Approximating the Set of Pareto Optimal Solutions in Both the Decision and Objective Spaces by an Estimation of Distribution Algorithm

Aimin Zhou, Qingfu Zhang and Yaochu Jin

Technical Report CES-485
Department of Computing and Electronic Systems
University of Essex
June 25, 2008

Abstract

Most existing multiobjective evolutionary algorithms aim at approximating the PF, the distribution of the Pareto optimal solutions in the objective space. In many real-life applications, however, a good approximation to the PS, the distribution of the Pareto optimal solutions in the decision space, is also required by a decision maker. This paper considers a class of MOPs, in which the dimensionalities of the PS and PF are different so that a good approximation to the PF might not approximate the PS very well. It proposes a probabilistic model based multiobjective evolutionary algorithm, called MMEA, for approximating the PS and the PF simultaneously for a MOP in this class. In the modelling phase of MMEA, the population is clustered into a number of subpopulations based on their distribution in the objective space, the PCA technique is used to detect the dimensionality of the centroid of each subpopulation, and then a probabilistic model is built for modelling the distribution of the Pareto optimal solutions in the decision space. Such modelling procedure could promote the population diversity in both the decision and objective spaces. To ease the burden of setting the number of subpopulations, a dynamic strategy for periodically adjusting it has been adopted in MMEA. The experimental comparison between MMEA and the two other methods, KP1 and Omni-Optimizer on a set of test instances, some of which are proposed in this paper, have been made in this paper. It is clear from the experiments that MMEA has a big advantage over the two other methods in approximating both the PS and the PF of a MOP when the PS is a nonlinear manifold, although it might not be able to perform significantly better in the case when the PS is a linear manifold.

Index Terms

Multiobjective optimization, Pareto optimality, estimation of distribution algorithm, principal component analysis.

I. Introduction

This paper considers the following *continuous multiobjective optimization problem* (continuous MOP):

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \cdots, f_m(x)) \\ \text{subject to} \quad & x \in \prod_{i=1}^{n} [a_i, b_i] \end{aligned} \tag{1}$$

where $-\infty < a_i < b_i < +\infty$ for all $i = 1, \ldots, n$. $\prod_{i=1}^{n} [a_i, b_i] \subset R^n$ is the decision space and $x = (x_1, \ldots, x_n) \in R^n$ is the decision variable vector. $F : \prod_{i=1}^{n} [a_i, b_i] \rightarrow R^m$ consists of m real-valued continuous objective functions $f_i(x)$, $i = 1, \ldots, m$. R^m is the objective space.

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 $u \neq v$ and $u_i \leq v_i$ for all $i = 1, \ldots, m$. $x^\star$ is called *(globally) Pareto optimal* if there is no other x 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 image of the PS on the objective space, PF $= \{y \in R^m | y = F(x), x \in \text{PS}\}$, is called the *Pareto front* [1], [2].

Most existing multiobjective evolutionary algorithms (MOEAs) aim at finding an approximation to PFs [2]–[15]. However, in some real-world applications, particularly when the preference (i.e. utility function) of a decision maker is not clearly defined, a good approximation to both the PF and the PS should be required by the decision maker for facilitating their decision making as argued in [16]–[18]. If the mapping from the PS to the PF is one-to-one and relatively uniform, a good approximation to the PF could approximate the PS well too. Otherwise, this could not be the case. Two typical classes of continuous MOPs, in which the approximation of their PSs should be carefully addressed, are as follows:

A. Zhou and Q. Zhang are with Department of Computing and Electronic Systems, University of Essex, Wifehood Park, Colchester, CO4 3SQ, U.K.
Y. Jin is with Honda Research Institute Europe, Carl-Legien-STD. 30, 63073 Offenbach, Germany.

- Class I: A finite number of different points in the PS may have the same image in the PF under the mapping F from the PS to the PF, but the PS and the PF are of the same dimensionality, which is $m-1$ under some mild conditions [19]. Therefore, the PS could consist of a number of disconnected continuous $(m-1)$-D manifolds. ZDT6 [20], Jin1 [21] and the SYM-PART instances [22] are test instances in this class.
- Class II: The PF is a $(m-1)$-D continuous manifold and the PS is a continuous manifold of a higher dimensionality. All the inverse images of a point in the PF could constitute a non-zero dimensional continuous manifold. Some WFG test instances [23] are in this class.

To generate a good approximation to both the PS and the PF of a MOP, an algorithm should arguably have an effective mechanism to encourage and maintain the population diversity, not only in the objective space as most MOEAs do, but also in the decision space. For this reason, Deb and Tiwari introduced the crowding distance in the decision space into the nondominated sorting scheme in Omni-Optimizer [24], a generalization of NSGA-II [25], for promoting the population diversity in the decision space. Chan and Ray suggested using two selection operators in MOEAs, one encourages the diversity in the objective space and the other does so in the decision space [26]. They implemented KP1 and KP2, two algorithms using these two selection operators. It should be pointed out that the MOPs that KP1, KP2 and Omni-Optimizer attempt to deal with are of Class I. Rudolph *et al* also proposed to use a restart strategy for finding a good approximation to the PS of a MOP of Class I [22], [27]. To the best of our knowledge, no effort has been made for dealing with problems of Class II. The major purpose of this paper is to study how to approximate both the PS and the PF of a MOP of Class II.

In [13], we studied a "regular" continuous MOP in which both the PF and FS are piecewise continuous manifolds of $(m-1)$-D dimensionality, and proposed RM-MEDA, an estimation of distribution algorithm (EDA) for approximating its PF. In this paper, we generalize the idea of RM-MEDA and propose a model based multiobjective evolutionary algorithm, called **MMEA**, for approximating the PS and the PF of a MOP of Class II simultaneously. MMEA has the following features:

- The population diversity in the decision space is promoted in its reproduction generator, instead of in the selection operators as in Omni-Optimizer, KP1 and KP2. The NDS-selection, which is used in RM-MEDA, is employed in MMEA.
- To build a probabilistic model of promising solutions, the population is divided, based on their distribution in the objective space, into a number of subpopulations. Therefore, the population diversity in the objective space can be promoted. To ease the burden of tuning the number of subpopulations, a dynamical strategy for tuning it has been adopted in MMEA.
- The principal component analysis technique is used to detect the dimensionality of the centroid of each subpopulation in the decision space, and then a probabilistic model can be built for modelling its distribution in the decision space. In such a way, the population diversity in the decision space can be encouraged.

The rest of the paper is organized as follows: Section II gives the details of the algorithm. Section III presents the performance metrics and the test instances, some of which are firstly proposed in this paper. Section IV compares MMEA with KP1 and Omni-Optimizer on these test instances. More discussions on the ability of MMEA are provided in Section V. Section VI concludes this paper and lists some future research topics.

II. Algorithm

A. Framework

At each generation, the proposed algorithm, MMEA, maintains:

- A population of N solutions (i.e. points in $\prod_{i=1}^{n}[a_i, b_i]$):

$$x^1, \ldots, x^N.$$

- Their F-values: $F(x^1), \ldots, F(x^N)$.

MMEA adopt the following widely used estimation of distribution algorithm framework:

Phase 1 Initialization: Generate an initial population P and compute the F-values of these solutions in P.

Phase 2 Modelling: Build a model for modelling the distribution of the individuals in P.

Phase 3 Reproduction: Generate a set of new solutions Q by sampling from the model built in Phase 2 and compute the F-values of these solutions in Q.

Phase 4 Selection: Select N solutions from $P \bigcup Q$ and replace all the solutions in P by them.

Phase 5 Stopping Condition: If a stoping condition is met, stop and return all the solutions in P and their corresponding F-values. Otherwise, go to **Phase 2**.

In the following, we give and discuss the implementations of modelling, reproduction and selection.

B. Modelling

In a successful algorithm for approximating both the PS and the PF of (1), the individuals in its population should approximate the PS in the decision space and their images should converge to the PF in the objective space as the search goes on. Therefore, one could model the PS and the PF based on information extracted from the population. Such models can be further used for

sampling new good solutions. Such an idea has been used to some extent in RM-MEDA. The problem that RM-MEDA was designed for is a 'regular" continuous MOP, in which both the PS and the PF are of the same dimensionality. In this paper, the same idea is used in the modelling phase of MMEA for dealing with a MOP of Class II.

The modelling phase in MMEA works as follows:

Step 1 Building a Utopian PF: Based on information from the current population P, build a $(m-1)$-D simplex in the objective space as a Utopian PF.

Step 2 Determining the Number of Subpopulations: Determine K, the number of subpopulations used in modelling the PS.

Step 3 Selecting Reference Points: Set $Y^1, \ldots, Y^K$, K points which are uniformly spread on UPF in the objective space, to be K reference points.

Step 4 Clustering: For each reference point Y^i, compute, in the objective space, its Euclidian distances to all the individual solutions in P. Select the $[\frac{2N}{K}]$ closest solutions to Y^i and let them constitute subpopulation P^i.

Step 5 Principal Component Analysis and Modelling: Do Principal Component Analysis (PCA) on each subpopulation P^i and build a model for it.

In the following, we give the details of the major steps in the above modelling phase.

1) Building a Utopian PF: We assume that the PF of the MOP in question is of $(m-1)$-D. Therefore, it is reasonable to use a $(m-1)$-D simplex as a Utopian PF. The following procedure is used to construct such a simplex S:

Step 1.1 For $i = 1, \ldots, m$, find the individual solution z^i in P such that z^i is a nondominated solution in P and it has the largest f_i function value among all the nondominated solutions in P.

Step 1.2 Initialize S as the $(m-1)$-D simplex with vertexes $F(z^1), \ldots, F(z^m)$ in the objective space. Move S along its normal direction to a position such that (a) no point in S can be dominated by any solutions in P, and (b) the moved distance should be as short as possible.

Step 1.3 Let $A_1, \ldots, A_m$ be the vertexes of the moved simplex S. Compute the center of S:

$$O = \frac{1}{m}\sum_{i=1}^{m} A_i.$$

Then enlarge S by moving its vertexes:

$$A_i := A_i + 0.25(A_i - O)$$

for $i = 1, \ldots, m$.

The major reason why we enlarge S is to guide the algorithm to extend its search in the objective space. When m, the number of the objectives is 2, S is a one-D line segment in the objective space. Fig. 1 illustrates how S is generated in such a case.

