

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

Aimin Zhou, Qingfu Zhang, *Senior Member, IEEE*, and Yaochu Jin, *Senior Member, IEEE*

Abstract—Most existing multiobjective evolutionary algorithms aim at approximating the Pareto front (PF), which is the distribution of the Pareto-optimal solutions in the objective space. In many real-life applications, however, a good approximation to the Pareto set (PS), which is the distribution of the Pareto-optimal solutions in the decision space, is also required by a decision maker. This paper considers a class of multiobjective optimization problems (MOPs), in which the dimensionalities of the PS and the PF manifolds 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 an MOP in this class. In the modeling phase of MMEA, the population is clustered into a number of subpopulations based on their distribution in the objective space, the principal component analysis technique is used to estimate the dimensionality of the PS manifold in each subpopulation, and then a probabilistic model is built for modeling the distribution of the Pareto-optimal solutions in the decision space. Such a modeling procedure could promote the population diversity in both the decision and objective spaces. MMEA is compared with three other methods, KP1, Omni-Optimizer and RM-MEDA, on a set of test instances, five of which are proposed in this paper. The experimental results clearly suggest that, overall, MMEA performs significantly better than the three compared algorithms in approximating both the PS and the PF.

Index Terms—Estimation of distribution algorithm, multiobjective optimization, Pareto optimality, principal component analysis.

I. INTRODUCTION

THIS PAPER considers the following *continuous multiobjective optimization problem* (continuous MOP):

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

where $-\infty < a_i < b_i < +\infty$ for all $i = 1, \dots, n$. $\prod_{i=1}^n [a_i, b_i] \subset R^n$ is the decision space, and $x =$

Manuscript received July 1, 2008; revised January 11, 2009 and April 6, 2009; accepted April 8, 2009. Current version published September 30, 2009.

A. Zhou is with the Department of Computer Science and Technology, East China Normal University, Shanghai 200241, China (e-mail: amzhou@cs.ecnu.edu.cn).

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

Y. Jin is with the Honda Research Institute Europe, Offenbach 63073, Germany (e-mail: yaochu.jin@honda-ri.de).

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

Digital Object Identifier 10.1109/TEVC.2009.2021467

$(x_1, \dots, x_n)^T \in R^n$ 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 $f_i(x)$, $i = 1, \dots, m$. R^m is the objective space.

Let $u = (u_1, \dots, u_m)^T$, $v = (v_1, \dots, v_m)^T \in R^m$ be two vectors; u is said to *dominate* v , if $u \neq v$ and $u_i \leq v_i$ for all $i = 1, \dots, m$. x^* is called (*globally*) *Pareto-optimal* if there is no other x such that $F(x)$ dominates $F(x^*)$. The set of all the Pareto-optimal points, denoted by PS, is called the *Pareto set* (PS). The image of the PS on the objective space, $PF = \{y \in R^m | y = F(x), x \in PS\}$, is called the *Pareto front* (PF) [1], [2].

Most existing *multiobjective evolutionary algorithms* (MOEAs) aim at approximating PFs [2]–[16]. 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 [17]–[20]. For example, if two objectives f_1 and f_2 are much more important than objective v in engineering design, one often needs to first optimize f_1 and f_2 and obtain a good approximation to both the PF and the PS, then finds from the approximate PS a solution that optimizes v subject to certain constraints as their final solution. In some cases, a good approximation to the PF might not approximate the PS well. Two typical classes of continuous MOPs, in which the approximation of their PSs should be carefully addressed, are as follows.

- 1) 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. ZDT6 [21], Jin1 [22], and the SYM-PART instances [23] are test instances in this class. In all these instances, the PS consists of a number of disconnected continuous $(m - 1)$ -D manifolds.
- 2) Class II: The PF is an $(m - 1)$ -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 nonzero-dimensional continuous manifold. Some WFG test instances [24] belong to this class. For example, in WFG6 with some parameter setting, the PF is a 1-D continuous curve in the objective space, while the PS is a 2-D rectangle. The inverse image of a point in the PF is a 1-D curve in the decision space.

