terms of IGD-metric. The experimental results presented in Figs. 11–14, together with the results on F2 in Section IV-C show that the performance of MOEA/D-DE has become slightly worse in the case of concave PFs, but its solution quality is still acceptable. These results are not very surprising since MOEA/D-DE utilizes Tchebycheff decomposition

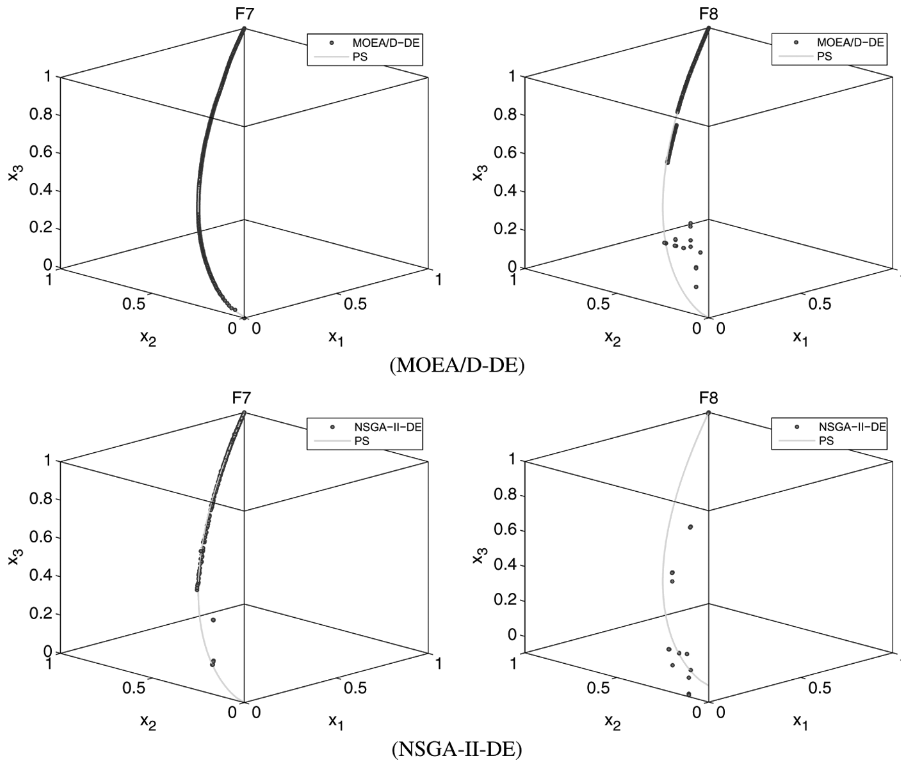


Fig. 9. Plots of the final populations with the lowest IGD-metric values found by MOEA/D-DE and NSGA-II-DE in 20 runs in the $x_1 - x_2 - x_3$ space on F7 and F8.