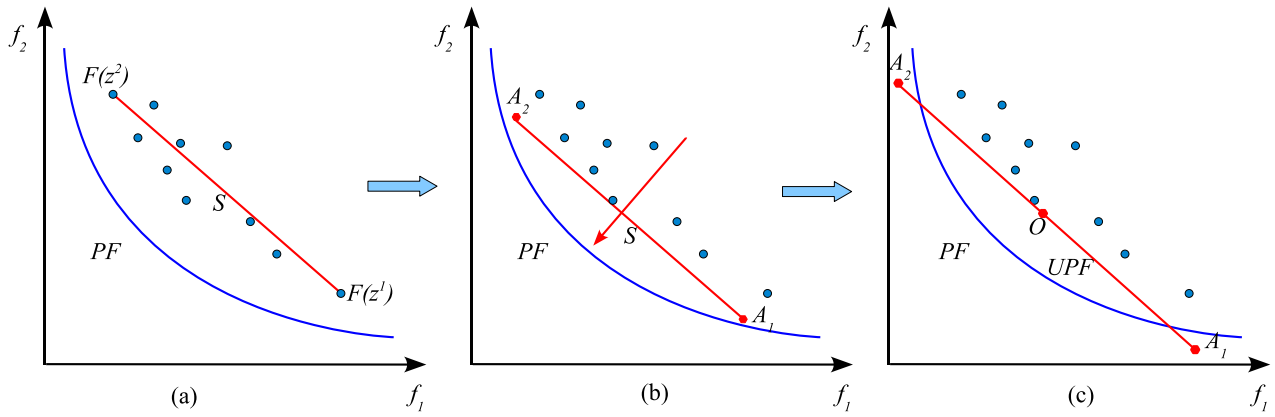


Fig. 1. Illustration of building a Utopian PF in the case of two objectives. (a) Find the extreme points and initialize simplex S. (b) Move S along its normal direction. (c) Enlarge S.

2) Determining the Number of Subpopulations: To reduce the problem-dependence of K, the value of K is changing dynamically and periodically during the search. More particularly,

$$K = K_{min} + (\lfloor \frac{t}{\Delta t} \rfloor) mod (K_{max} - K_{min})$$

where

- t is the current generation number,
- K_{min} and K_{max} are predetermined minimal and maximal values for K, respectively,
- Δt is a predetermined length of the time interval during which the value of K is fixed.

3) Selecting Reference Points and Clustering: Reference points uniformly spread on the Utopian PF and different subproblems are associated with different reference points. Therefore, the search effort could be distributed to different reference points of the Utopian PF in a relatively uniform way. In clustering, different subpopulations may overlap, which aims at improving the search performance in between different reference points.

4) PCA and Modelling: The individual solutions in subpopulation P^i should, hopefully, scatter around part of the PS in the decision space as the search goes on. For simplicity, we can model the centroid of P^i as a hyper-rectangle Φ^i in the decision space and regard each individual in P^i as an observation of the following random rector:

$$\xi = \zeta + \varepsilon,$$

where ζ is uniformly randomly distributed on Φ^i, $\varepsilon \sim N(0, \sigma_i^2 I)$ is a n-dimensional zero-mean Gaussian vector, I is the $n \times n$ identity matrix and $\sigma_i > 0$.

Now the task is how to estimate Φ^i and σ_i. We do it as follows:

Step 5.1 Compute the sample mean and the sample covariance matrix of the individual solutions in P^i:

$$\bar{x}^i = \frac{1}{|P^i|} \sum_{x \in P^i} x,$$

and

$$Cov = \frac{1}{|P^i| - 1} \sum_{x \in P^i} (x - \bar{x})(x - \bar{x})^T.$$

where $|P^i|$ is the cardinality of P^i.

Step 5.2 Compute the eigenvalues of Cov^i:

$$\lambda_1^i \geq \lambda_2^i \geq \ldots \geq \lambda_n^i$$

and their corresponding unity eigenvectors:

$$V_1^i, V_2^i \ldots, V_n^i.$$

Step 5.3 Set n^i, the dimensionality of Φ^i to be the smallest integer such that:

$$\sum_{j=1}^{n^i} \lambda_j^i \geq 0.8 \sum_{j=1}^{n} \lambda_j^i$$

Step 5.4 Compute the range of the projections of the points in P^i onto the first n^i principal component directions:

$$l_j^i = \min_{x \in P^i} (x - \bar{x}^i)^T V_j^i$$

and

$$u_j^i = \max_{x \in P^i} (x - \bar{x}^i)^T V_j^i$$

for $j = 1, \ldots, n^i$.

Step 5.5 Set

$$\begin{gathered} \Phi^i = \{x \in R^n | x = \bar{x}^i + \textstyle\sum_{j=1}^{n^i} \alpha_j V_j^i, \\ l_j^i - 0.25(l_j^i - u_j^i) \leq \alpha_j \leq u_j^i + 0.25(u_j^i - l_j^i), \\ j = 1, \ldots, n^i.\}. \end{gathered}$$

Step 5.6 Set

$$\sigma_i = \frac{1}{n - n^i} \sum_{j=n^i+1}^{n} \lambda_j^i.$$

The dimensionality of the PS is unknown, so is that of Φ^i, the centroid of P^i. In Step 5.3, the dimensionality of Φ^i is set such that Φ^i holds at least 80% of the variation in the solutions in P^i. Φ^i extends by 50% along each of the first n^i principal component directions, the smallest n^i-D hyper-rectangle containing the projections of all the solutions of P^i on the space spanned by $V_1^i, \ldots, V_{n^i}^i$ from $\bar{x}^i$. The motivation behind this extension is to extrapolate the points in P^i for searching unexploited promising areas in the decision space. ε is modelled as a Gaussian noise vector and all its components are i.i.d., which facilitates the sampling procedure.

The three major differences in the modelling phase between RM-MEDA and MMEA are:

- In clustering, RM-MEDA uses the local PCA [28] technique to partition the population into several clusters. In contrast, MMEA in this paper selects the subpopulation centers from the Utopian PF and does clustering based on the distances

in the objective space, which is computationally cheaper. Moreover, the local PCA could not be applied in MMEA since the dimensionality of each subpopulation centroid must be predetermined in the local PCA and such dimensionality is unknown in the problems MMEA aims to solve.
- The number of clusters is preset in RM-MEDA. While MMEA in this paper changes the number of subpopulations dynamically, which lightens the burden of tuning this algorithmic parameter.
- In modelling each subpopulation, RM-MEDA sets the dimensionality of its centroid to be $(m-1)$, while MMEA needs to estimate it. This difference is due to the fact that these two algorithms are for different MOPs.

C. Sampling

A new solution x is generated in Phase 4 of MMEA as follows:

Step 1 Uniformly randomly generate an integer k from $\{1, 2, \ldots, K\}$.

Step 2 Uniformly randomly generate a point x' from Φ^k. Generate a noise vector ε' from $N(0, \sigma_k I)$.

Step 3 Return $x = x' + \varepsilon'$.

Step 4 If x_j, an element of x is not in $[a_j, b_j]$, randomly select a solution y from subpopulation k (obtained in Step 4 in modelling phase), and rest x^j:

$$x^j = \frac{1}{2}(y^j + \theta)$$

where

$$\theta = \begin{cases} a^j, & \text{if } x^j < a^j, \\ b^j & \text{if } x^j > b^j, \end{cases}$$

In our implementation, the above procedure is repeated N times for generating N solutions in Phase 4.

D. Selection

The selection operator used in the experimental studies is the NDS-selection, a version of non-dominated sorting scheme [25] proposed in [13]. It works as follows:

Step 1 Set $Q = P \cup Q$ and $P = \emptyset$.

Step 2 Partition Q into different fronts $F_1, \ldots, F_l$ by using the fast non-dominated sorting approach [25]. Set $k = 0$.

Do

$k = k + 1$,
$P = P \cup F_k$,

Until $|P| \geq N$.

Step 3 While $|P| > N$, Do

For all the individual members in $F_k \cap P$, compute their crowding distances in $F_k \cap P$. Remove the element in $F_k \cap P$ with the smallest crowding distance from P. In the case when there are more than one members with the smallest crowding distance, randomly choose one and remove it.

In Step 2, the NDS-Selection partitions Q into different fronts $F_1, \ldots, F_l$ such that the j-th front F_j contains all the non-dominated solutions in $\{P \cup Q\} \backslash (\cup_{i=1}^{j-1} F_i)$. Therefore, there is no solution in $\{P \cup Q\} \backslash (\cup_{i=1}^{j-1} F_i)$ that could dominate a solution in F_j.

The crowding distance, used in Step 3, of point x in S is defined as the average side length of the largest m-D rectangle in the objective space subject to the two constraints: (a) each of its sides is parallel to a coordinate axis, and (b) $F(x)$ is the only point in $F(S) = \{F(y) | y \in S\}$ that is an interior point in the rectangle. A solution with a larger crowding distance is given priority to be selected since it could increase the population diversity in the objective space.

III. Test Instances and Performance Metrics

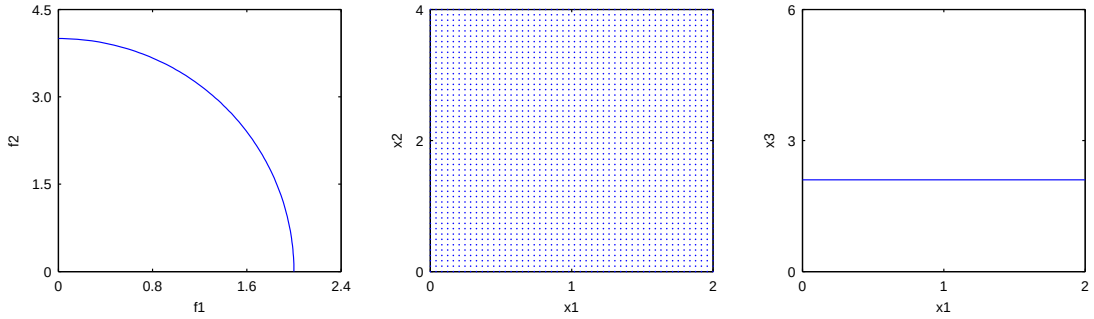
A. Test Instances

MMEA is for approximating both the PS and the PF of a MOP of Class II. Only the WFG instances, among the continuous MOP test instances we have found in the literature, are in Class II. Two WFG instances: WFG6 and WFG7, have been used in our experiments. Based on the experiments in [23], the PFs of WFG7 could be "easily" and "quickly" found by NSGA-II, while WFG6 is "hard" for NSGA-II. It could be because that the objectives in WFG7 are separable while it is not the case in WFG6. The PSs of these two test instances are 2-D rectangles in the decision space when their control parameters are set as in Table I. To study the behaviors of algorithms on nonlinear PSs, we have designed several MOP test instances of Class II with nonlinear PSs. All these test instances are listed in Table I. Figs. 2 and 3 plot their PFs and the projections of their PSs onto lower-D spaces.