To generate a good approximation to both the PS and the PF of an MOP, an MOEA 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 [25] introduced the crowding distance in the decision space into the nondominated sorting scheme in Omni-Optimizer, which is a generalization of NSGA-II [26], for promoting the population diversity in the decision space. Chan and Ray [27] suggested using two selection operators in MOEAs; one encourages the diversity in the objective space and the other does so in the decision space. 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. Preuss *et al.* [28] and Rudolph *et al.* [23] also proposed to use a restart strategy for finding a good approximation to the PS of an MOP of Class I. 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 an MOP of Class II.

In [14], we studied a “regular” continuous MOP in which both the PF and the FS are piecewise $(m - 1)$ -D continuous manifolds, and proposed RM-MEDA, which is an *estimation of distribution algorithm* (EDA) for approximating its PF. In this paper, we generalize the idea of RM-MEDA and propose a *probabilistic model-based multiobjective evolutionary algorithm*, called MMEA, for approximating the PS and the PF of an MOP of Class II simultaneously. MMEA has the following features.

- 1) 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 non-dominated sorting (NDS) selection, which is used in RM-MEDA, is employed in MMEA.
- 2) 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 random strategy is used for setting it.
- 3) The *principal component analysis* (PCA) technique is used to estimate the dimensionality of the PS manifold in each subpopulation, and then a probabilistic model can be built for modeling the distribution of promising solutions 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 proposed in this paper for the first time. Section IV compares MMEA with KP1, Omni-Optimizer, and RM-MEDA on these test instances. More discussions on the ability of MMEA are provided in Section V. Section VI concludes this paper and suggests some future research topics.

II. ALGORITHM

A. Framework

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

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

$$x^1, \dots, x^N;$$
- 2) their function values: $F(x^1), \dots, F(x^N)$.

MMEA adopts the following widely used EDA framework.

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

Phase 2 Modeling: Build a model for modeling 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 function values of these solutions in Q .

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

Phase 5 Stopping Condition: If a stopping 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 details of modeling, reproduction, and selection.

B. Modeling

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. This 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 modeling phase of MMEA for dealing with an MOP of Class II.

The modeling phase in MMEA works as follows.

Step 1 Building a Utopian PF: Based on information from the current population P , build an $(m - 1)$ -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 modeling the PS.

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

Step 4 Clustering: Cluster the population P into K subpopulations P^1, \dots, P^K .

Step 5 Principal Component Analysis and Modeling: Perform PCA on each subpopulation P^k , $k = 1, \dots, K$ and build a model for it.

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

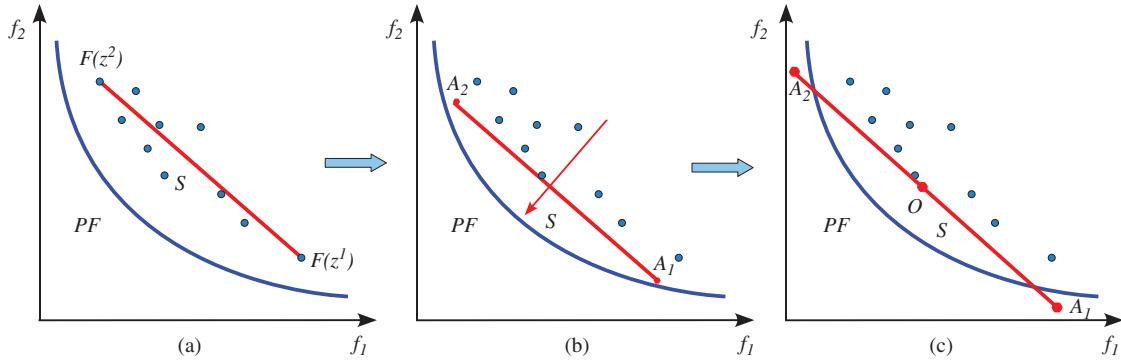


Fig. 1. Illustration of building a Utopian PF in the case of two objectives. (a) Find the extreme points and initialize simplex S , (b) Move S along its normal direction, and (c) Enlarge S .

