Comparison Between NSGA-II and MOEA/D on a Set of Multiobjective Optimization Problems with Complicated Pareto Sets

by

Hui Li and Qingfu Zhang, *Senior Member, IEEE* [1]

Technical Report CES-476
October 2007

[1]H. Li and Q. Zhang are with Department of Computing and Electronic Systems, University of Essex,Wivenhoe Park, Colchester, CO4 3SQ, U.K (e-mail: {qzhang, hlil}@essex.ac.uk).

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 MOEAs for dealing with complicated PS shapes. It also proposes a new version of MOEA/D, i.e., MOEA/D-DE and compares it 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.

keywords: multiobjective optimization, test problems, Pareto Set, decomposition, evolutionary algorithm, Pareto optimality.

1 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 (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 $u_i \leq v_i$ for all $i = 1, \ldots, m$, and $u \neq v$.[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 curve in R^n while the PS of a continuous MOP with three objectives is a piecewise continuous surface. This is the so-called regularity property of continuous MOPs.

Recent years have witnessed significant progress in the development of evolutionary algorithms (EA) for multiobjective optimization problems (MOP) [3–14]. Multiobjective evolutionary algorithms (MOEA) 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 [15]) that the geometrical shapes of the PF, among other characteristics of a 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 [16–19], 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 [17] with two objectives are in a line segment defined by

$$\begin{gathered} 0 \leq x_1 \leq 1, \\ x_2 = x_3 = \ldots = x_n = 0, \end{gathered}$$

[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”.

and the PSs of the DTLZ test instances [18] with three objectives are subsets of

$$0 \le x_1, x_2 \le 1,$$
$$x_3 = \ldots = x_n = 0.5.$$

There is no reason that real world problems have such simple PSs. Observing these oversimplified PSs in 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 [20]. They have constructed several test instances with complicated PSs. However, their test instances have two objectives and two decision variables. Very recently, Deb *et al.* [21] and Huband *et al.* [15] 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 test instances constructed in [21] and [15] 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 [21] and [15]. 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 [20] [21], we have recently proposed several continuous test instances with variable linkages/complicated PSs [22], in which the PS shapes could be easily described. These test instances have also been used in [23] 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 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, 8–10]. 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] [10] [24]. Arguably, NSGA-II is the most popular Pareto dominance based MOEAs. The characteristic feature of NSGA-II is its fast non-dominated sorting procedure for ranking solutions in its selection.

A Pareto optimal solution to a MOP could be an optimal solution of a scalar 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 scalar objective optimization subproblems. This is a basic idea behind many traditional mathematical programming methods for approximating the PF [1]. A very small number of MOEAs adopt this idea to some extent [25–30], among them MOEA/D is a very recent one [30]. MOEA/D attempts to optimize these subproblems simultaneously. The 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 only from its neighboring subproblems.

We believe that comparison studies between MOEAs based on Pareto dominance and those using decomposition 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 is proposed.
- a new implementation of MOEA/D with a DE operator and polynomial mutation has been proposed, in which extra measures have been introduced for maintaining population diversity.
- experiments have been conducted to compare MOEA/D and NSGA-II with the same DE operator and mutation 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 and provides a theorem about their Pareto set shapes. Section III proposes MOEA/D-DE, a new implementation of MOEA/D with a DE operator and polynomial mutation, and 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.

2 Multiobjective Test Problem with Prescribed Pareto Set

In our proposed generic continuous test problem, the search 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 $i = 1, \dots, n$. 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

- $x = (x_1, \dots, x_n) \in \Omega$, $x_I = (x_1, \dots, x_{m-1})$ and $x_{II} = (x_m, \dots, x_n)$ 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

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

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

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

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

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

$$\{(\alpha_1(x_I), \ldots, \alpha_m(x_I))|x_I \in D\}.$$

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 $x_I \in D$ 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 $x_I \in D$ and $x_{II} = g(x_I)$, then, by [i]

$$F(x) = (\alpha_1(x_I), \ldots, \alpha_m(x_I))$$

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

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

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

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 "only if " part.

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

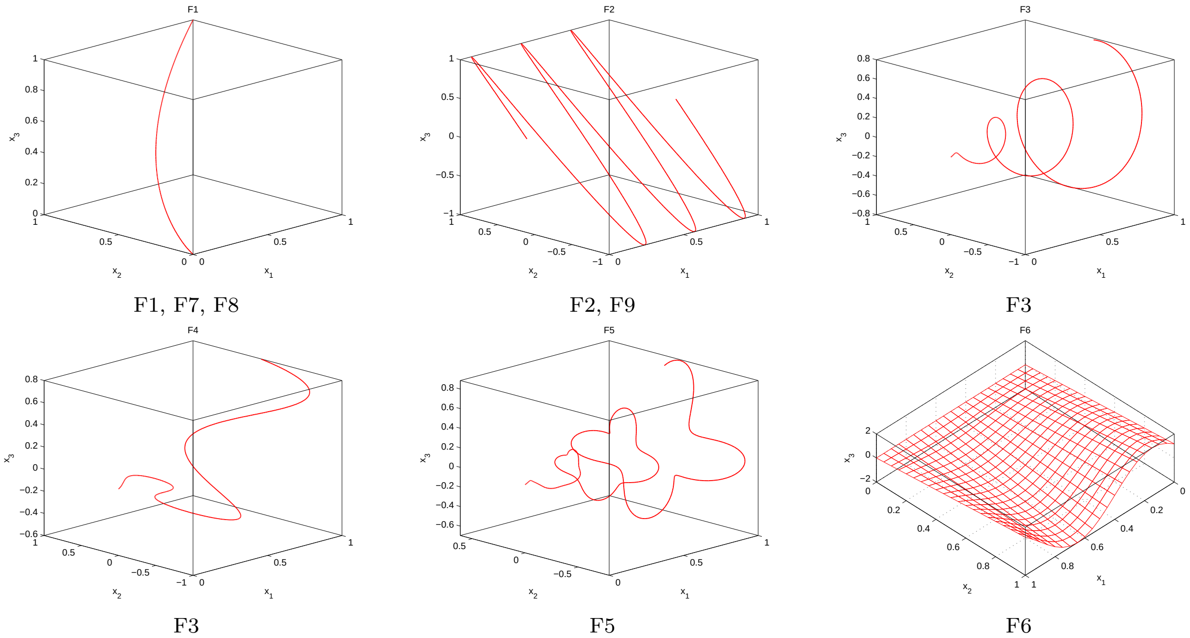


Figure 1: The projections of PSs of F1-F6 on x_1-x_2-x_3 space

- If the dimensionality of D is $m-1$ as in all the test instances constructed in this paper (in fact, $D = \prod_{i=1}^{m-1}[a_i, b_i]$ in all of them), the PS and PF of the 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 [18] [15], 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. Table 1 lists 9 test instances generated from (2) and (3), and Figure 1 plots the projections of their PSs in the space of x_1, x_2 and x_3. In the following, taking three test instances in Table 1 for example, we explain how the genetic test problem could be instantiated.