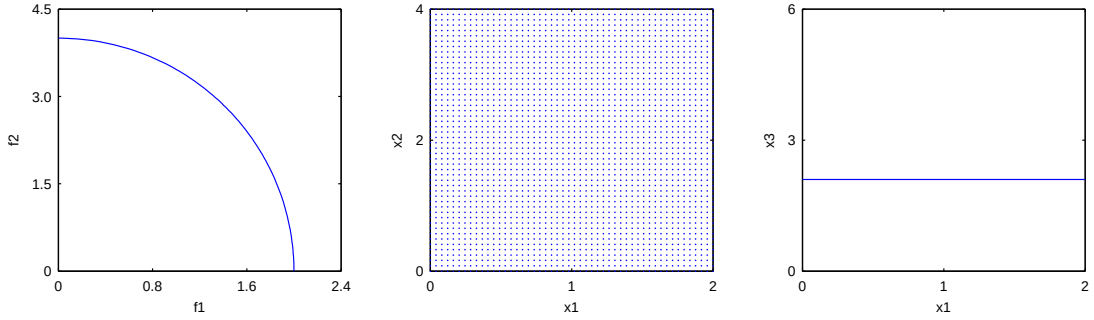
TABLE I

TEST INSTANCES USED IN OUR EXPERIMENTS: F1-2 ARE WFG INSTANCES, F3-F7 ARE NEW DESIGNED TEST INSTANCES. $x = (x_1, \ldots, x_n)$.

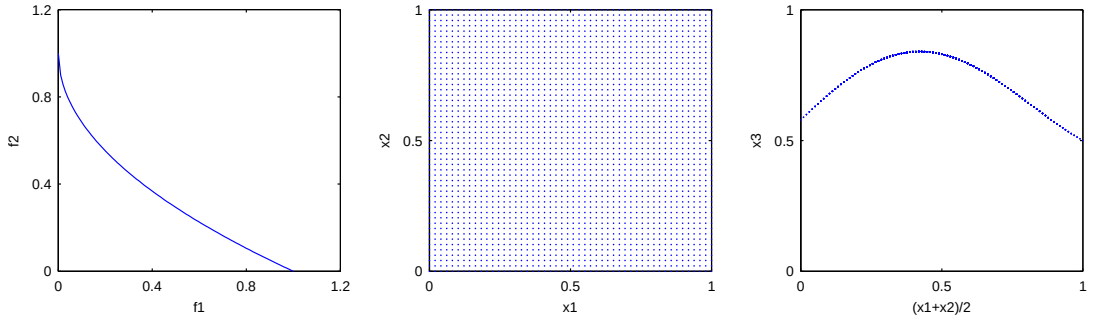
Instance	Range of x_i	Objectives, PS and PF	Remarks
$F1$	$[0, 2i]$	$WFG6$ ($M = 2, k = 2$ [23] PS: $x_i = 0.7i$, for $i = 3, \ldots, n$, $0 \le x_1 \le 2$, $0 \le x_2 \le 4$. PF: $f_1 = 2\sin(t)$, $f_2 = 4\cos(t)$, $0 \le t \le 0.5\pi$.	PS is a 2-D rectangle. PS is concave. two objectives.
$F2$	$[0, 2i]$	$WFG7$ ($M = 2, k = 2$) [23] PS: $x_i = 0.7i$, for $i = 3, \ldots, n$, $0 \le x_1 \le 2$, $0 \le x_2 \le 4$. PF: $f_1 = 2\sin(t)$, $f_2 = 4\cos(t)$, $0 \le t \le 0.5\pi$.	PS is a 2-D rectangle. PF is concave. two objectives.
$F3$	$[0, 1]$	$f_1(x) = (x_1 + x_2)/2$, $f_2(x) = g(x)(1 - \sqrt{\frac{f_1}{g}})$, where $g(x) = 1 + \frac{5}{n-2}\sum_{i=3}^{n} h(x_i)^2$ and $h(x_i) = \begin{cases} 2x_i - \sin(0.5f_1\pi)\cos(2\pi f_1 + i\pi/n) - 1, & i \text{ is even}, \\ 2x_i - \cos(0.5f_1\pi)\sin(\frac{1}{3}(2\pi f_1 + i\pi/n)) - 1, & i \text{ is odd}. \end{cases}$ PS: $x_i = \begin{cases} 0.5 + 0.5\sin(0.5f_1\pi)\cos(2\pi f_1 + i\pi/n), & i \text{ is even}, \\ 0.5 + 0.5\cos(0.5f_1\pi)\sin(\frac{1}{3}(2\pi f_1 + i\pi/n)), & i \text{ is odd}. \end{cases}$ for $i = 3, \ldots, n$,and $0 \le x_1, x_2 \le 1$. PF: $f_2 = 1 - \sqrt{f_1}$, $0 \le f_1 \le 1$.	PS is a 2-D nonlinear surface. PF is convex. two objectives.
$F4$	$[0, 1]$	$f_1(x) = (x_1 + x_2)/2$, $f_2(x) = g(x) - f_1^2$, where $g(x) = 1 + \frac{5}{n-2}\sum_{i=3}^{n} h(x_i)^2$, and $h(x_i) = \begin{cases} 2x_i - f_1\cos(2\pi f_1 + i\pi/n) - 1, & i \text{ is even}, \\ 2x_i - f_1\sin(2\pi f_1 + i\pi/n) - 1, & i \text{ is odd}. \end{cases}$ PS: $x_i = \begin{cases} 0.5 + 0.5f_1\cos(2\pi f_1 + i\pi/n), & i \text{ is even}, \\ 0.5 + 0.5f_1\sin(2\pi x_1 + i\pi/n), & i \text{ is odd}, \end{cases}$ for $i = 3, \ldots, n$, and $0 \le x_1, x_2 \le 1$. PF: $f_2 = 1 - f_1^2$, $0 \le f_1 \le 1$.	PS is a 2-D nonlinear surface. PF is concave. two objectives.
$F5$	$[0, 1]$	$f_1(x) = (x_1 + x_2)/2$, $f_2(x) = g(x) - f_1 + \sin(2\pi f_1)/(2\pi)$, where $g(x) = 1 + \frac{5}{n-2}\sum_{i=3}^{n} h(x_i)^2$, and $h(x_i) = \begin{cases} 2x_i - f_1\cos(2\pi f_1 + i\pi/n) - 1, & i \text{ is even}, \\ 2x_i - f_1\sin(\frac{1}{3}(2\pi f_1 + i\pi/n)) - 1, & i \text{ is odd}. \end{cases}$ PS: $x_i = \begin{cases} 0.5 + 0.5f_1\cos(2\pi f_1 + i\pi/n), & i \text{ is even}, \\ 0.5 + 0.5f_1\sin(\frac{1}{3}(2\pi x_1 + i\pi/n)), & i \text{ is odd}, \end{cases}$ for $i = 3, \ldots, n$, and $0 \le x_1, x_2 \le 1$. PF: $f_2 = 1 - f_1 + \sin(2\pi f_1)/(2\pi)$, $0 \le f_1 \le 1$.	PS is a 2-D nonlinear surface. PF is not concave or convex. two objectives.
$F6$	$[0, 1]$	$f_1(x) = (x_1 + x_2 + x_3)/3$, $f_2(x) = g(x) - f_1^2$, where $g(x) = 1 + \frac{5}{n-3}\sum_{i=4}^{n} h(x_i)^2$, and $h(x_i) = \begin{cases} 2x_i - \sin(0.5f_1\pi)\cos(2\pi f_1 + i\pi/n) - 1, & \text{i is even}, \\ 2x_i - \cos(0.5f_1\pi)\sin(\frac{1}{3}(2\pi f_1 + i\pi/n)) - 1 & \text{i is odd}. \end{cases}$ PS: $x_i = \begin{cases} 0.5 + 0.5sin(0.5f_1\pi)\cos(2\pi f_1 + i\pi/n), & \text{i is even}, \\ 0.5 + 0.5\cos(0.5f_1\pi)\sin(\frac{1}{3}(2\pi f_1 + i\pi/n)), & \text{i is odd}, \end{cases}$ for $i = 4, \ldots, n$, and $0 \le x_1, x_2, x_3 \le 1$. PF: $f_2 = 1 - f_1^2$, $0 \le f_1 \le 1$.	PS is 3-D continuous nonlinear manifold. PF is concave. two objectives.
$F7$	$[0, 1]$	$f_1(x) = g(x)\cos(0.25\pi(x_1 + x_2))\sin(0.5\pi x_3)$, $f_2(x) = g(x)\cos(0.25\pi(x_1 + x_2))\cos(0.5\pi x_3)$, $f_3(x) = g(x)\sin(0.25\pi(x_1 + x_2))$, where $g(x) = 1 + \frac{5}{n-3}\sum_{i=4}^{n} h(x_i)^2$, $h(x_i) = \begin{cases} 2x_i - \sin(0.5\pi y)\cos(2\pi y + i\pi/n) - 1, & i \text{ is even}, \\ 2x_i - \cos(0.5\pi y)\sin(\frac{1}{3}(2\pi y + i\pi/n)) - 1, & i \text{ is odd}, \end{cases}$ and $y = (x_1 + x_2 + x_3)/3$. PS: $x_i = \begin{cases} 0.5 + 0.5\sin(0.5f_1\pi)\cos(2\pi f_1 + i\pi/n), & \text{i is even}, \\ 0.5 + 0.5\cos(0.5f_1\pi)\sin(\frac{1}{3}(2\pi f_1 + i\pi/n)), & \text{i is odd}, \end{cases}$ for $i = 4, \ldots, n$,and $0 \le x_1, x_2, x_3 \le 1$. PF: $f_1 = \cos(s)\sin(t)$, $f_2 = \cos(s)\cos(t)$, $f_3 = \sin(s)$, $0 \le s, t \le \pi/2$.	PS is 3-D continuous nonlinear manifold. PF is concave. three objectives.