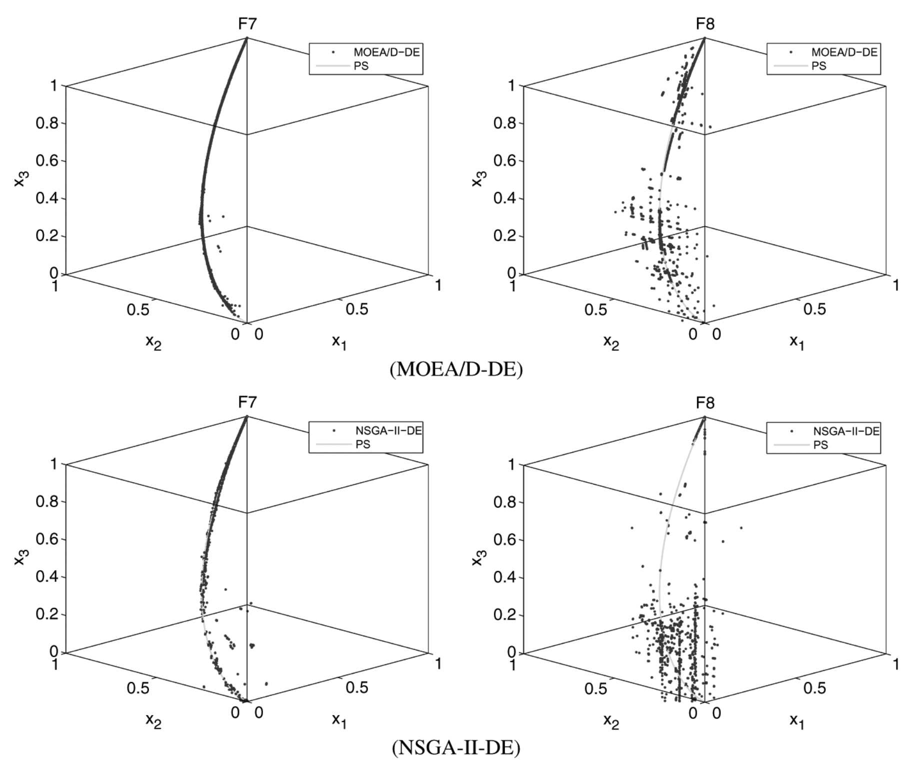


Fig. 10. Plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in the $x_1 - x_2 - x_3$ space on F7 and F8.

method, which is not very sensitive to PF shapes. It is also evident from Figs. 11–14 and Figs. 3–6 that a concave PF does hinder the performance of NSGA-II-DE very much if the PS is nonlinear as in F9.

F. Impacts of Parameter Settings

1) CR and F in MOEA/D-DE and NSGA-II-DE: CR and F are two control parameters in the DE operator for generating new solutions. To investigate the impacts of CR and F on the

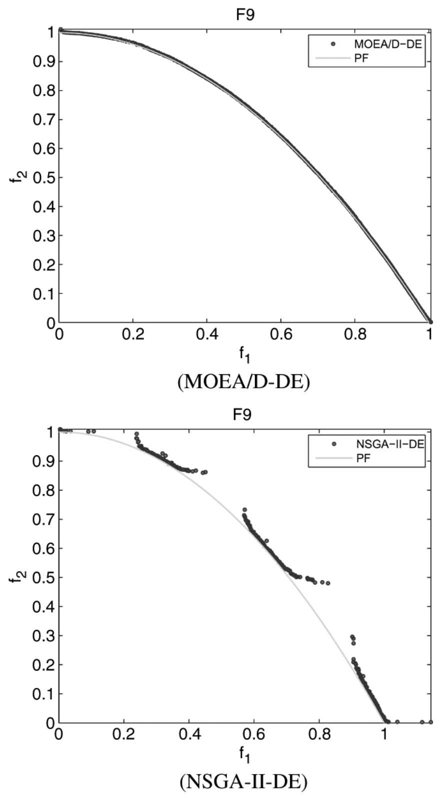


Fig. 11. Plots of the final populations with the lowest IGD-metric value found by MOEA/D-DE and NSGA-II-DE in the objective space on F9.

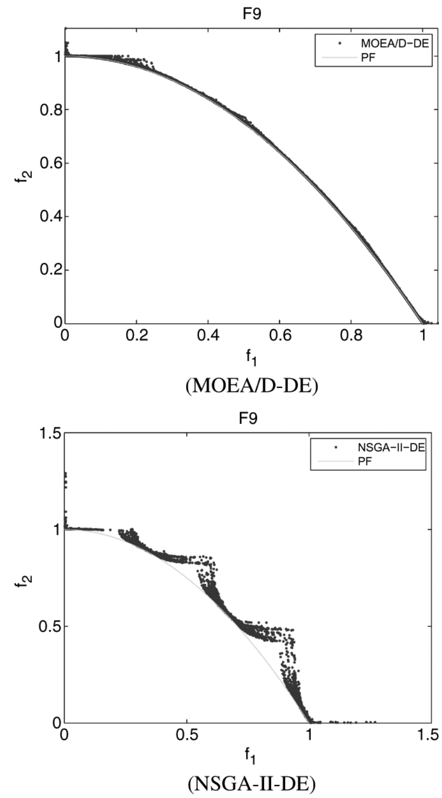


Fig. 12. Plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in the objective space on F9.

performances of both algorithms. We have tested 50 combinations of five values of F (i.e., 0.1, 0.25, 0.5, 0.75, and 1.0) and ten values of CR (i.e., 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0) in both algorithms on F3. All the other parameters remain the same as in Section IV-A. Each combination of CR and F has been tested 20 times in both algorithms.