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 an $(m - 1)$ -D simplex as a Utopian PF. The following procedure is used to construct such a simplex S .

Step 1.1 For $i = 1, \dots, m$, 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 $F(z^1), \dots, F(z^m)$ 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, \dots, 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

$$A_i := A_i + [(1 + \alpha)^{\frac{1}{m-1}} - 1](A_i - O)$$

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

$\alpha > 0$ is a control parameter. It is easy to work out that the volume of the simplex S is increased by $100\alpha\%$ in **Step 1.3**. 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 1-D line segment in the objective space. Fig. 1 illustrates how S is generated in this case.

2) Determining the Number of Subpopulations: To reduce the problem-dependence of K , the value of K is uniformly randomly chosen from the set $\{1, 2, \dots, K_{\max}\}$. K_{\max} is a control parameter.

3) Selecting Reference Points: It is desirable that reference points uniformly spread on the Utopian PF. Note that the Utopian PF is a simplex; in our implementation, we use the simplex point picking method [29] for selecting reference points.

4) Clustering: For each reference point Y^k obtained in **Step 3**, we select a number of points from P closer to it for forming P^k as follows.

Step 4.1 Compute, in the objective space, the Euclidian distances between Y^k to all the individual solutions in P .

Step 4.2 Select the $\min\{N, [2N/K]\}$ closest solutions to Y^k and let them constitute subpopulation P^k .

In clustering, different subpopulations may overlap, which could improve the search performance in between different reference points.

5) PCA and Modeling: The individual solutions in subpopulation P^k , $k = 1, \dots, K$ should, hopefully, scatter around the PS in the decision space as the search goes on. For simplicity, we can model the subpopulation P^k as a hyper-cuboid Φ^k in the decision space and regard each individual in P^k as an observation of the following random vector

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

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

Now the task is to estimate Φ^k and σ^k . We do it as follows.

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

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

and

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

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

Step 5.2 Compute the eigenvalues of Cov^k

$$\lambda_1^k \geq \lambda_2^k \geq \dots \geq \lambda_n^k$$

and their corresponding unity eigenvectors

$$V_1^k, V_2^k, \dots, V_n^k.$$

Step 5.3 Set n^k , the dimensionality of the hyper-cuboid Φ^k to be the smallest integer such that

$$\sum_{j=1}^{n^k} \lambda_j^k \geq \theta \sum_{j=1}^n \lambda_j^k$$

where the threshold $0 \leq \theta \leq 1$ is an algorithm parameter.

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

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

and

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

for $j = 1, \dots, n^k$.

Step 5.5 Set

$$\begin{aligned} \Phi^k = \left\{ x \in R^n \mid x = \bar{x}^k + \sum_{j=1}^{n^k} c_j V_j^k, \right. \\ l_j^k - \frac{1}{2} \frac{(1+\beta)n^k - 1}{2} (u_j^k - l_j^k) \leq c_j \\ \left. \leq u_j^k + \frac{1}{2} \frac{(1+\beta)n^k - 1}{2} (u_j^k - l_j^k), \right. \\ \left. j = 1, \dots, n^k \right\} \end{aligned}$$

where β is a control parameter.

Step 5.6 Set

$$\sigma^k = \frac{1}{n - n^k} \sum_{j=n^k+1}^n \lambda_j^k.$$

The dimensionality of the PS is unknown, neither is that of Φ^k . In **Step 5.3**, the dimensionality of Φ^k is set such that Φ^k holds at least $100\theta\%$ of the variation in the solutions in P^k . In **Step 5.5**, Φ^k is enlarged along each of the first n^k principal component directions such that its volume is $100\beta\%$ larger than that the smallest n^k -D hyper-cuboid containing the projections of all the solutions of P^k on the space spanned by $V_1^k, \dots, V_{n^k}^k$ from \bar{x}^k . The motivation behind this extension is to extrapolate the points in P^k for searching unexplored promising areas in the decision space. ε is modeled as a Gaussian noise vector and all its components are i.i.d., which facilitates the sampling procedure.