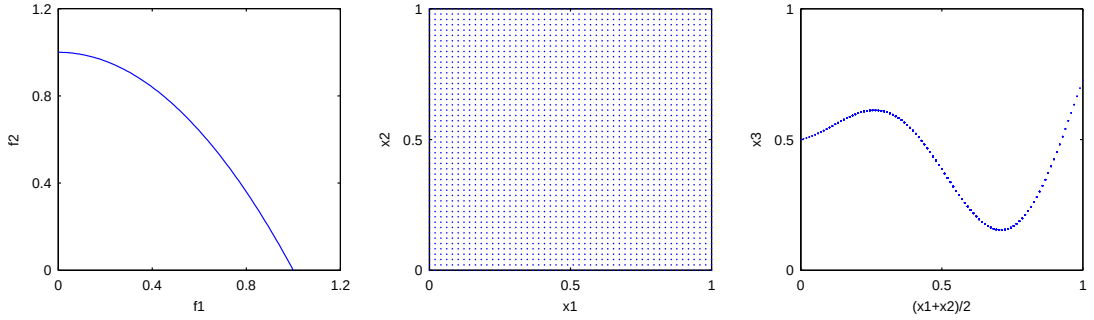
(a) F1



(b) F2

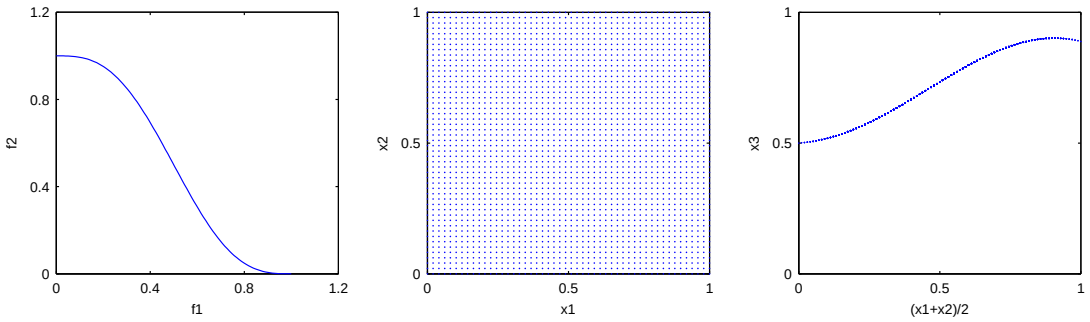


(c) F3

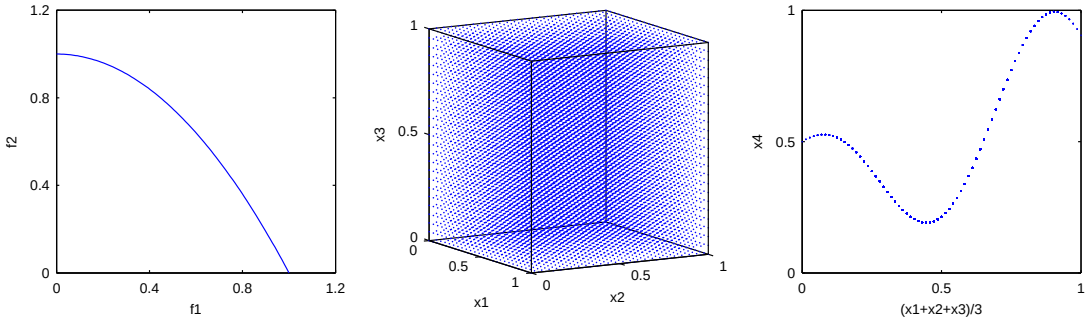


(d) F4

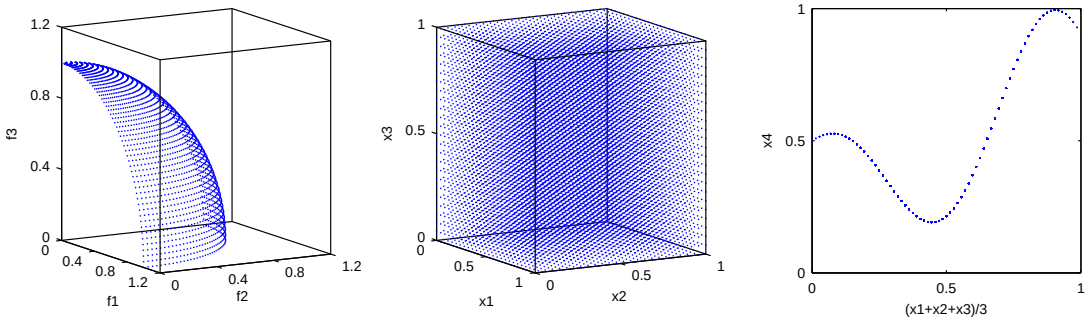
Fig. 2. The PFs and the PSs for F1-F4. (a): F1. (b): F2. (c): F3. (d): F4. Left: the PFs in the objective space. Middle: the projections of the PSs onto the x_1-x_2 space. Right: the projections of the PSs onto the x_1-x_3 space for F1 and F2, and onto the $\frac{x_1+x_2}{2}$-x_3 space for F3 and F4.



(a) F5



(b) F6



(c) F7

Fig. 3. The PFs and the PSs for F5-F7. (a): F5. (b): F6. (c): F7. Left: PFs in the objective space. Middle: the projections of the PSs onto the x_1-x_2 space for F5, and onto the x_1-x_2-x_3 space for F6 and F7. Right: the projections of the PSs onto the $\frac{x_1+x_2}{2} - x_3$ space for F5, and onto the $\frac{x_1+x_2+x_3}{3}$-x_4 space for F6 and F7.

B. Performance Metrics

The Inverted Generational Distance (IGD) metric [13], [29] is used to assess the algorithm performances in our experimental studies. Let P^* be a set of uniformly distributed Pareto optimal points in the PF (or PS). Let P be an approximation of P^*. The metric is defined as follows.

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

where $d(v, P)$ is a distance between v and P and $|P^*|$ is the cardinality of P^*.

The IGD metric can measure both diversity and convergence. To have a low IGD value, P must be very close to the PF (or PS) and cannot miss any part of the whole PF (or PS).

We denote IGD metric as $IGDF$ when P^* is a set of points in the PF and $d(v, P)$ is the Euclidian distance measured in the objective space, and as $IGDX$ when P^* is a set of points in the PS and $d(v, P)$ is the Euclidian distance measured in the decision space.

In our experiments, 1,000 points, in which f_1 or t taking 1,000 equidistant values from their lower bounds to their upper bounds, are selected from the respective PFs of F1-F6 to be $P^\star$ for computing the $IGDF$ metrics. $50 \times 50 = 2,500$ points in the PF of F7 with $s, t = \frac{0}{49} \times \frac{\pi}{2}, \frac{1}{49} \times \frac{\pi}{2}, \ldots, \frac{49}{49} \times \frac{\pi}{2}$, are taken to form $P^\star$ for computing the $IGDF$ metric for experiments on F7. $50 \times 50 = 2,500$ points in the respective PSs of F1-F5, in which x_1 and x_2 take 50 equidistant values from their lower bounds to their upper bounds respectively, are taken to form $P^\star$ for computing the $IGDX$ metrics. $25 \times 25 \times 25 = 15,625$ points in the respective PSs of F6 and F7, in which x_1, x_2 and x_3 take 25 equidistant values from their lower bounds to their

upper bounds respectively, are taken to be $P^\star$ for computing the $IGDX$ metrics.

IV. Experimental Results

A. Experimental Settings and Algorithms in Comparison

The studies in [22], [26], [27] have shown that popular MOEAs, such as PAES [30], NSGA-II [25] and SPEA2 [31], cannot approximate both the PF and the PS simultaneously since these methods could not maintain the population diversity in decision space. MOEA/D, a recent MOEA based on aggregation proposed in [32], can not do so either for the same reason. In our experiments, we have compared MMEA with KP1 [26][1] and Omni-Optimizer [24][2]. As mentioned in Introduction, both of them try to approximate both the PF and the PS of a MOP by promoting the population diversity in the decision space in their selection operators. The simulated binary crossover (SBX) [33] and the polynomial mutation [34] are used in these two methods for generating offspring.

Table II lists all the parameter settings in our experiments.

TABLE II
Experimental Settings

	The number of variables	$n = 20$
	population size for all the algorithms	$N = 250$ (F1-F5) $N = 500$ (F6-F7)
MMEA	maximal number of subpopulation	$K_{max} = 30$
	minimal number of subpopulation	$K_{min} = 5$
	the length of the time interval	$\Delta t = 10$
Omni KP1	crossover parameter in SBX	$\eta_c = 20$
	crossover rate	$P_c = 0.8$
	parameter in polynomial mutation	$\eta_m = 20$
	mutation rate	$P_m = 1/n$

All the algorithms stop after 500 generations. The population in each algorithm is initialized uniformly and randomly in the decision space. The following results are based on 20 independent runs of each algorithm on each test instance.

TABLE III
Statistical Results of $IDGF$ and $IGDX$ Metrics of the Three Algorithms on F1-F7 ($mean \pm std.$)

Instance		Omni	KP1	MMEA
F1	$IGDF$	0.0682 ± 0.0128	0.0574 ± 0.0076	**0.0106** ± 0.0022
	$IGDX$	26.8558 ± 4.2276	21.4472 ± 5.2046	**0.5480** ± 0.1751
F2	$IGDF$	0.0385 ± 0.0204	0.0107 ± 0.0007	**0.0080** ± 0.0006
	$IGDX$	0.3126 ± 0.0431	**0.2749** ± 0.0414	0.3076 ± 0.0442
F3	$IGDF$	0.5515 ± 0.0197	0.0918 ± 0.0194	**0.0177** ± 0.0090
	$IGDX$	1.0390 ± 0.0139	0.4456 ± 0.0638	**0.1280** ± 0.0226
F4	$IGDF$	0.2100 ± 0.1242	0.1316 ± 0.1107	**0.0313** ± 0.0479
	$IGDX$	0.6692 ± 0.1951	0.4729 ± 0.2220	**0.1398** ± 0.1107
F5	$IGDF$	0.1914 ± 0.0532	0.0294 ± 0.0038	**0.0139** ± 0.0033
	$IGDX$	0.5234 ± 0.0850	0.2473 ± 0.0385	**0.1008** ± 0.0072
F6	$IGDF$	0.1376 ± 0.0657	0.1037 ± 0.0177	**0.0294** ± 0.0236
	$IGDX$	0.5744 ± 0.1342	0.4680 ± 0.0668	**0.1584** ± 0.0209
F7	$IGDF$	0.6045 ± 0.0600	0.7451 ± 0.0000	**0.0591** ± 0.0024
	$IGDX$	1.0656 ± 0.0289	1.3131 ± 0.0291	**0.2400** ± 0.0116

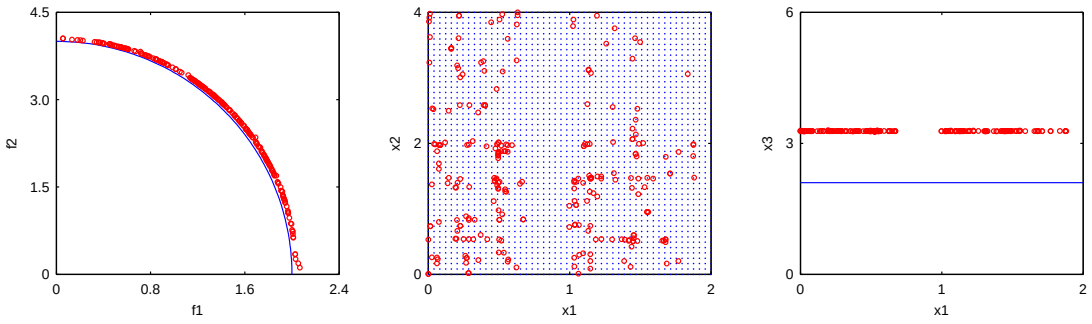
B. F1-F2

F1 and F2 have the same PS, which is a 2-D rectangle parallel to x_1-x_2 space. The objectives are nonseparable in F1 but separable in F2 [23]. The means and standard deviations can be found in Table III of the $IGDF$ and $IGDX$ values of the 20 final populations obtained by each algorithm for F1 and F2.