Fig. 15 shows the mean IGD-metric values in both algorithms with 50 different combinations of CR and F. It is evident that MOEA/D-DE is less sensitive to the settings of CR and F under the ranges considered than NSGA-II-DE. It also suggests that the tuning of CR and F could not make NSGA-II-DE beat MOEA/D-DE on this instance, and thus the advantage of MOEA/D-DE over NSGA-II-DE may come from its decomposition strategy.

2) Setting of Weight Vectors: As shown in Section IV-C, MOEA/D-DE performs very poorly on F5. It could be due to the fact, as shown in Fig. 16, that the optimal solutions to two neighboring subproblems are not very close to each other under the setting of weight vectors described in Section IV-A. As a result, mating among the solutions to these neighboring problems makes little sense. To verify our analysis, we have chosen a set of 500 weight vectors such that the solutions to their corresponding subproblems are uniformly distributed along the PS in the decision space, set the number of generations be 1000 and keep all the other algorithmic parameters as the same as in Section IV-A. It is clear from Fig. 17 that MOEA/D-DE with this new setting of weight vectors can successfully solve F5. These experimental results suggest that the setting of weight vectors could be very important in MOEA/D-DE for dealing problems like F5. Adjusting the setting of weight vectors adaptively might be a practical solution for improving the performance of MOEA/D on this kind of problems.

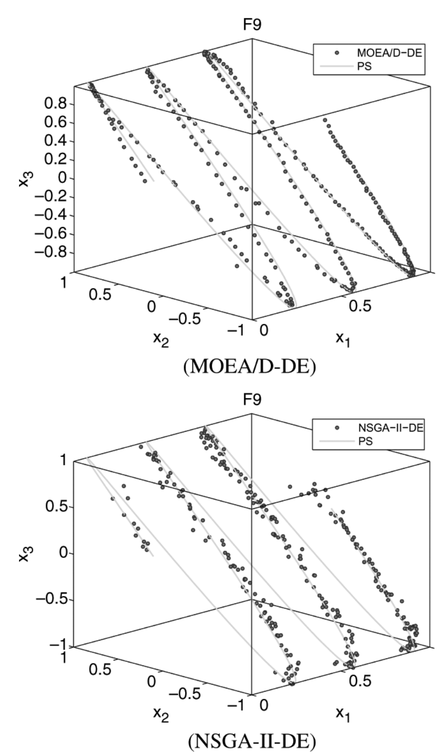


Fig. 13. Plots of the final populations with the lowest IGD-metric value found by MOEA/D-DE and NSGA-II-DE in the $x_1 - x_2 - x_3$ space on F9.

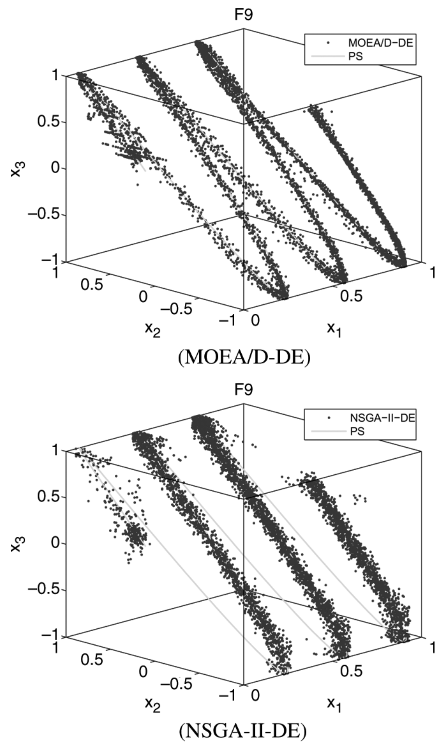


Fig. 14. Plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in the $x_1 - x_2 - x_3$ space on F9.