The reference points in the objective space used in clustering are hopefully uniformly distributed along the PF, therefore they could guide the search to generate a good approximation to the PF in the objective space. The modeling in the decision space attempts to model the PS, which enables the algorithm to generate a set of points for approximating the PS in the decision space.

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

- 1) RM-MEDA uses the local PCA [30] technique to partition the population into several clusters. In contrast, MMEA in this paper selects the subpopulation centers from the Utopian PF and performs 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 the PS manifold must be predetermined in the local PCA and it is unknown in the problems MMEA aims to solve.
- 2) The number of clusters is preset in RM-MEDA, while MMEA in this paper chooses the number of subpopulations randomly, which lightens the burden of tuning this control parameter.
- 3) In modeling each subpopulation, RM-MEDA sets the dimensionality of the PS manifold 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 $\{1, 2, \dots, K\}$.

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

Step 3 Set $y = x' + \varepsilon'$, and let the new solution x as

$$x_j = \begin{cases} y_j, & \text{if } a_j \leq y_j \leq b_j \\ \frac{1}{2}(z_j + a_j), & \text{if } y_j < a_j \\ \frac{1}{2}(z_j + b_j), & \text{if } y_j > b_j \end{cases}$$

where $j = 1, \dots, n$, and z is a randomly selected solution from the subpopulation P^k .

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, which is a variant of non-dominated sorting scheme [26] proposed in [14]. It works as follows.

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

Step 2 Partition Q into different fronts F_1, \dots, F_l by using the fast non-dominated sorting approach [26]. Set $k = 0$.

Do

$$\begin{aligned} k &= k + 1, \\ P &= P \cup F_k, \end{aligned}$$

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 is more than one member with the smallest crowding distance, randomly choose one and remove it.

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

The crowding distance, used in **Step 3**, of point x is defined as the average side length of the largest m -D rectangle in the objective space subject to two constraints: (a) each of its sides is parallel to a coordinate axis and (b) $F(x)$ is the only interior point of the rectangle among all the points in $\{F(y)|y \in F_k \cap P\}$. 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 an MOP of Class II. WFG instances are the only Class II instances that we have found in the literature. Two WFG instances: WFG6 and WFG7, have been used in our experiments. Based on the experiments in [24], the PFs of WFG7 could be “easily” and “quickly” found by NSGA-II, while WFG6 is “hard” for NSGA-II. It can be because the objectives in WFG7 are separable while it is not the case in WFG6. The PSs of these two test instances are a 2-D rectangle in the decision space when their control parameters are set as in Table I. To study the behaviors of MOEAs on nonlinear PSs, we have designed several new 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 dimensional spaces.

B. Performance Metrics

The inverted generational distance (IGD) metric [14], [31] and hypervolume difference (I_H^-) metric [32] are 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 to the PF (or the PS). The IGD 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^* .

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

The I_H^- metric is defined as

$$I_H^-(P^*, P) = I_H(P^*) - I_H(P)$$

where $I_H(P)$ is the hypervolume between the set P and a bounded reference point [33].

Both the IGD metric and the I_H^- metric measure convergence and diversity. To have low IGD and I_H^- values, P must be close to the PF (or PS) and cannot miss any part of the whole PF (or PS).

In our experiments, 1000 points, in which f_1 or t taking 1000 equidistant values from their lower bounds to their upper bounds, are selected from the respective PFs of $F1-F6$ to be P^* for computing the $IGDF$ metrics. $50 \times 50 = 2500$ points in the PF of $F7$ with $s, t = 0/49 \times \pi/2, 1/49 \times \pi/2, \dots, 49/49 \times \pi/2$, are taken to form P^* for computing the $IGDF$ metric for experiments on $F7$. $50 \times 50 = 2500$ 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^* for computing the $IGDX$ metrics. $25 \times 25 \times 25 = 15625$ 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^* for computing the $IGDX$ metrics. In calculating the I_H^- values, the bounded reference point is chosen to be $(2.4, 4.5)^T$ for $F1$ and $F2$, $(1.2, 1.2)^T$ for $F3-F6$ and $(1.2, 1.2, 1.2)^T$ for $F7$.