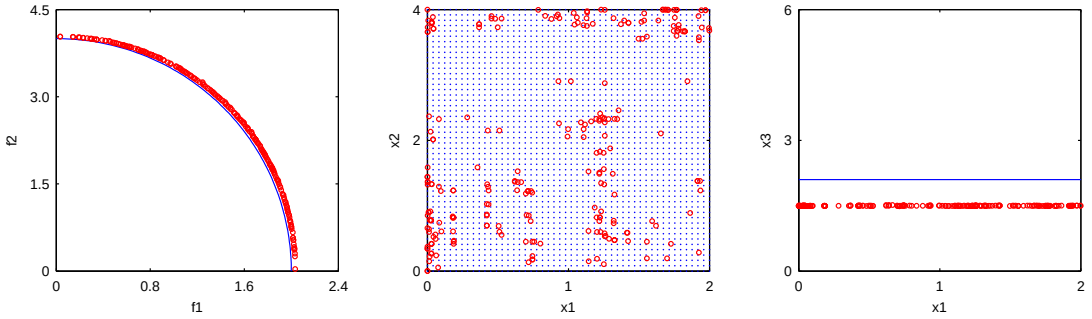
Figs. 4 and 5 show, in the objective and decision spaces, the distribution of the final solutions obtained in the runs with the lowest $IGDF$ and $IGDX$ values of each algorithm for these two test instances, respectively.

[1]We use KP1 in this paper because the experimental results in [26] have shown that KP1 is slightly better than KP2.

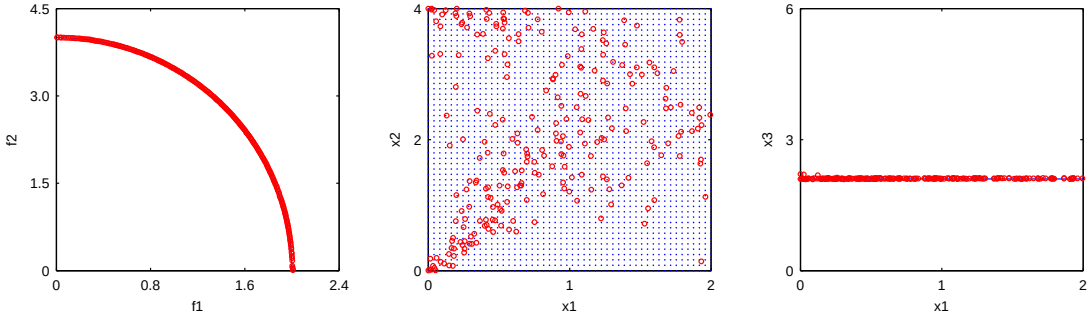
[2]The C++ source codes of KP1 was obtained from its authors. Omni-Optimizer was implemented by ourselves.



(a) Omni-Optimizer



(b) KP1



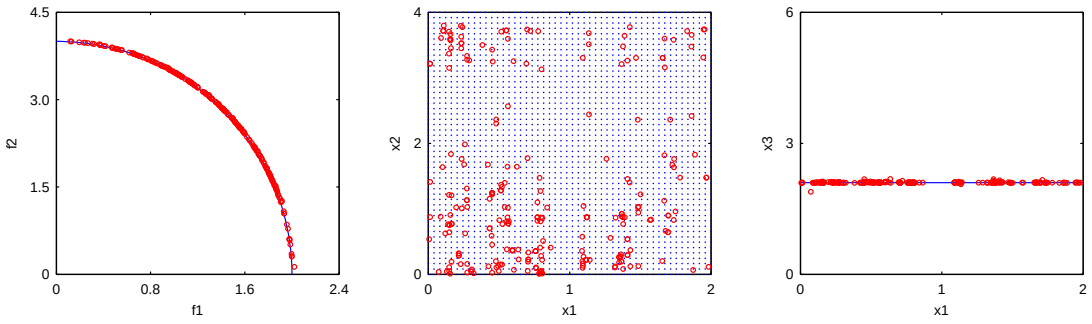
(c) MMEA

Fig. 4. The best approximations obtained by three algorithms for F1. (a): Omni-Optimizer. (b): KP1. (c): MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions of the final solutions in the x_1-x_2 space obtained in the runs with the lowest $IGDX$ values. Right: the distributions of the final solutions in the x_1-x_3 space obtained in the runs with the lowest $IGDX$ values.

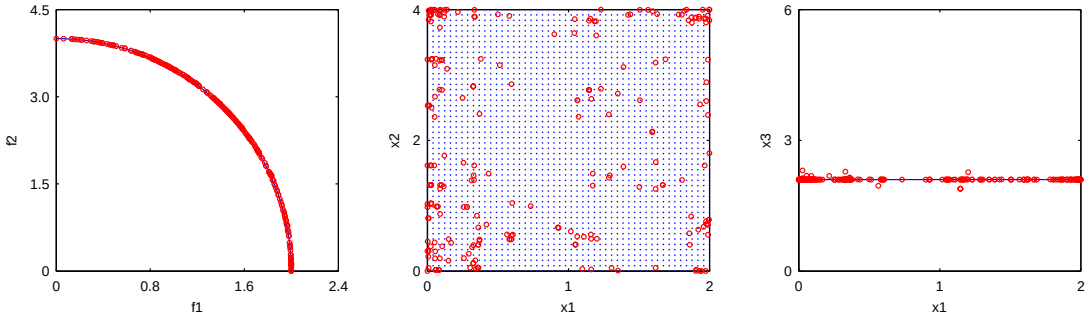
It is clear from Table III that in terms of the $IGDF$ metrics, MMEA is better than the two other algorithms on these two test instances. Figs. 4 and 5 confirm it to some extent: the final populations with lowest $IGDF$ values obtained by MMEA approximate the PFs better than that obtained the two other algorithms. Table III indicates that in terms of the $IGDX$ metrics, MMEA outperforms the two other competitors on F1 but is slightly worse than KP1 on F2. Actually, one could visually distinguish from Fig. 4 the differences in approximation quality in the x_1-x_2 and x_1-x_3 spaces between MMEA and two other methods on F1. It is hard, however, to tell any big difference in approximation quality in the decision space between MMEA and KP1 on F2 from Fig. 5. These results indicate that MMEA could not always have the edge over KP1 on MOPs with linear PSs like F2. The major reason might be that in MMEA, using a mixture of several different linear models could lead to overfitting on these linear PSs and thus deteriorate its performance.

C. F3-F7

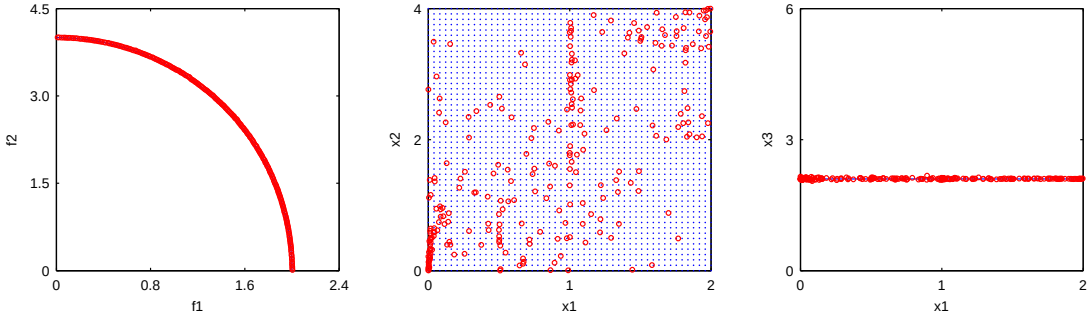
All these test instances have nonlinear PSs in the decision space. The dimensionality of the PSs of F3-F5 are 2 while that of F6 and F7 are 3. Table III suggests that in terms of both the $IGDF$ and $IGDX$ metrics, MMEA performs better than the two other algorithms. Taking F3 as an example, the $IGDF$ mean value in MMEA is just about 3% and 19% of that in Omni-Optimizer and KP1, respectively, and the $IGDX$ mean value is about 12% and 29% of that in Omni-Optimizer and KP1, respectively. It is evident from Figs.6-10 that for F3-F7, the best approximations obtained by MMEA can approximate the whole PF in the objective space very well, while KP1 and Omni-Optimizer always miss part of the PFs. Figs. 6- 10 also



(a) Omni-Optimizer



(b) KP1



(c) MMEA

Fig. 5. The best approximations obtained by three algorithms for F2. (a): Omni-Optimizer. (b): KP1. (c): MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions of the final solutions in the x_1-x_2 space obtained in the runs with the lowest $IGDX$ values. Right: the distributions of the final solutions in the x_1-x_3 space obtained in the runs with the lowest $IGDX$ values.

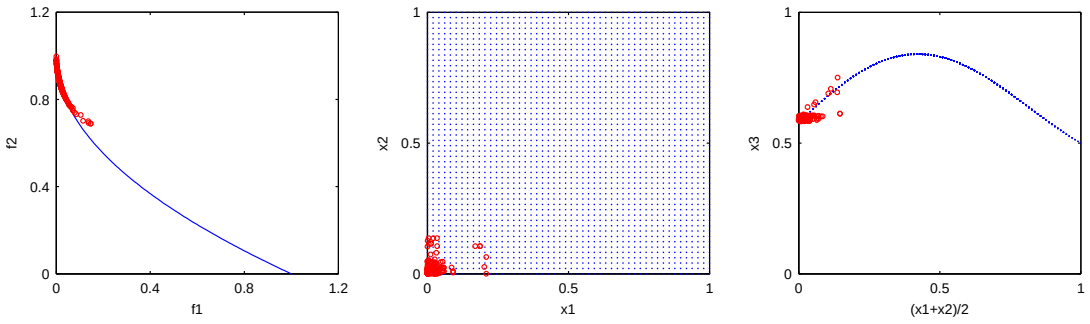
reveal that in the decision space, the best approximations obtained by MMEA is significant better than that by the two other algorithms for F3-F7.

KP1 and Omni-Optimizer promote the population diversity in their selection operators and mainly the SBX, which wad originally proposed for single objective optimization, to generate new solutions. By contrast, MMEA uses the NDS-selection operator, which does not explicitly encourage the population diversity in the decision space, and it estimates the geometrical shape of the PS and attempts to make the new solutions uniformly distribute around the estimated PS. Our experiments have suggested that reproduction operators are of crucial importance in MOEAs for approximating both the PS and the PF and one should use problem-specific knowledge in designing reproduction operators in MOEAs. The major reason that KP1 and Omni-Optimizer fail in F3-F7 might be that the SBX is not suitable for a MOP with nonlinear PSs. Actually, if two parent solutions are Pareto optimal (i.e. in the PS), it is very likely that their offspring under the SBX are far away from the PS.