3) T and n_r in MOEA/D-DE: The concept of neighborhood relationship among the subproblems is a defining characteristic of MOEA/D [34]. The neighborhood size T should be, as recommended in [34], much smaller than the population size N. T is set to be 20 in Section IV-A. Two extra measures, taken in MOEA/D-DE to maintain the population diversity (please refer to the remarks in Section III-A), are implemented by using the control parameters δ and n_r. n_r should be set to be much smaller than T to ensure these measures work. A very natural question arises whether the neighborhood concept and these two extra measures are beneficial. To study this question, we set $T = N$ and $n_r = N$ in MOEA/D-DE, which will make δ no effect, and thus the neighborhood relationship and two extra measures will be removed from the algorithm. We keep all the other parameters in MOEA/D-DE as the same as in Section IV-A. We have tested MOEA/D-DE on F3 with this new setting.

Fig. 18 plots 20 final populations of F3 found by MOEA/D-DE with this setting. Compared with the results in Figs. 4 and 6, it is very clear that the performance of MOEA/D-DE has worsened significantly. This result indicates that the neighborhood relationship and the two extra measures are helpful.

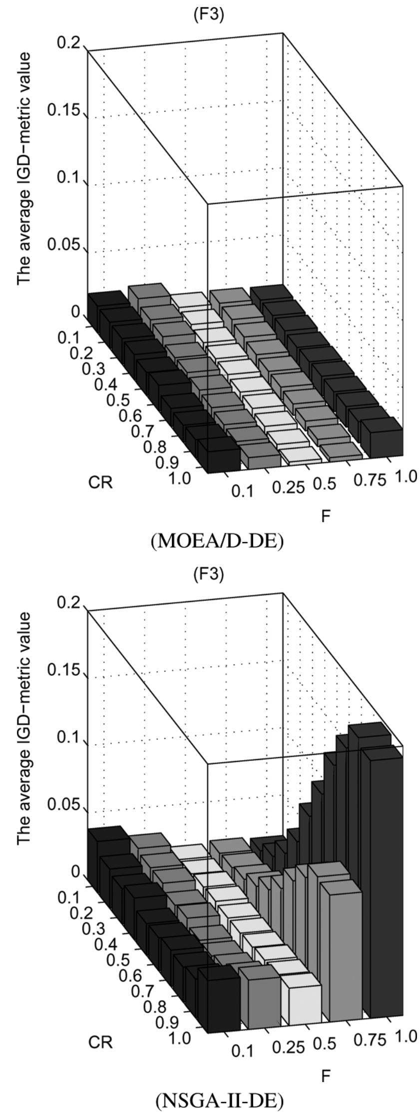


Fig. 15. Average IGD-values found by MOEA/D-DE and NSGA-II-DE with 50 different combinations of CR and F values on F3.

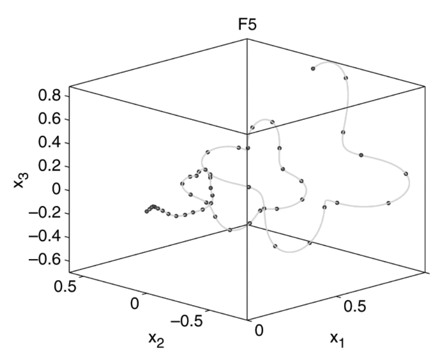


Fig. 16. Distribution of the optimal solutions of 50 subproblems with uniformly distributed weight vectors in $x_1 - x_2 - x_3$ space.

G. More Discussions

The Pareto-domination based selection in NSGA-II aims at driving the whole population towards the PS (PF). However, it has no direct control over the movement of each individual in its population and then it has no good mechanism to control the distribution of its computational effort over different ranges of the PF or PS. In contrast, MOEA/D directly defines a single objective optimization subproblem for each individual and then the computational effort can be evenly distributed among these subproblems. This could be one of the major reasons why MOEA/D-DE outperforms NSGA-II-DE on a set of continuous test instances with complicated PS shapes.