IV. EXPERIMENTAL RESULTS

A. Experimental Settings and Algorithms in Comparison

The studies in [23], [27], [28] have shown that popular MOEAs, such as PAES [34], NSGA-II [26], and SPEA2 [35], cannot approximate both the PF and the PS simultaneously since these methods cannot maintain the population diversity in the decision space. MOEA/D, which is a recent MOEA based on aggregation proposed in [14], [36], cannot do so either for the same reason. In our experiments, we compare MMEA with KP1 [27]¹ and Omni-Optimizer [25].² As mentioned in Introduction, both KP1 and Omni-Optimizer try to approximate both the PF and the PS of an MOP by promoting the population diversity in the decision space in their selection operators. The simulated binary crossover (SBX) [37] and the polynomial mutation [38] are used in these two methods for generating offspring. Since MMEA is based on RM-MEDA [14], we also compare MMEA with RM-MEDA on these problems.

Table II lists all the parameter settings in our experiments. The population in each algorithm is initialized uniformly and randomly in the decision space. All the following results are based on 20 independent runs of each algorithm on each test instance.

B. $F1-F2$

$F1$ and $F2$ have the same PS, which is a 2-D rectangle parallel to the $x_1 - x_2$ space. The objectives are nonseparable in $F1$ but separable in $F2$ [24]. The means and standard deviations can be found in Table III of the I_H^- , $IGDF$, 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.

¹We use KP1 in this paper because the experimental results in [27] have shown that KP1 is slightly better than KP2.

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

TABLE I

TEST INSTANCES USED IN OUR EXPERIMENTS: F_1, F_2 ARE WFG INSTANCES, F_3-F_7 ARE NEW DESIGNED TEST INSTANCES. $x = (x_1, \dots, x_n)$. ALL THESE INSTANCES BELONG TO CLASS II