F2

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

Table 1: 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 $J_1 = \{j \mid j$ is odd and $2\le j\le n\}$ and $J_2 = \{j \mid j$ 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	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\sum_{j\in J_1}(x_j - 0.3x_1(x_1\cos(4(6\pi x_1 + \frac{j\pi}{n})) + 2)\sin(6\pi x_1 + \frac{j\pi}{n}))^2$ $f_2 = 1-\sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\sum_{j\in J_2}(x_j - 0.3x_1(x_1\cos(4(6\pi x_1 + \frac{j\pi}{n})) + 2)\cos(6\pi x_1 + \frac{j\pi}{n}))^2$ where J_1 and J_2 are the same as those of F1. Its PS is $x_j = \begin{cases} 0.3x_1(x_1\cos(4(6\pi x_1 + \frac{j\pi}{n})) + 2)\cos(6\pi x_1 + \frac{j\pi}{n}) & j\in J_1 \\ 0.3x_1(x_1\cos(4(6\pi x_1 + \frac{j\pi}{n})) + 2)\sin(6\pi x_1 + \frac{j\pi}{n}) & j\in J_2\end{cases}$	$[0,1]\times[-1,1]^{n-1}$
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 $J_1 = \{j \mid 3\le j\le n$, and $j-1$ is a multiplication of 3$\}$, $J_2 = \{j \mid 3\le j\le n$, and $j-2$ is a multiplication of 3$\}$, $J_3 = \{j \mid 3\le j\le n$, 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	$f_1 = x_1 + \frac{4}{\lvert J_1\rvert}(2\sum_{j\in J_1} y_j^2 - \prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 1)$ $f_2 = 1-\sqrt{x_1} + \frac{4}{\lvert J_2\rvert}(2\sum_{j\in J_2} y_j^2 - \prod_{j\in J_2}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 1)$ 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 $x_j = \sin(6\pi x_1 + \frac{j\pi}{n}), j=2,\ldots,n.$	$[0,1]\times[-1,1]^{n-1}$

- $g(x_1) = (g_2(x_1), \dots, g_n(x_1))$ and
$$g_j(x_1) = \sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, \dots, n;$$
- β_1 and β_2 are functions from R^{n-1} to R^+. Let $y_{2:n} = (y_2, \dots, y_n) \in R^{n-1}$,
$$\begin{aligned}\beta_1(y_{2:n}) &= \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2\\ \beta_2(y_{2:n}) &= \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2\end{aligned}$$
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 F2 in the objective space is:
$$\{(x_1, 1-\sqrt{x_1}) \in R^2 | 0 \le x_1 \le 1\}$$
and its PS is:
$$x_{II} = g(x_1), \quad 0 \le x_1 \le 1.$$

F6

- The decision space $\Omega = [0,1]^2 \times [-2,2]^{n-2}$, $x_I = (x_1, x_2)$ and $x_{II} = (x_3, \dots, x_n)$;
- $$\begin{aligned}\alpha_1(x_I) &= \cos(0.5\pi x_1)\cos(0.5\pi x_2)\\ \alpha_2(x_I) &= \cos(0.5\pi x_1)\sin(0.5\pi x_2)\\ \alpha_3(x_I) &= \sin(0.5\pi x_1)\end{aligned}$$
- $g(x_I) = (g_2(x_I), \dots, g_n(x_I))$ and
$$g_j(x_I) = 2x_2\sin(2\pi x_1 + \frac{j\pi}{n}), j = 3, \dots, n;$$
- β_1 and β_2 are functions from R^{n-2} to R^+. Let $y_{3:n} = (y_3, \dots, y_n) \in R^{n-2}$,
$$\begin{aligned}\beta_1(y_{3:n}) &= \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2\\ \beta_2(y_{3:n}) &= \frac{2}{|J_2|}\sum_{j\in J_2} y_j^2\\ \beta_3(y_{3:n}) &= \frac{2}{|J_3|}\sum_{j\in J_3} y_j^2\end{aligned}$$

where

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

By Theorem 1, the PF of F6 is:

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

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 $x_{II} = (x_2, \dots, x_n)$;
- $\alpha_1(x_1) = x_1$ and $\alpha_2(x_1) = 1 - \sqrt{x_1}$;
- $g(x_1) = (g_2(x_1), \dots, g_n(x_1))$ and

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

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

$$\begin{aligned}
\beta_1(y_{2:n}) &= \frac{4}{|J_1|}(2\sum_{j \in J_1} y_j^2 - \prod_{j \in J_1} \cos(\frac{40\pi y_i}{\sqrt{j}}) + 1) \\
\beta_2(y_{2:n}) &= \frac{4}{|J_2|}(2\sum_{j \in J_2} y_j^2 - \prod_{j \in J_2} \cos(\frac{40\pi y_i}{\sqrt{j}}) + 1)
\end{aligned}$$

  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:

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

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.

3 Algorithms in Comparison

3.1 MOEA/D with DE

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

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

where $\lambda = (\lambda_1, \ldots, \lambda_m)^T$ is a weight vector, i. e., $\lambda_i \geq 0$ for all $i = 1, \ldots, m$ and $\sum_{i=1}^m \lambda_i = 1$. $z^* = (z_1^*, \ldots, z_m^*)^T$ is the reference point, i. e., $z_i^* = \min\{f_i(x)|x \in \Omega\}$[2] for each $i = 1, \ldots, m$.

It is well-known that 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 (1). Let $\lambda^1, \ldots, \lambda^N$ be a set of weight vectors. Correspondingly, we have N single objective optimization subproblems where the i-th subproblem is (5) with $\lambda = \lambda^i$. If N is reasonably large and $\lambda^1, \ldots, \lambda^N$ are properly selected, then the optimal solutions to these subproblems will provide a good approximation to the PF or PS of (1).

MOEA/D attempts to optimize these N single optimization problems simultaneously instead of solving (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 i-th 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 [30]. A simple implementation of MOEA/D, called hereafter MOEA/D-SBX, in which the SBX operator was used for generating new solutions, has been tested on ZDT test instances. The experimental results have shown that MOEA/D-SBX performs well on these test instances. 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 (a) the population in MOEA/D-SBX may lose diversity, which is needed for exploring the search space effectively, at the early stage of the search when applied to MOPs with complicated PSs, and (b) the SBX operator often generates inferior solutions in MOEAs, as shown in the recent experiments in [21] [31]. 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 DE operator and a polynomial mutation operator for producing new solutions, and it has extra measures for maintaining population diversity. The Tchebycheff approach is used in MOEA/D-DE for decomposing the MOP (1).

At each generation, MOEA/D-DE maintains:

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

- a population of N points $x^1, \dots, x^N \in \Omega$, where x^i is the current solution to the i-th subproblem;
- $FV^1, \dots, FV^N$, where FV^i is the F-value of x^i, i.e., $FV^i = F(x^i)$ for each $i = 1, \dots, N$;
- $z = (z_1, \dots, z_m)^T$, where z_i is the best value found so far for objective f_i;

The algorithm works as follows:

Input:
- MOP (1);
- a stopping criterion;
- N : the number of the subproblems considered in MOEA/D;
- a set of N weight vectors: $\lambda^1, \dots, \lambda^N$;
- 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: $\{x^1, \dots, x^N\}$.
- Approximation to the PF: $\{F(x^1), \dots, F(x^N)\}$.