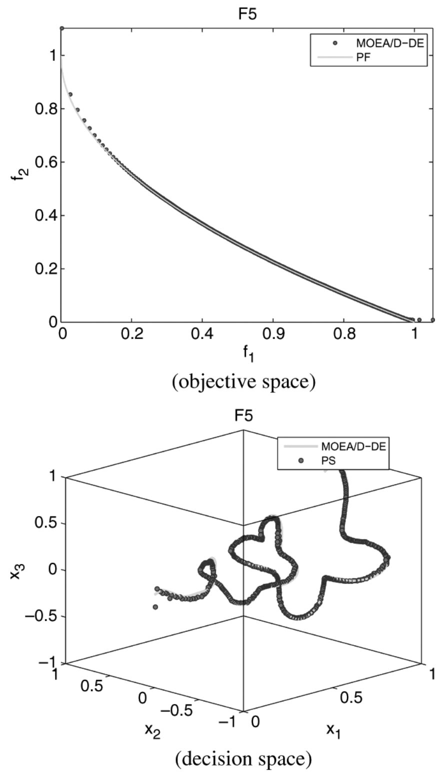


Fig. 17. Plots of the final solutions of F5 in the objective space and the $x_1 - x_2 - x_3$ space found by MOEA/D-DE, where 500 weight vectors are chosen such that the optimal solutions to their corresponding subproblems are uniformly distributed optimal solutions along the PS.

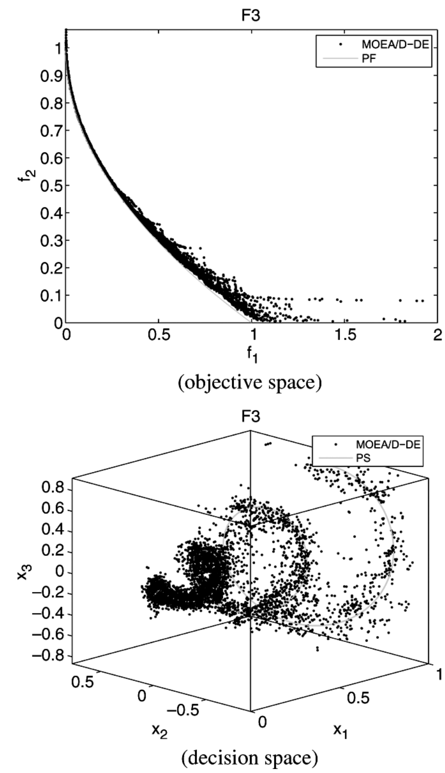


Fig. 18. Plots of the final solutions of F3 of 20 runs in the objective space and decision space by using MOEA/D-DE with $T = n_r = N = 300$.

An EA has two main components, i.e., selection and reproduction operators. We have shown that NSGA-II could not deal with complicated PSs very well if a DE operator and polynomial mutation are employed as reproduction operators. However, it does not imply that an NSGA-II with other reproduction operators is hopeless. In fact, the second author and his coworkers have very recently shown that although RM-MEDA, an EDA version of NSGA-II reported in [27], does not work well on the test instances introduced in this paper, a new implementation of RM-MEDA performs very similarly to MOEA/D-DE on these test instances.[5] In this new implementation, some measures are taken to balance the computational efforts to approximate different ranges of the PF. We also would like to point out that this version of RM-MEDA is much more complicated than MOEA/D-DE.

V. Conclusion

This paper has proposed a general class of continuous multiobjective optimization test instances with arbitrary prescribed PS shapes. These test instances could be used for studying the ability of MOEAs for dealing with complicated PS shapes. We have also proposed a new version of MOEA/D, i.e., MOEA/D-DE and compared it with NSGA-II with the same reproduction operators. The experimental results indicate that MOEA/D could significantly outperform NSGA-II on these test instances. It implies that decomposition based multiobjective evolutionary algorithms are very promising in dealing with complicated PS shapes.

The experimental results also suggest that the presence of many local Pareto optimal solutions in MOPs with complicated PSs could be very challenging for both algorithms. We have found that MOEA/D might not work very well if the solutions to neighboring subproblems are not very close in the decision space. Our experimental results reveal that a proper setting of the weight vectors could overcome this shortcoming. However, it is often hard, if it is not impossible, to know beforehand which setting is proper. A possible solution may be to tune weight vectors adaptively based on the information collected during the search. We have studied the impact of the control parameters in the DE operator on the performances of both algorithms and found that MOEA/D-DE is less sensitive to these parameters than NSGA-II-DE. The benefit of the neighborhood relationship and the two extra measures for maintaining the population diversity has also been experimentally studied.