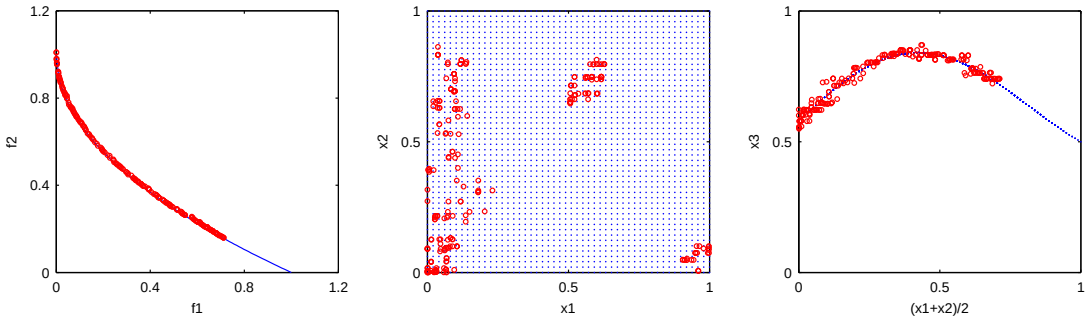
V. More discussions

A. Could MMEA solve regular MOPs?

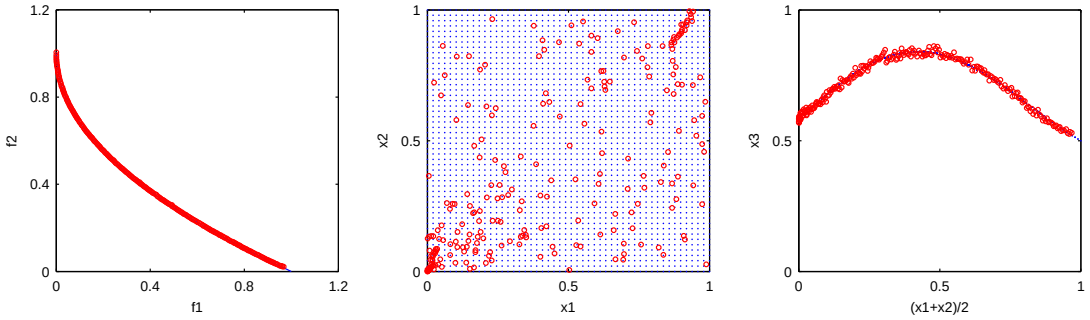
MMEA was designed for solving a MOP of Class II, in which the dimensionality of its PS is not lower than the number of the objectives and not known. In fact, MMEA uses the PCA technique to detect the PS dimensionality before modelling the PS in its search. Now a question arises whether MMEA can effectively solve a regular MOP in which the PS is a $(m-1)$ continuous manifold in the decision space. To study this question, we have tested MMEA on LZ08-F4 [32], a regular MOP,



(a) Omni-Optimizer



(b) KP1



(c) MMEA

Fig. 6. The best approximations obtained by three algorithms for F3. (a): Omni-Optimizer. (b): KP1. (c): MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions of the final solutions in the x_1-x_2 space obtained in the runs with the lowest $IGDX$ values. Right: the distributions of the final solutions in the $\frac{x_1+x_2}{2}$-x_3 space obtained in the runs with the lowest $IGDX$ values.

in which two objectives to be minimized are defined as:

$$f_1(x) = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}[x_j - 0.8x_1\cos(\frac{6\pi x_1+j\pi/n}{3})]^2$$
$$f_2(x) = 1-\sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}[x_j - 0.8x_1\cos(6\pi x_1+j\pi/n)]^2,$$

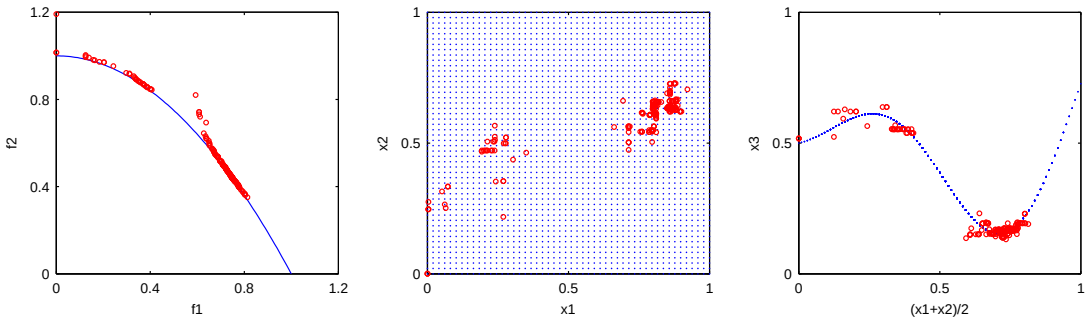
where $J_1 = \{j|j \text{ is odd and } 2 \le j \le n\}$ and $J_2 = \{j|j \text{ is even and } 2 \le j \le n\}$. Its search space is $[0,1]\times[-1,1]^{n-1}$. The PS of LZ08-F4 is:

$$x_1 \in [0,1] \text{ and } x_j = \begin{cases} 0.8x_1\cos(\frac{6\pi x_1+j\pi/n}{3}) & j\in J_1 \\ 0.8x_1\cos(6\pi x_1+j\pi/n) & j\in J_2 \end{cases},$$

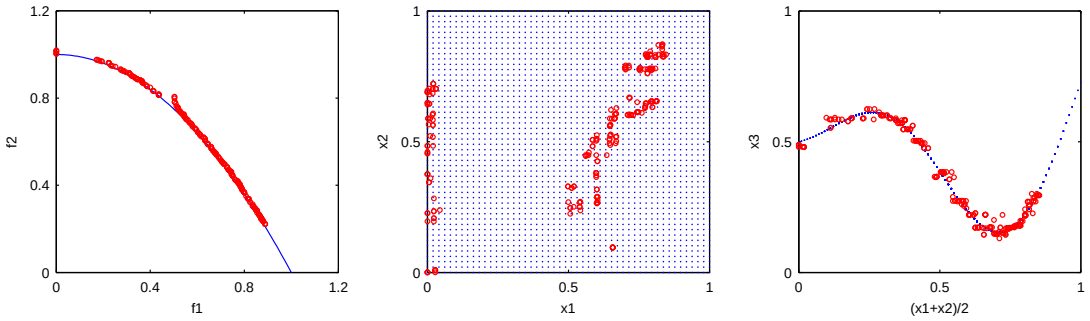
and its PF is:

$$f_2 = 1-\sqrt{f_1}, f_1 \in [0,1].$$

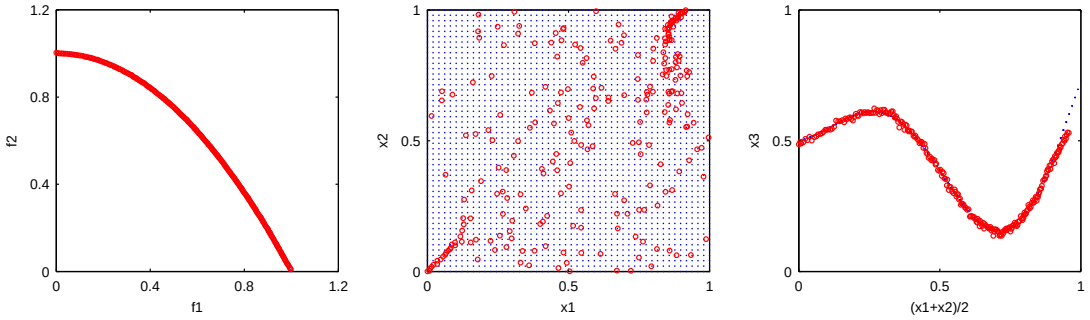
In our experiment on LZ08-F4, n, the number of decision variables is set to be 30, and N, the population size is 300 as suggested in [32]. All the other parameter settings are the same as in Section IV. Fig. 11 plots the final population obtained in the run with the lowest $IGDF$ value among 20 independent runs. It is evident that MMEA can solve LZ08-F4 effectively. Such success implies that MMEA is able to detect the dimensionality of the PS of a regular MOP like LZ08-F4 and a regular MOP does not pose a serious challenge to MMEA.



(a) Omni-Optimizer



(b) KP1



(c) MMEA

Fig. 7. The best approximations obtained by three algorithms for F4. (a): Omni-Optimizer. (b): KP1. (c): MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions of the final solutions in the x_1-x_2 space obtained in the runs with the lowest $IGDX$ values. Right: the distributions of the final solutions in the $\frac{x_1+x_2}{2}$-x_3 space obtained in the runs with the lowest $IGDX$ values.

B. Can MMEA deal with a MOP of Class I?

A MOP of Class II has a continuous PS of dimensionality larger than $m-1$, while the PS of a MOP of Class I consists of a number of disconnected continuous manifolds. To investigate the ability of MMEA to tackle MOPs in Class I, we have executed MMEA on DT05-F4.4 [24], in which the two objectives to be minimized are as follows:

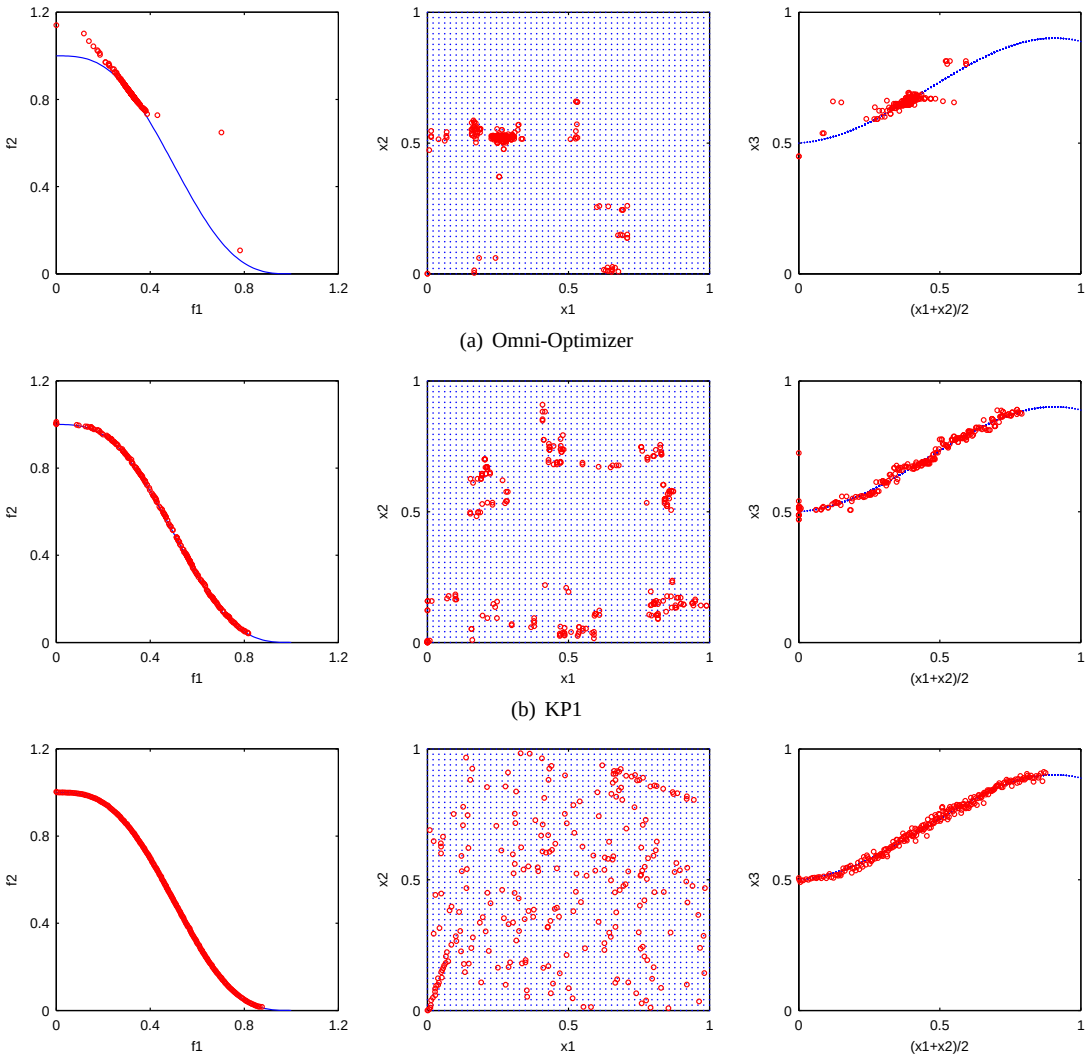
$$\begin{aligned} f_1(x) &= \textstyle\sum_{i=1}^{n} \sin(\pi x_i) \\ f_2(x) &= \textstyle\sum_{i=1}^{n} \cos(\pi x_i), \end{aligned}$$

and the search space is $[0,6]^n$. The PF of DT05-F4.4 is:

$$f_2 = -\sqrt{25 - f_1^2}, f_1 \in [-5, 0],$$

and its PS consists of 3^n disconnected parts, each of them is a line segment.

In our experiment on DT05-F4.4, n, the number of decision variables is set to be 5, and N, the population size is $1,000$. All the other parameter settings are the same as in Section IV. Fig. 12 presents the final population obtained in the run with the lowest $IGDF$ value among 20 independent runs. Clearly, MMEA have not produced a satisfactory approximation to the PS. This could be because that MMEA bases on the distance defined in the objective space to clusters its population, it has no ability to distinguish different parts of the DT05-F4.4 PS in the decision space and thus cannot find a good approximation to its PS.



(a) Omni-Optimizer

(b) KP1

(c) MMEA

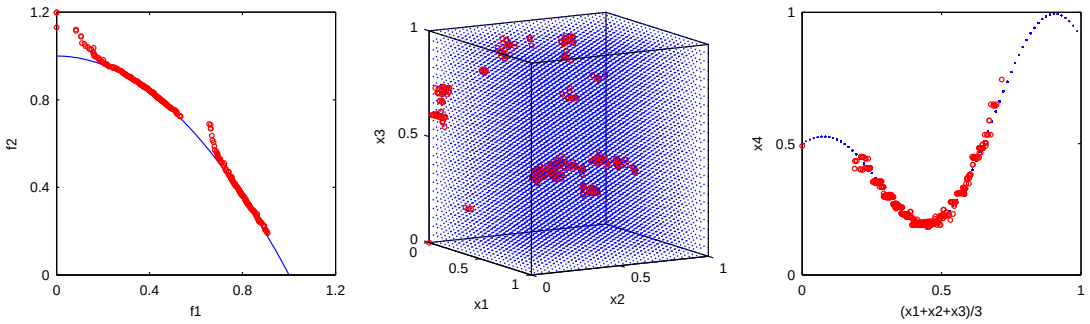
Fig. 8. The best approximations obtained by three algorithms for F5. (a): Omni-Optimizer. (b): KP1. (c): MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions of the final solutions in the x_1-x_2 space obtained in the runs with the lowest $IGDX$ values. Right: the distributions of the final solutions in the $\frac{x_1+x_2}{2}$-x_3 space obtained in the runs with the lowest $IGDX$ values.

VI. Conclusion

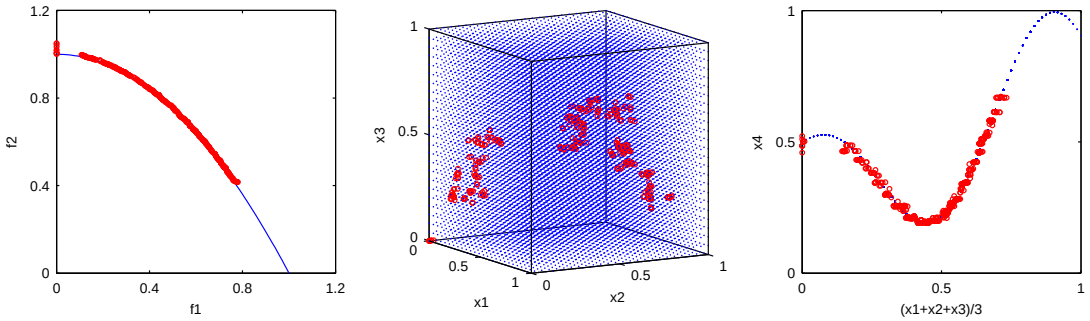
A good approximation to both the PS and PF of a MOP could be required in some real-world applications. A good approximation to the PF of a MOP might not be a good approximation to the PS, for example, when the MOP in question is of Class I or II. Some effort has been made to approximate both the PS and the PF of a MOP of Class I. This paper represents a first attempt to do so for a MOP of Class II.

MMEA proposed in this paper generalizes the idea used in RM-MEDA to a MOP of Class II for approximating its PS and PF simultaneously. In the modelling phase of MMEA, the population is clustered into a number of subpopulations based on their distribution in the objective space, the PCA technique is used to detect the dimensionality of the centroid of each subpopulation, and then a probabilistic model is built for modelling the distribution of the Pareto optimal solutions in the decision space. Such modelling procedure could promote the population diversity in both the decision and objective spaces. New solutions are sampled from the model thus built. To ease the burden of setting the number of subpopulations, a dynamic strategy for periodically adjusting the number of subpopulations has been adopted in MMEA. The population for the next generation is selected by the NDS-selection. The comparison between MMEA and the two other algorithms, KP1 and Omni-Optimizer on a set of test instances, some of which were proposed in this paper, have been made in this paper. It is very clear that MMEA has a big advantage over the two other algorithms in approximating both the PS and the PF of a MOP of Class II when the PS is a nonlinear manifold, although it might not be able to perform significantly better when the PS is a linear manifold. We have also investigated the ability of MMEA to deal with a regular MOP and a MOP of Class I.

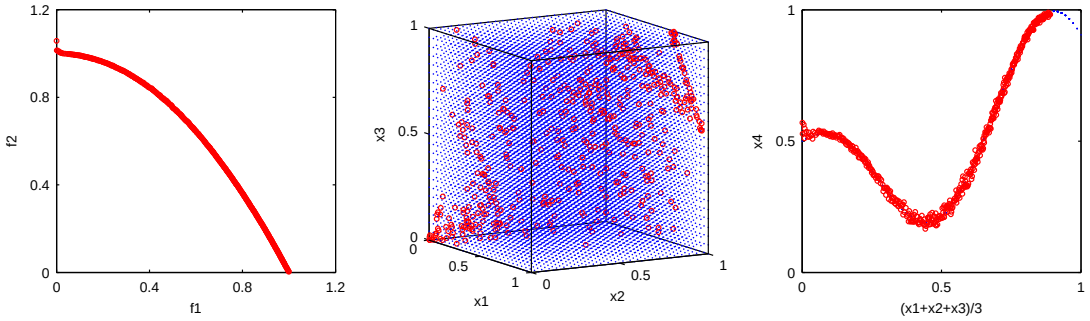
The future research topics along this line may include:



(a) Omni-Optimizer



(b) KP1



(c) MMEA

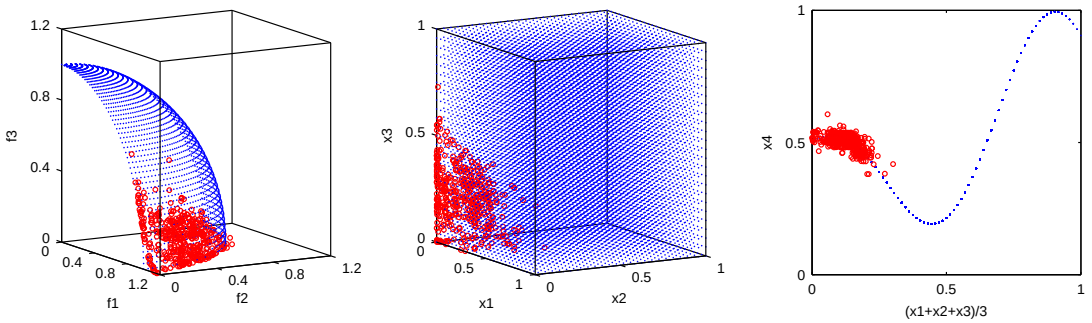
Fig. 9. The best approximations obtained by three algorithms for F6. (a): Omni-Optimizer. (b): KP1. (c): MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions of the final solutions in the x_1-x_2-x_3 space obtained in the runs with the lowest $IGDX$ values. Right: the distributions of the final solutions in the $\frac{x_1+x_2+x_3}{3}$-x_4 space obtained in the runs with the lowest $IGDX$ values.

- extension of MMEA to constrained MOPs, and MOPs under dynamic and/or noisy environment for approximating both their PS and PF [35], [36].
- study of the scalability of MMEA on the numbers of decision variables and objectives [37], [38].
- use of other machine learning methods in MMEA [39].
- combination of other techniques, particularly, traditional mathematical programming methods, with MMEA for improving the algorithm performance [40].