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 $i = 1, \dots, N$, set $B(i) = \{i_1, \dots, i_T\}$ where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.
Step 1.2 Generate an initial population $x^1, \dots, x^N$ by uniformly randomly sampling from Ω. Set $FV^i = F(x^i)$.
Step 1.3 Initialize $z = (z_1, \dots, z_m)$ by setting $z_j = \min_{1 \le i \le N} f_j(x^i)$.

Step 2 Update

For $i = 1, \dots, N$, do
Step 2.1 Selection of Mating/Update Range: Uniformly randomly generate a number $rand$ from $(0, 1)$. Then set

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

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 reset to be a randomly selected value inside the boundary.
Step 2.4 Update of z: For each $j = 1, \dots, m$, 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 $g(y|\lambda^j, z) \leq g(x^j|\lambda^j, z)$, 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 Criteria If stopping criteria is satisfied, then stop and output EP. Otherwise go to **Step 2**.

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

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

where CR and F are two control parameters.

The polynomial mutation in Step 2 generates $y = (y_1, \ldots, y_n)$ from $\bar{y}$ in the following way:

$$y_i = \begin{cases} \bar{y}_i + \sigma_i \times (b_i - a_i) & \text{with probability } p_m, \\ \bar{y}_i & \text{with probability } 1 - p_m, \end{cases} \tag{7}$$

with

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

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_i and b_i are the lower upper bounds of the i-th 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.

- Unlike MOEA/D-SBX, MOEA/D-DE allows three parent solutions, with a low probability $1 - \delta$, mate without any restriction (Step 2.1 and Step 2.2), In such a way, a very wide range of child solutions could be generated due to 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.

- DE operators often outperform other genetic operators in single objective optimization. Furthermore, if CR is set to be 1 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.

3.2 NSGA-II with DE

Comparing qualities of different solutions in NSGA-II is based on their non-domination ranks and crowded distances, which can be obtained by a fast sorting algorithm proposed in [10]. The lower non-domination rank of a solution is, the better it is. If two solutions have the same non-domination rank, NSGA-II prefers the solution with the shorter crowded distance. NSGA-II-DE used in our experimental studies is the same as NSGA-II-SBX in [10] 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.

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 [31–33] have been proposed for dealing with rotated MOPs or MOPs with nonlinear variable linkages. There are no big difference among these variants. None of them employs mutation after DE operators. Our pilot experimental studies have shown that the polynomial mutation operator does slightly improve the algorithm performance, particularly, on MOPs with complicated PSs.

4 Experimental Results

4.1 Parameter Setting

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

4.1.1 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.

4.1.2 The number of decision variables

It is set to be 30 in F1-F5 and F9, and 10 in F6, F7 and F8.

4.1.3 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.

4.1.4 The population size and weight vectors in MOEA/D-DE

They are controlled by a parameter H. More precisely, $\lambda^1, \ldots, \lambda^N$ are all the weight vectors in which each individual weight takes a value from

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

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.

4.1.5 The population size in NSGA-II-DE

Its setting is the same as in MOEA/D-DE.

4.1.6 Other control parameters in MOEA/D-DE

- $T = 20$,
- $\delta = 0.9$,
- $n_r = 2$.

Table 2: The IGD 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

4.2 Performance Metric

The inverted generational distance (IGD) [34] 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:

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

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, $D(P^*, P)$ could measure both the diversity and convergence of P in a sense. To have a low value of $D(P^*, P)$, 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 2 objectives, and 900 points for each test instance with 3 objectives.

4.3 Performance of Two Algorithms on Instances with Various PS Shapes

We first compare MOEA/D-DE and NSGA-II-DE on 6 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.

Figure 2 shows the evolution of the average IGD-metric of the population with the number of generations in two algorithms. Table 2 presents the minimum, mean and standard deviation of the IGD-metric values of the 20 final populations. Since no single metric is

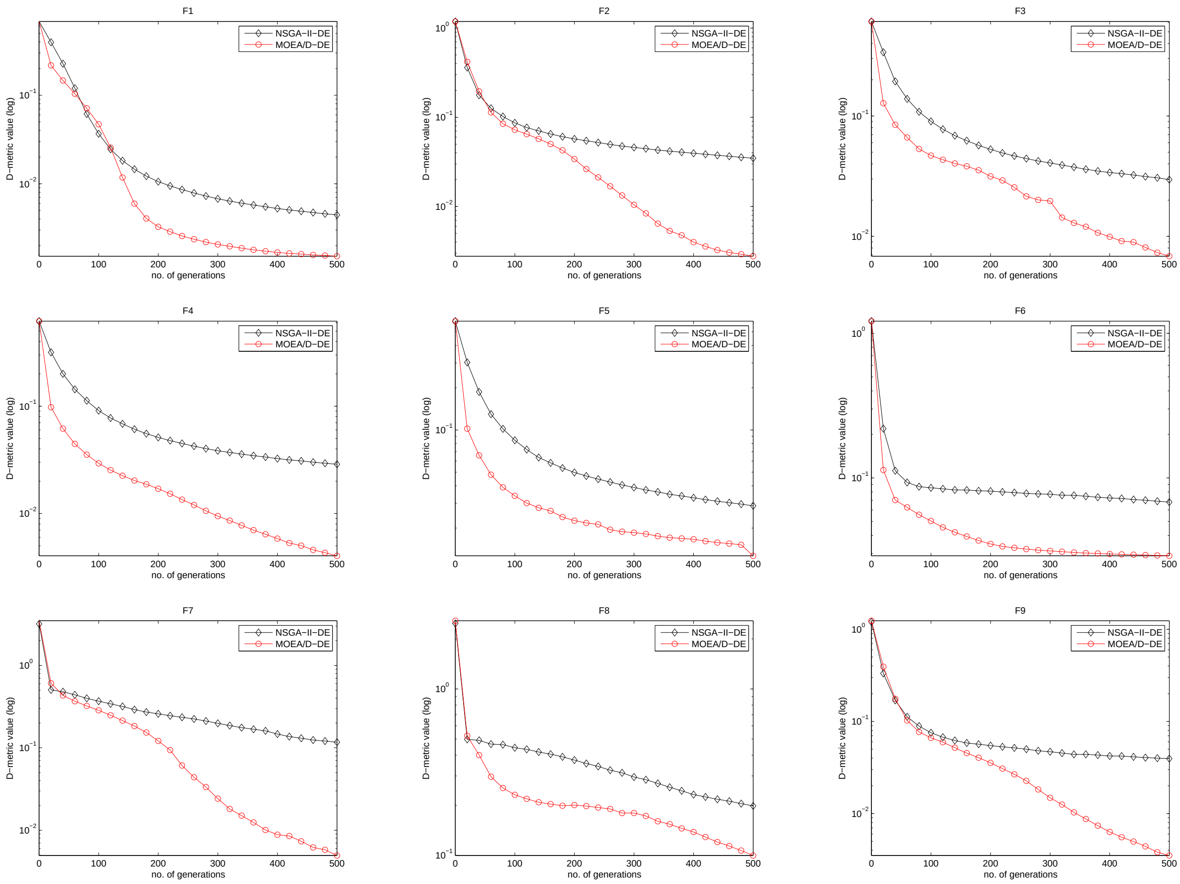


Figure 2: The evolution of IGD values vs. the number of generations.