We do not imply in this paper that MOEA/D is always superior to NSGA-II and other Pareto domination-based algorithms. The strengths and weaknesses of these algorithms should be thoroughly studied on test problems with different characteristics. Such study will definitely be helpful for one to choose and modify an algorithm for solving their problems. In the future, we plan to study the ability of MOEA/D for dealing with MOPs with many objectives, and those in uncertain and noisy environments [48], [49].

The source code of MOEA/D-DE can be downloaded from Q. Zhang's homepage: http://www.dces.essex.ac.uk/staff/qzhang/

[5]The results will be reported in a forthcoming working report soon.

ACKNOWLEDGMENT

The authors would like to thank the three anonymous reviewers for their very helpful and constructive comments, which have helped to improve the quality of this paper significantly.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MAA: Kluwer, 1999.

[2] M. Ehrgott, *Multicriteria Optimization*. Berlin, Germany: Springer, 2005.

[3] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proc. 1st Int. Conf. Genetic Algorithms*, Pittsburgh, PA, 1985, pp. 93–100.

[4] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[5] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1st Int. Conf. Evol. Comput.*, Piscataway, NJ, 1994, pp. 82–87.

[6] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[7] C. A. C. Coello, "An updated survey of GA-based multiobjective optimization techniques," *ACM Comput. Surv.*, vol. 32, no. 2, pp. 109–143, 2000.

[8] J. D. Knowles and D. W. Corne, "Local search, multiobjective optimization and the Pareto archived evolution strategy," in *Proc. 3rd Australia-Japan Joint Workshop Intell. Evol. Sys.*, Ashikaga, Japan, 1999, pp. 209–216.

[9] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Proc. Evol. Methods Design, Optimisation and Control With Applications to Industrial Problems (EUROGEN 2001)*, Athens, Greece, 2001, pp. 95–100.

[10] K. C. Tan, T. H. Lee, and E. F. Khor, "Evolutionary algorithm with dynamic population size and local exploration for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 5, no. 6, pp. 565–588, Dec. 2001.

[11] 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.

[12] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, 2003.

[13] H. Lu and G. G. Yen, "Rank-density-based multiobjective genetic algorithm and benchmark test function study," *IEEE Trans. Evol. Comput.*, vol. 7, no. 4, pp. 325–343, Aug. 2003.

[14] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[15] C. A. C. Coello, D. A. V. Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer, 2002.

[16] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications*, ser. Advanced Information and Knowledge Processing. Secaucus, NJ: Springer-Verlag, 2005.

[17] 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.

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

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

[20] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. Congr. Evol. Comput.—CEC '02*, Piscataway, NJ, 2002, vol. 1, pp. 825–830.

[21] V. L. Huang, A. K. Qin, K. Deb, E. Zitzler, P. N. Suganthan, J. J. Liang, M. Preuss, and S. Huband, "Problem definitions for performance assessment of multi-objective optimization algorithms," Nanyang Technological Univ., Singapore, 2007, Tech. Rep..

[22] E. Kasprzak and K. Lewis, "An approach to facilitate decision tradeoffs in pareto solution sets," *J. Eng. Valuation Cost Analysis*, vol. 3, no. 1, pp. 173–187, 2000.

[23] C. Hillermeier, *Nonlinear Multiobjective Optimization: A Generalized Homotopy Approach*. Basel, Switzerland: Birkhauser-Verlag, 2000.

[24] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On test functions for evolutionary multi-objective optimization," in *Proc. Parallel Problem Solving From Nature—PPSN VIII*, Birmingham, U.K., 2004, pp. 792–802.

[25] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Proc. Genetic and Evol. Comput. Conf.—GECCO'06*, Seattle, WA, 2006, pp. 1141–1148.

[26] H. Li and Q. Zhang, "A multi-objective differential evolution based on decomposition for multi-objective optimization with variable linkages," in *Proc. Parallel Problem Solving From Nature—PPSN IX*, Reykjavik, Iceland, 2006, pp. 583–592.

[27] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 41–63, Feb. 2008.

[28] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[29] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Trans. Syst., Man Cybern. C*, vol. 28, no. 3, pp. 392–403, 1998.

[30] Y. W. Leung and Y. Wang, "Multiobjective programming using uniform design and genetic algorithm," *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, no. 3, pp. 293–304, Aug. 2000.