Instance	Range of x_i	Objectives, PS and PF	Remarks
F_1	$[0, 2i]$	$WFG6 (M = 2, k = 2)$ [24] PS: $x_i = 0.7i$, for $i = 3, \dots, n$, $0 \leq x_1 \leq 2$, $0 \leq x_2 \leq 4$. PF: $f_1 = 2 \sin(t)$, $f_2 = 4 \cos(t)$, $0 \leq t \leq 0.5\pi$.	PS is a 2-D rectangle. PF is concave. two objectives.
F_2	$[0, 2i]$	$WFG7 (M = 2, k = 2)$ [24] PS: $x_i = 0.7i$, for $i = 3, \dots, n$, $0 \leq x_1 \leq 2$, $0 \leq x_2 \leq 4$. PF: $f_1 = 2 \sin(t)$, $f_2 = 4 \cos(t)$, $0 \leq t \leq 0.5\pi$.	PS is a 2-D rectangle. PF is concave. two objectives.
F_3	$[0, 1]$	$f_1(x) = (x_1 + x_2)/2$, $f_2(x) = g(x)(1 - \sqrt{\frac{f_1}{g}})$, 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 $i = 3, \dots, n$, and $0 \leq x_1, x_2 \leq 1$. PF: $f_2 = 1 - \sqrt{f_1}$, $0 \leq f_1 \leq 1$.	PS is a 2-D nonlinear surface. PF is convex. two objectives.
F_4	$[0, 1]$	$f_1(x) = (x_1 + x_2)/2$, $f_2(x) = g(x) - f_1^2$, 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: $x_i = \begin{cases} 0.5 + 0.5f_1 \cos(2\pi f_1 + i\pi/n), & i \text{ is even}, \\ 0.5 + 0.5f_1 \sin(2\pi f_1 + i\pi/n), & i \text{ is odd}, \end{cases}$ for $i = 3, \dots, n$, and $0 \leq x_1, x_2 \leq 1$. PF: $f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$.	PS is a 2-D nonlinear surface. PF is concave. two objectives.
F_5	$[0, 1]$	$f_1(x) = (x_1 + x_2)/2$, $f_2(x) = g(x) - f_1 + \sin(2\pi f_1)/(2\pi)$, 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: $x_i = \begin{cases} 0.5 + 0.5f_1 \cos(2\pi f_1 + i\pi/n), & i \text{ is even}, \\ 0.5 + 0.5f_1 \sin(\frac{1}{3}(2\pi f_1 + i\pi/n)), & i \text{ is odd}, \end{cases}$ for $i = 3, \dots, n$, and $0 \leq x_1, x_2 \leq 1$. PF: $f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$.	PS is a 2-D nonlinear surface. PF is neither concave nor convex. two objectives.
F_6	$[0, 1]$	$f_1(x) = (x_1 + x_2 + x_3)/3$, $f_2(x) = g(x) - f_1^2$, where $g(x) = 1 + \frac{5}{n-3} \sum_{i=4}^n h(x_i)^2$, and $h(x_i) = \begin{cases} 2x_i - \sin(0.5f_1\pi) \cos(2\pi f_1 + i\pi/n) - 1, & i \text{ is even}, \\ 2x_i - \cos(0.5f_1\pi) \sin(\frac{1}{3}(2\pi f_1 + i\pi/n)) - 1, & i \text{ is odd}. \end{cases}$ 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 $i = 4, \dots, n$, and $0 \leq x_1, x_2, x_3 \leq 1$. PF: $f_2 = 1 - f_1^2$, $0 \leq f_1 \leq 1$.	PS is a 3-D continuous nonlinear manifold. PF is concave. two objectives.
F_7	$[0, 1]$	$f_1(x) = g(x) \cos(0.25\pi(x_1 + x_2)) \sin(0.5\pi x_3)$, $f_2(x) = g(x) \cos(0.25\pi(x_1 + x_2)) \cos(0.5\pi x_3)$, $f_3(x) = g(x) \sin(0.25\pi(x_1 + x_2))$, 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: $x_i = \begin{cases} 0.5 + 0.5 \sin(0.5f_1\pi) \cos(2\pi f_1 + i\pi/n), & i \text{ is even}, \\ 0.5 + 0.5 \cos(0.5f_1\pi) \sin(\frac{1}{3}(2\pi f_1 + i\pi/n)), & i \text{ is odd}, \end{cases}$ for $i = 4, \dots, n$, and $0 \leq x_1, x_2, x_3 \leq 1$. PF: $f_1 = \cos(s) \sin(t)$, $f_2 = \cos(s) \cos(t)$, $f_3 = \sin(s)$, $0 \leq s, t \leq \pi/2$.	PS is a 3-D continuous nonlinear manifold. PF is concave. three objectives.