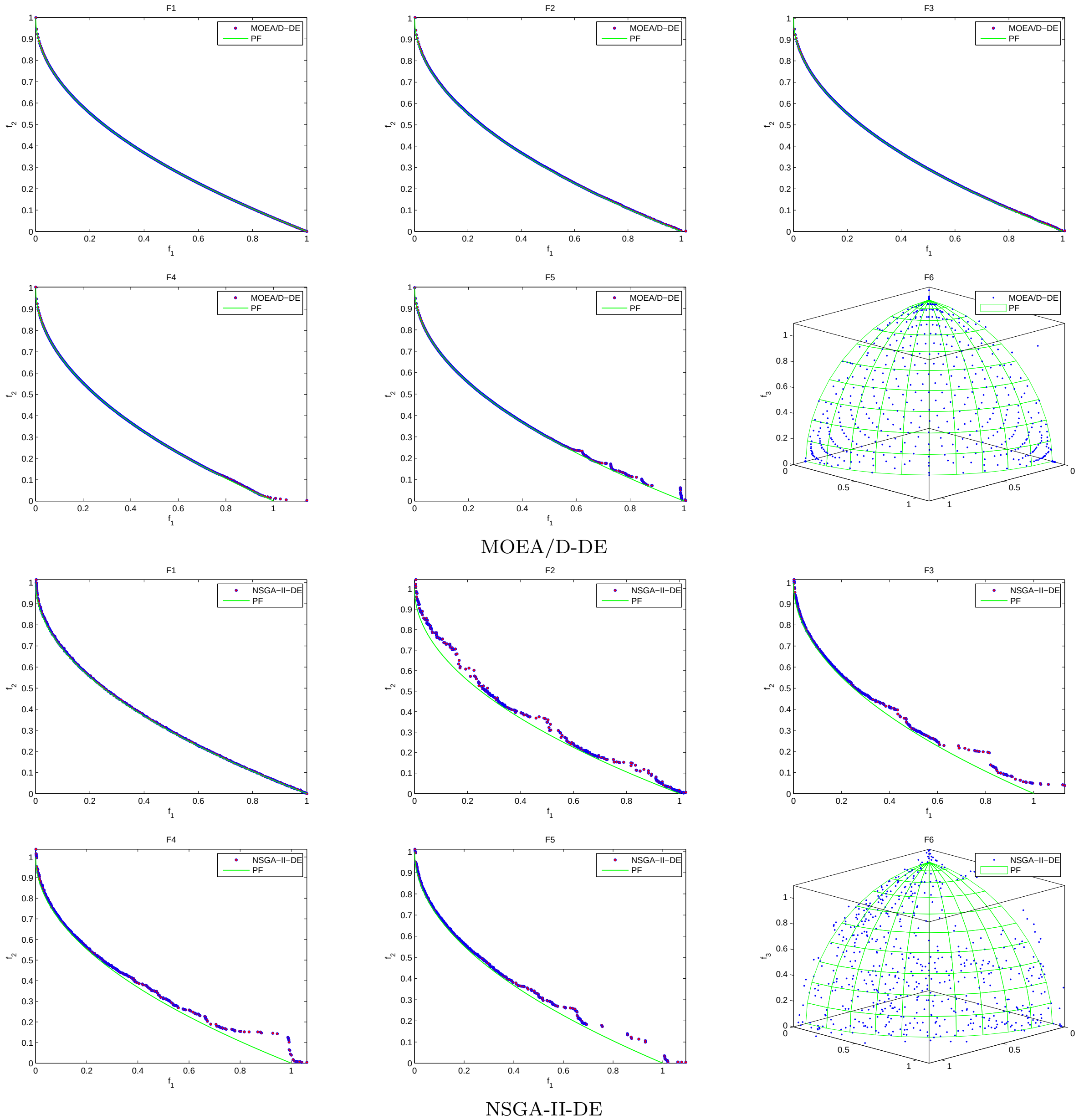
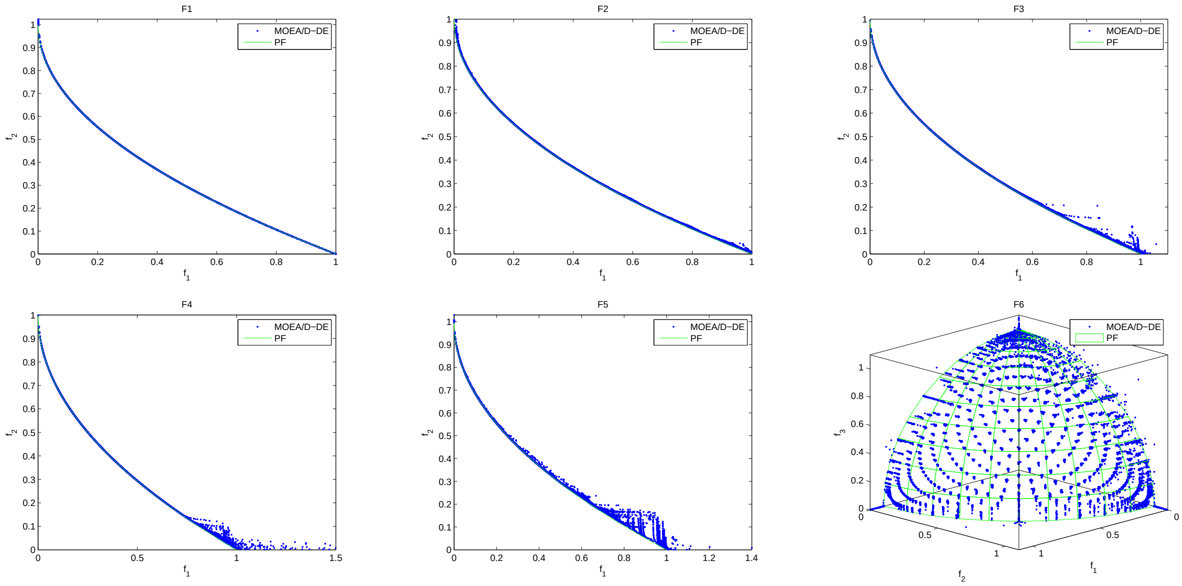
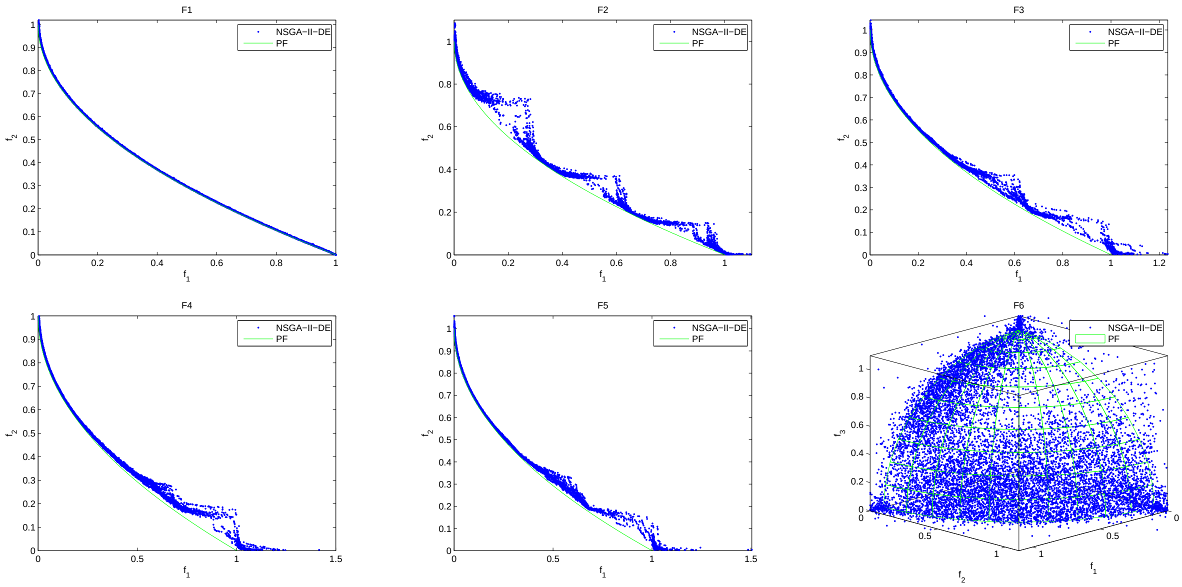