[31] Y. Jin, T. Okabe, and B. Sendhoff, "Adapting weighted aggregation for multiobjective evolutionary strategies," in *Proc. Evol. Multi-Criterion Optimization—EMO' 01*, Zurich, Switzerland, 2001, pp. 96–110.

[32] 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.

[33] E. J. Hughes, "Multiple single objective pareto sampling," in *Proc. Congr. Evol. Comput.—CEC '03*, Canberra, Australia, 2003, vol. 4, no. 8–12, pp. 2678–2684.

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

[35] A. Jaszkiewicz, "Genetic local search for multi-objective combinatorial optimization," *Eur. J. Oper. Res.*, vol. 137, pp. 50–71, 2002.

[36] A. Jaszkiewicz, "On the computational efficiency of multiple objective metaheuristics: The knapsack problem case study," *Eur. J. Oper. Res.*, vol. 158, pp. 418–433, 2004.

[37] E. J. Hughes, "Evolutionary many-objective optimization: Many once or one many?," in *Proc. Congr. Evol. Comput.—CEC '05*, Edinburgh, U.K., 2005, pp. 222–227.

[38] H. Ishibuchi, T. Doi, and Y. Nojima, "Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms," in *Proc. Parallel Problem Solving From Nature—PPSN IX*, Reykjavik, Iceland, 2006, pp. 493–502.

[39] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *Proc. Congr. Evol. Comput.—CEC '07*, Singapore, 2007, pp. 3944–3951.

[40] H. Ishibuchi, Y. Nojima, K. Narukawa, and T. Doi, "Incorporation of decision maker's preference into evolutionary multiobjective optimization algorithms," in *Proc. Genetic Evol. Comput. Conf.—GECCO '06*, New York, 2006, pp. 741–742.

[41] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," in *Proc. Genetic Evol. Comput. Conf.—GECCO '06*, New York, 2006, pp. 635–642.

[42] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution," in *Proc. Joint Austral. Conf. Artif. Intell.*, G. Webb and X. Yu, Eds., Cairns, Australia, 2004, pp. 861–872.

[43] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, ser. Natural Computing. Secaucus, NJ, USA: Springer-Verlag, 2005.

[44] N. K. Madavan, "Multiobjective optimization using a pareto differential evolution approach," in *Proc. Congr. Evol. Comput.—CEC '02*, Honolulu, HI, 2002, pp. 1145–1150.

[45] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Proc. Congr. Evol. Comput.—CEC '05*, Edinburgh, U.K., 2005, pp. 239–246.

[46] V. L. Huang, A. K. Qin, P. N. Suganthan, and M. F. Tasgetiren, "Multi-objective optimization based on self-adaptive differential evolution," in *Proc. Congr. Evol. Comput.—CEC '07*, Singapore, 2007, pp. 3601–3608.

[47] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[48] M. Farina, K. Deb, and P. Amato, "Dynamic multiobjective optimization problems: Test cases, approximations, and applications," *IEEE Trans. Evol. Comput.*, vol. 8, no. 5, pp. 425–442, Oct. 2004.

[49] C. K. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 354–381, Jun. 2007.

Hui Li received the B.Sc. and M.Sc. degrees in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 1999 and 2002, respectively, and the Ph.D. degree in computer science from University of Essex, Colchester, U.K., in 2008.

He is currently a Research Fellow in the Automated Scheduling, Optimization, and Planning (ASAP) Research Group, School of Computer Science, University of Nottingham. His research interests include multiobjective optimization, evolutionary computation, metaheuristics, and scheduling algorithms.

Qingfu Zhang (M'01–SM'06) received the B.Sc. degree in mathematics from Shanxi University, Shanxi, 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 currently a Reader in the Department of Computing and Electronic Systems, University of Essex, Colchester, U.K. From 1994 to 2000, he was with National University of Defence Science and Technology (China), Hong Kong Polytechnic University (Hong Kong), the German National Research Centre for Information Technology (now Fraunhofer-Gesellschaft, Germany), and the University of Manchester Institute of Science and Technology (U.K.). His main research areas are evolutionary computation, optimization, neural networks, data analysis, and their applications.

Dr. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B and a Guest Editor of a forthcoming special issue on evolutionary algorithms based on probabilistic models in the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.