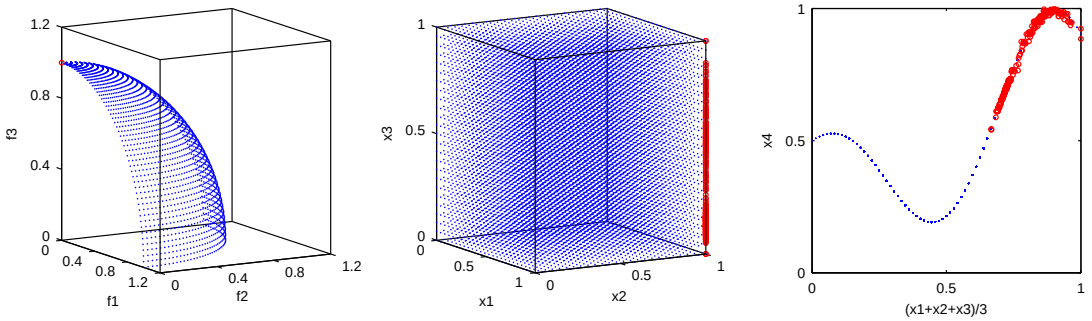
The C++ code of MMEA can be downloaded from Q. Zhang's home page: http://cswww.essex.ac.uk/staff/qzhang/

REFERENCES

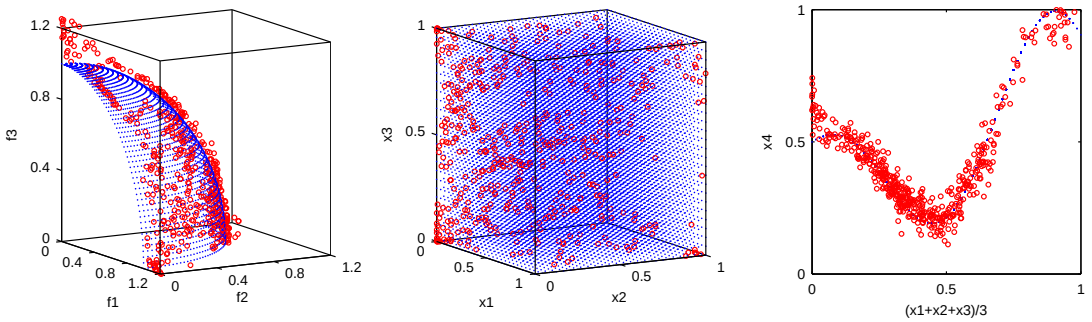
[1] K. Miettinen, *Nonlinear Multiobjective Optimization*, ser. Kluwer's International Series in Operations Research & Management Science. Kluwer Academic Publishers, 1999, vol. 12.
[2] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms*. Baffins Lane, Chichester: John Wiley & Sons, LTD, 2001.
[3] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Kluwer Academic Publishers, 2002.
[4] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proceedings of the Congress on Evolutionary Computation*, vol. 1. Washington D.C., USA: IEEE, 1999, pp. 98–105.
[5] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.
[6] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 3, pp. 253–274, 2003.



(a) Omni-Optimizer



(b) KP1



(c) MMEA

Fig. 10. The best approximations obtained by three algorithms for F7. (a): Omni-Optimizer. (b): KP1. (c): MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions of the final solutions in the x_1-x_2-x_3 space obtained in the runs with the lowest $IGDX$ values. Right: the distributions of the final solutions in the $\frac{x_1+x_2+x_3}{3}$-x_4 space obtained in the runs with the lowest $IGDX$ values.

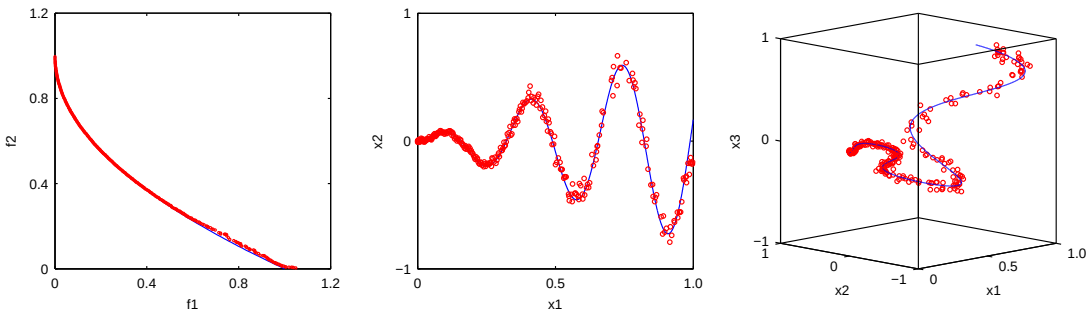


Fig. 11. The best approximation obtained in the 20 runs with the lowest $IGDF$ value by MMEA for LZ08-F4. Left: the distribution of the final solutions in the objective space. Middle: the distribution of the final solutions in the x_1-x_2 space. Right: the distribution of the final solutions in the x_1-x_2-x_3 space.

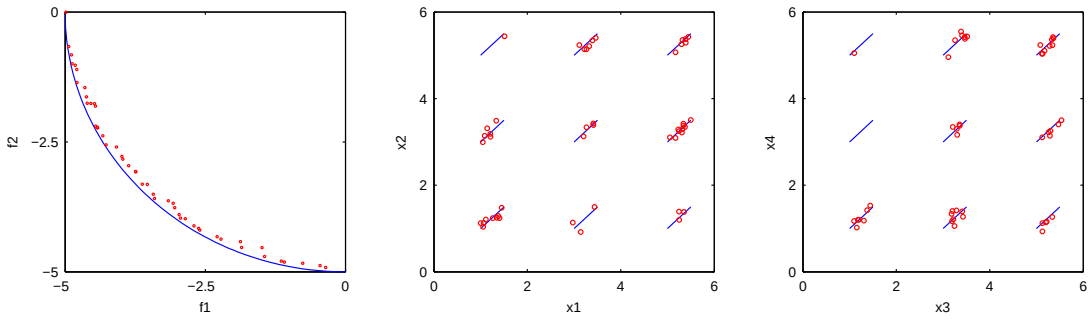


Fig. 12. The best approximation obtained in the 20 runs with the lowest $IGDF$ value by MMEA for DT05-F4.4. Left: the distribution of the final solutions in the objective space. Middle: the distribution of the final solutions in the x_1-x_2 space. Right: the distribution of the final solutions in the x_3-x_4 space.

[7] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, June 2004.

[8] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*, ser. Lecture Notes in Computer Science, vol. 3242. Birmingham, UK: Springer, September 2004, pp. 832–842.

[9] C. Gil, A. Márquez, R. Banos, M. G. Montoya, and J. Gómez, "A hybrid method for solving multi-objective global optimization problems," *Journal of Global Optimization*, vol. 38, no. 2, pp. 265–281, 2007.

[10] K. C. Tan, Y. J. Yang, and C. K. Goh, "A distributed cooperative coevolutionary algorithm for multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 527– 549, 2006.

[11] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evolutioanry Computation*, vol. 15, no. 1, pp. 1–28, 2007.

[12] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[13] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[14] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 269–283, 2008.

[15] K. I. S. adn Richard M. Everson, J. E. Fieldsend, C. Murphy, and R. Misra, "Dominance-based multiobjective simulated annealing," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 323–342, 2008.

[16] H. P. Benson and S. Sayin, "Optimization over the efficient set: Four special cases," *Journal of Optimization Theory and Applications*, vol. 80, no. 1, pp. 3–18, 1994.

[17] P. T. Thach, H. Konno, and D. Yokota, "Dual approach to minimization on the set of pareto-optimal solutions," *Journal of Optimization Theory and Applications*, vol. 88, no. 3, pp. 689–707, 1996.

[18] R. Horst and N. V. Thoai, "Utility function programs and optimization over the efficient set in multiple-objective decision making," *Journal of Optimization Theory and Applications*, vol. 92, no. 3, pp. 605–631, 1997.

[19] C. Hillermeier, *Nonlinear Multiobjective Optimization - A Generalized Homotopy Approach*. Birkhäuser, 2001.

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

[21] Y. Jin, "Effectiveness of weighted sum of the objectives for evolutionary multi-objective optimization: Methods, analysis and applications," 2002, on-line available at: http://www.soft-computing.de/edwa2002.pdf.

[22] G. Rudolph, B. Naujoks, and M. Preuss, "Capabilities of emoa to detect and preserve equivalent pareto subsets," in *Fouth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, ser. Lecture Notes in Computer Science, vol. 4403. Matsushima,Japan: Springer, March 2007, pp. 36–50.

[23] 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, 2006.

[24] K. Deb and S. Tiwari, "Omni-optimizer: A procedure for single and multi-objective optimization," in *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, ser. Lecture Notes in Computer Science, C. A. Coello Coello, A. H. Aguirre, and E. Zitzler, Eds., vol. 3410. Guanajuato, Mexico: Springer, March 2005, pp. 41–65.

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

[26] K. P. Chan and T. Ray, "An evolutionary algorithm to maintain diversity in the parametric and the objective space," in *Third International Conference on Computational Intelligence, Robotics and Autonomous Systems(CIRAS 2005)*, Singapore, 2005.

[27] M. Preuss, B. Naujoks, and G. Rudolph, "Pareto Set and EMOA behavior for simple multimodal multiobjective functions," in *Parallel Problem Solving from Nature - PPSN VIX*, ser. Lecture Notes in Computer Science, vol. 4193. Reykjavik, Iceland: Springer, September 2006, pp. 513–522.

[28] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1493–1516, October 1997.

[29] M. Reyes Sierra and C. A. Coello Coello, "A study of fitness inheritance and approximation techniques for multi-objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC 2005)*. Edinburgh, U.K: IEEE Press, September 2005, pp. 65–72.

[30] J. D. Knowles and D. W. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 100–116, 2003.

[31] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation and Control*. Barcelona, Spain: CIMNE, 2002, pp. 95–100.

[32] H. Li and Q. Zhang, "Comparison between NSGA-II and MOEA/D on a set of multiobjective optimization problems with complicated pareto sets," *IEEE Transactions on Evolutionary Computation*, 2008, accepted.

[33] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, 1995.

[34] K. Deb and M. Goyal, "A combined genetic adaptive search (geneas) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[35] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.

[36] A. Zhou, Q. Zhang, Y. Jin, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Fouth International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)*, ser. Lecture Notes in Computer Science, vol. 4403. Matsushima,Japan: Springer, March 2007, pp. 832–846.
[37] C. W. Ahn and R. S. Ramakrishna, "On the scalability of real-coded bayesian optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 3, pp. 307–322, 2008.
[38] L. Martí, J. García, A. Berlanga, and J. M. Molina, "Scalable continuous multiobjective optimization with a neural networkcbased estimation of distribution algorithm," in *Applications of Evolutionary Computing: EvoWorkshops 2008*, ser. Lecture Notes in Computer Science, M. Giacobini et al, Ed. Berlin/Heidelberg: Springer, 2008, no. 4974, pp. 535–544.
[39] R. S. Michalski, "Learnable evolution model: Evolutionary processes guided by machine learning," *Machine Learning*, vol. 38, no. 1-2, pp. 9–40, 2000.
[40] Q. Zhang, J. Sun, E. Tsang, and J. Ford, "Hybrid estimation of distribution algorithm for global optimisation," *Engineering Computations*, vol. 21, no. 1, pp. 91–107, 2003.