Figure 3: The plots of the final populations with the lowest values found by MOEA/D-DE and NSGA-II-DE in 20 runs in the objective space on F1-F6.



MOEA/D-DE



NSGA-II-DE

Figure 4: The plots of all the 20 final populations produced by MOEA/D-DE and NSGA-II-DE in the objective space on F1-F6.

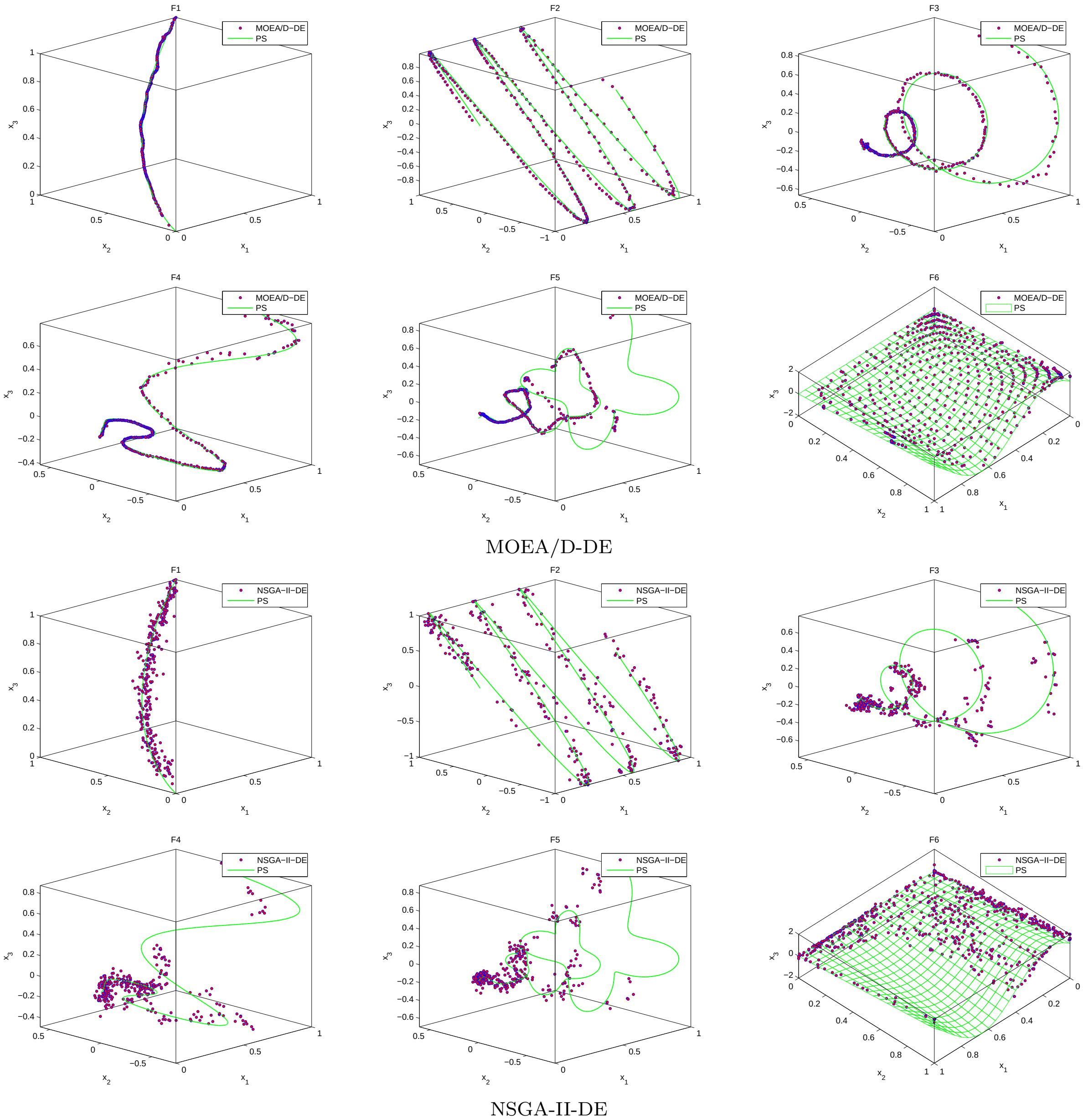
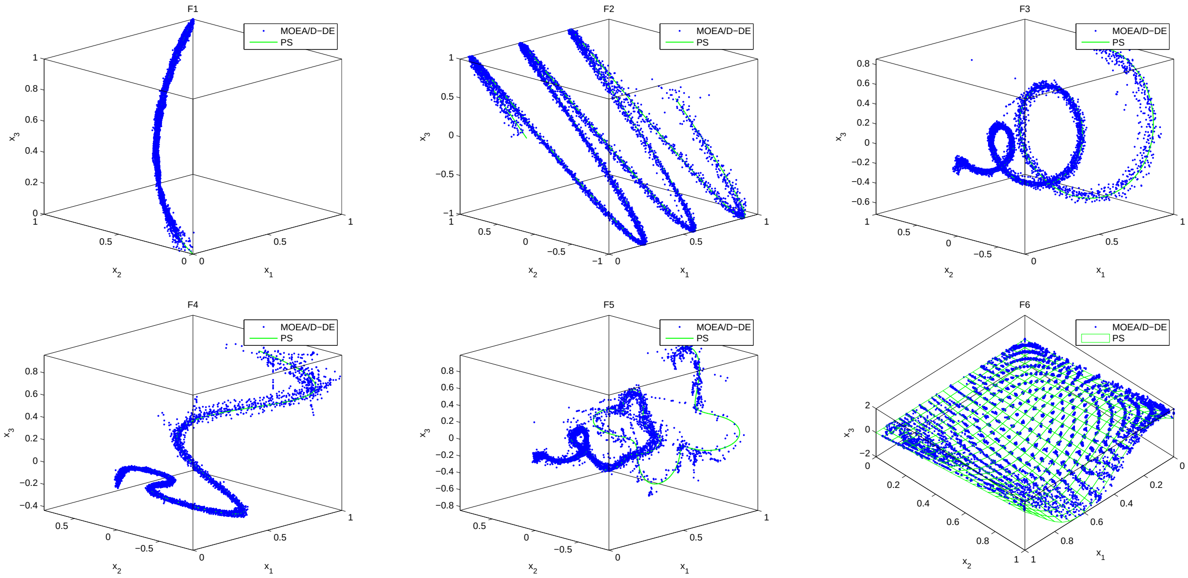
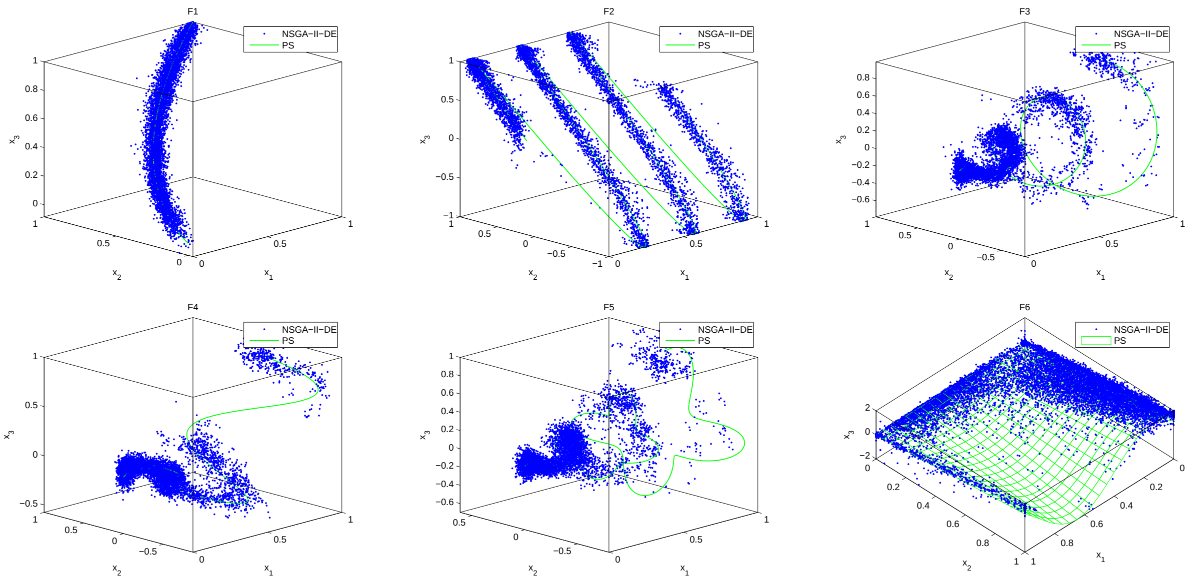


Figure 5: The plots of the final populations with the lowest IGD-value found by MOEA/D-DE and NSGA-II-DE in 20 runs in x_1-x_2-x_3 space on F1-F6.



MOEA/D-DE



NSGA-II-DE

Figure 6: The plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in x_1-x_2-x_3 space on F1-F6.

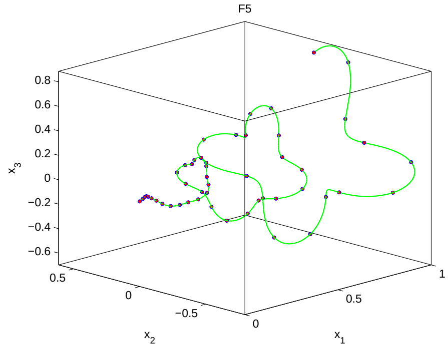


Figure 7: The distribution of the optimal solutions of 50 subproblems with uniformly distributed weight vectors in x_1-x_2-x_3 space.

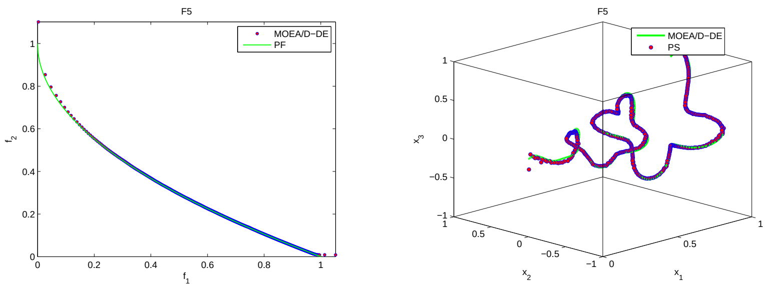


Figure 8: The plots of the final solutions of F5 in x_1-x_2-x_3 space and the objective space found by MOEA/D-DE, where 500 weight vectors are chosen such that the optimal solutions to their corresponding subproblems are uniformly distributed optimal solutions along the PS.

always able to rank different multiobjective evolutionary algorithms appropriately, Figures 3 and 5 plot, in both the objective space and decision space, the distribution of the final population with lowest IGD-metric 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 Figures 4 and 6.

It is clear from Figure 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. Figures 3 and 5 visually show that the final populations in MOEA/D-DE are significantly better than those in NSGA-II 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.

F5 is the hardest among these test instances and MOEA/D fails on it. It could be due to the fact, as shown in Figure 7 that the optimal solutions to two neighboring subproblems are not very close to each other under our setting of weight vectors. As a result, mating among 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 generation be 1000 and keep other control parameters as they are. It is clear from Figure 8 that MOEA/D-DE with this setting of weight vectors can successfully solve F5. This experimental result suggests that a good decomposition of the MOP plays a key role in MOEA/D-DE.

4.4 Instances with many local Pareto fronts

There are no local PFs in the test instances used in Section 4.3. 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 g_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.

Figure 2 shows that MOEA/D-DE performs much better than NSGA-II-DE in terms of IGD-metric. It can be observed from Figures 9-11 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.

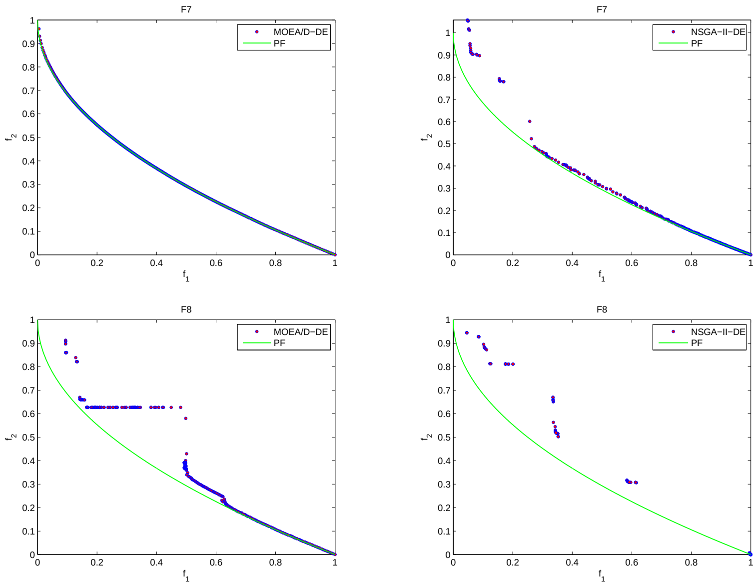


Figure 9: The plots of the final populations with the lowest IGD values found by two algorithms in 20 runs in the objective space on F7 and F8.

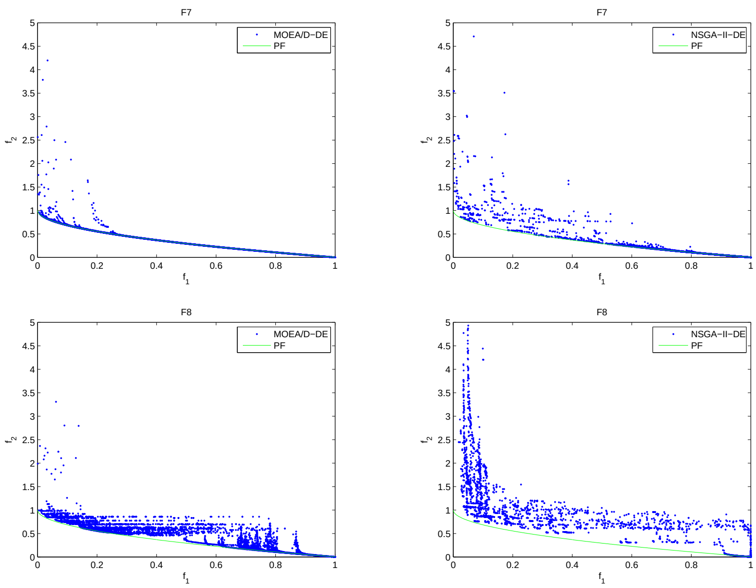


Figure 10: The plots of the final populations with the lowest IGD values found by two algorithms in the objective space on F7 and F8.

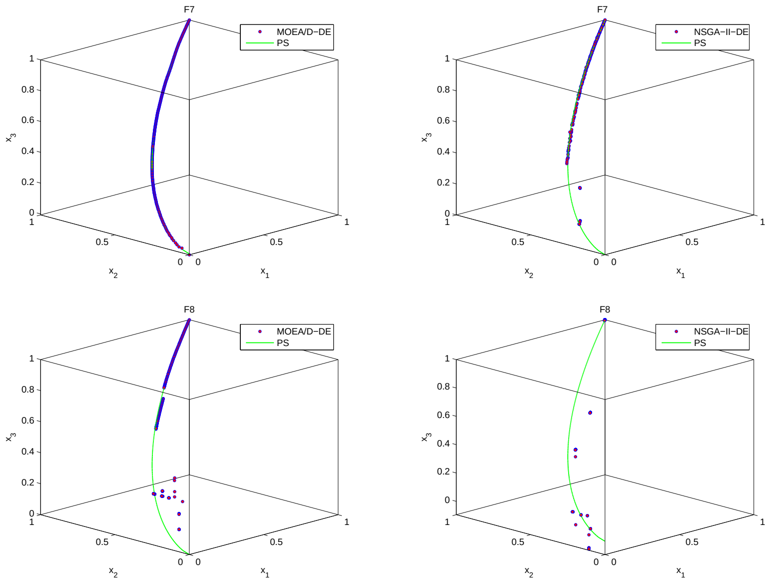


Figure 11: The plots of the final populations with the lowest IGD values found by two algorithms in 20 runs in x_1-x_2-x_3 space on F7 and F8.

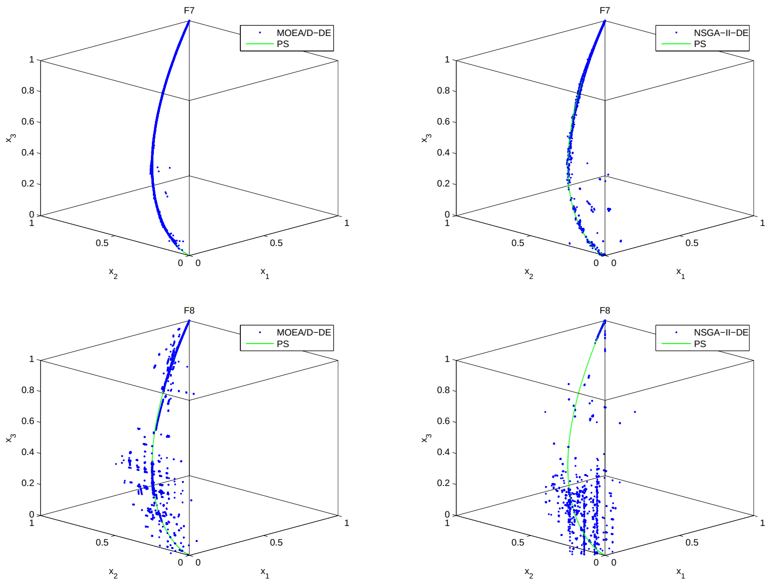


Figure 12: The plots of all the 20 final populations found by two algorithms in the decision space on F7 and F8.

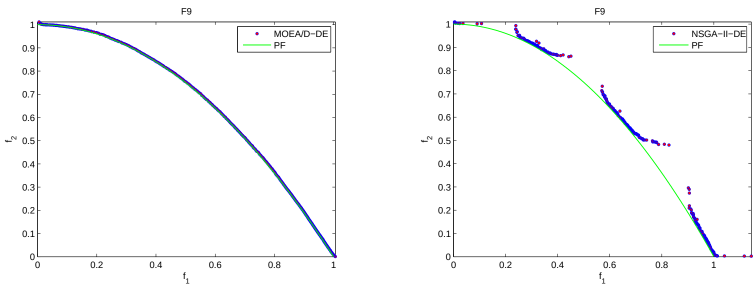


Figure 13: The plots of the final populations with the lowest IGD values found by MOEA/D-DE and NSGA-II-DE in the objective space on F9.

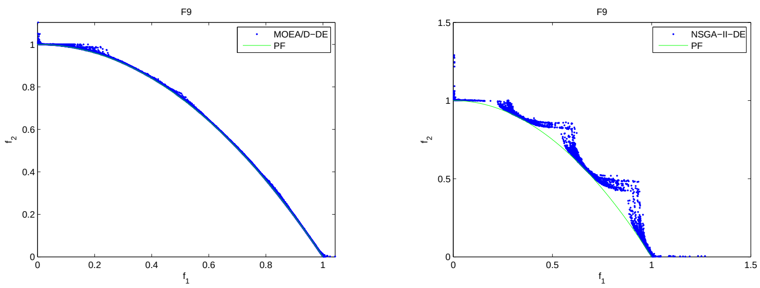


Figure 14: The plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in the objective space on F9.

4.5 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.

Figure 2 suggests that MOEA/D-DE outperforms NSGA-II-DE on this instance in terms of IGD-metric. The experimental results presented in Figures 13-16, together with the results on F2 in Section 4.3 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 method, which is not very sensitive to PF shapes. It is also evident from Figures 13-16 and Figures 3-6 that a concave PF does hinder the performance of NSGA-II-DE very much if the PS is nonlinear as in F9.

4.6 More Discussion

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 problem for each individual and then the computational ef-

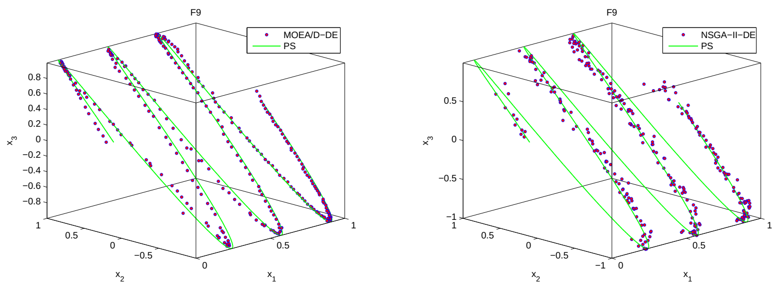


Figure 15: The plots of the final populations with the lowest IGD values found by MOEA/D-DE and NSGA-II-DE in the decision space on F9.

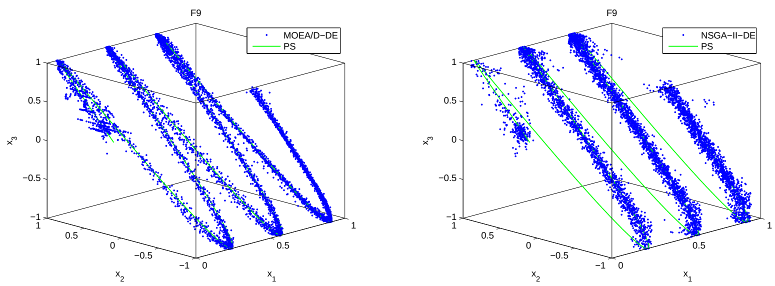


Figure 16: The plots of all the 20 final populations found by MOEA/D-DE and NSGA-II-DE in the decision space on F9.

fort 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.

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. But it does not imply that an NSGA-II with other reproduction operators is hopeless. In fact, the second author and his co-workers have very recently shown that although RM-MEDA, an EDA version of NSGA-II reported in [23], does not work well on the test instances, a new implementation of RM-MEDA performs very similarly to MOEA/D-DE on the test instances introduced in this paper [3]. 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 implementation is much more complicated than MOEA/D-DE.

5 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 an new version of MOEA/D, i.e., MOEA/D-DE and compared it 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 implies that decomposition based multiobjective evolutionary algorithms are very promising in dealing with complicated PS shapes.

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 information collected during the search. 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 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 modify an algorithm for solving their problems.

References

[1] K. Miettinen, *Nonlinear Multiobjective Optimization.* Kluwer Academic Publishers, 1999.

[3] the results will be reported in a forthcoming working report soon

[2] M. Ehrgott, *Multicriteria Optimization.* Springer, Second edition, 2005.

[3] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms.* Lawrence Erlbaum Associates, Inc., 1985, pp. 93–100.

[4] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, 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 *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence.* Piscataway, New jersey: IEEE Service Center, June 1994, pp. 82–87.

[6] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach." *IEEE Trans. Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 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 *Proceedings of Third Australia-Japan Joint Workshop on Intelligent and Evolutionary Systems*, Ashikaga, Japan, November 1999, pp. 209–216.

[9] 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, 2002, pp. 95–100.

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

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

[12] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley & Sons, Inc., 2001.

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

[14] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications (Advanced Information and Knowledge Processing).* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.

[15] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, October 2006.

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

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

[18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *CEC '02: Congress on Evolutionary Computation*, vol. 1. Piscataway, New Jersey: IEEE Service Center, 2002, pp. 825–830.

[19] 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," Special Session on Constrained Real-Parameter Optimization, Nanyang Technological University, Singapore, Tech. Rep., 2007.

[20] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On test functions for evolutionary multi-objective optimization." in *PPSN VIII : the 8th International Conference of Parallel Problem Solving from Nature.* Birmingham: Springer, Berlin, 2004, pp. 792–802.

[21] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies." in *GECCO*, 2006, pp. 1141–1148.

[22] H. Li and Q. Zhang, "A multi-objective differential evolution based on decomposition for multiobjective optimization with variable linkages." in *PPSN*, 2006, pp. 583–592.

[23] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: Modelling the regularity in an estimation of distribution algorithm for continuous multiobjective optimisation with variable linkages," *IEEE Trans. on Evolutionary Computation*, 2007, in press.

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

[25] H. Ishibuchi and T. Murata, "Multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 28, no. 3, pp. 392–403, August 1998.

[26] Y. W. Leung and Y. Wang, "Multiobjective programming using uniform design and genetic algorithm." *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, vol. 30, no. 3, pp. 293–304, 2000.

[27] Y. Jin, T. Okabe, and B. Sendhoff, "Adapting weighted aggregation for multiobjective evolutionary strategies," in *EMO '01: Evolutionary Multicriterion OPtimization*, Springer LNCS 1993, 2001, pp. 96–110.

[28] A. Jaszkiewicz, "On the performance of multiple-objective genetic local search on the 0:1 knapsack problem - a comparative experiment." *IEEE Trans. Evolutionary Computation*, vol. 6, no. 4, pp. 402–412, 2002.

[29] E. J. Hughes, "Multiple single objective pareto sampling," in *CEC '03: Congress on Evolutionary Computation*, vol. 4, no. 8-12, Canberra, 2003, pp. 2678–2684.

[30] Q. Zhang and H. Li, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation*, 2007, in press.

[31] A. W. Iorio and X. Li, "Solving rotated multi-objective optimization problems using differential evolution." in *Proceeding of the 17th Joint Australian Conference on Artificial Intelligence*, ser. Lecture Notes in Computer Science, G. Webb and X. Yu, Eds., 2004, pp. 861–872.

[32] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *CEC '05: Congress on Evolutionary Computation*. Edinburgh, UK: IEEE service Center, September 2005, pp. 239–246.

[33] V. L. Huang, A. K. Qin, P. N. Suganthan, and M. F. Tasgetiren, "Multi-objective optimization based on self-adaptive differential evolution," in *Proceedings of the 2007 Congress on Evolutionary Computation CEC2007*, 2007.

[34] 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. Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.