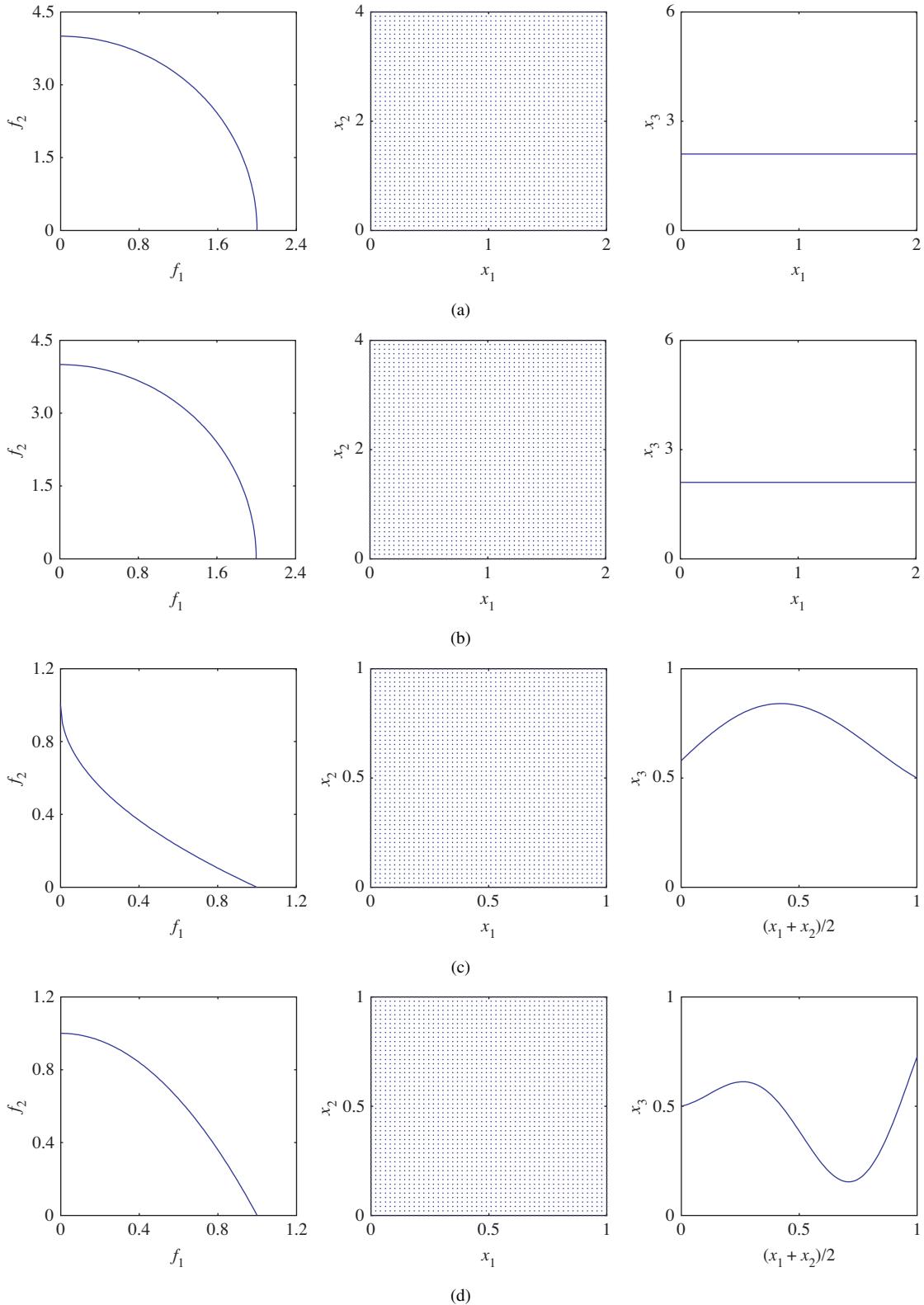


Fig. 2. PFs and the PSs for F1–F4. (a) F1, (b) F2, (c) F3, and (d) F4. Left: the PFs in the objective space. Middle: the projections of the PSs onto the x_1 – x_2 space. Right: the projections of the PSs onto the x_1 – x_3 space for F1 and F2, and onto the $[(x_1+x_2)/2]$ – x_3 space for F3 and F4.

It is clear from Table III that in terms of the *IGDF* metric, MMEA is significantly better than the three other algorithms on these two test instances, and in terms of I_H^- metric MMEA performs better than the three other competitors on F1 but slightly worse than KP1 on F2. The plots in Figs. 4 and 5

show that all the four algorithms can approximate the PF very well and the final populations with the lowest *IGDF* values, obtained by MMEA approximate the PFs slightly better than those obtained Omni-Optimizer and KP1. It should be pointed out that Figs. 4 and 5 do not contradict with Table III. Figs. 4

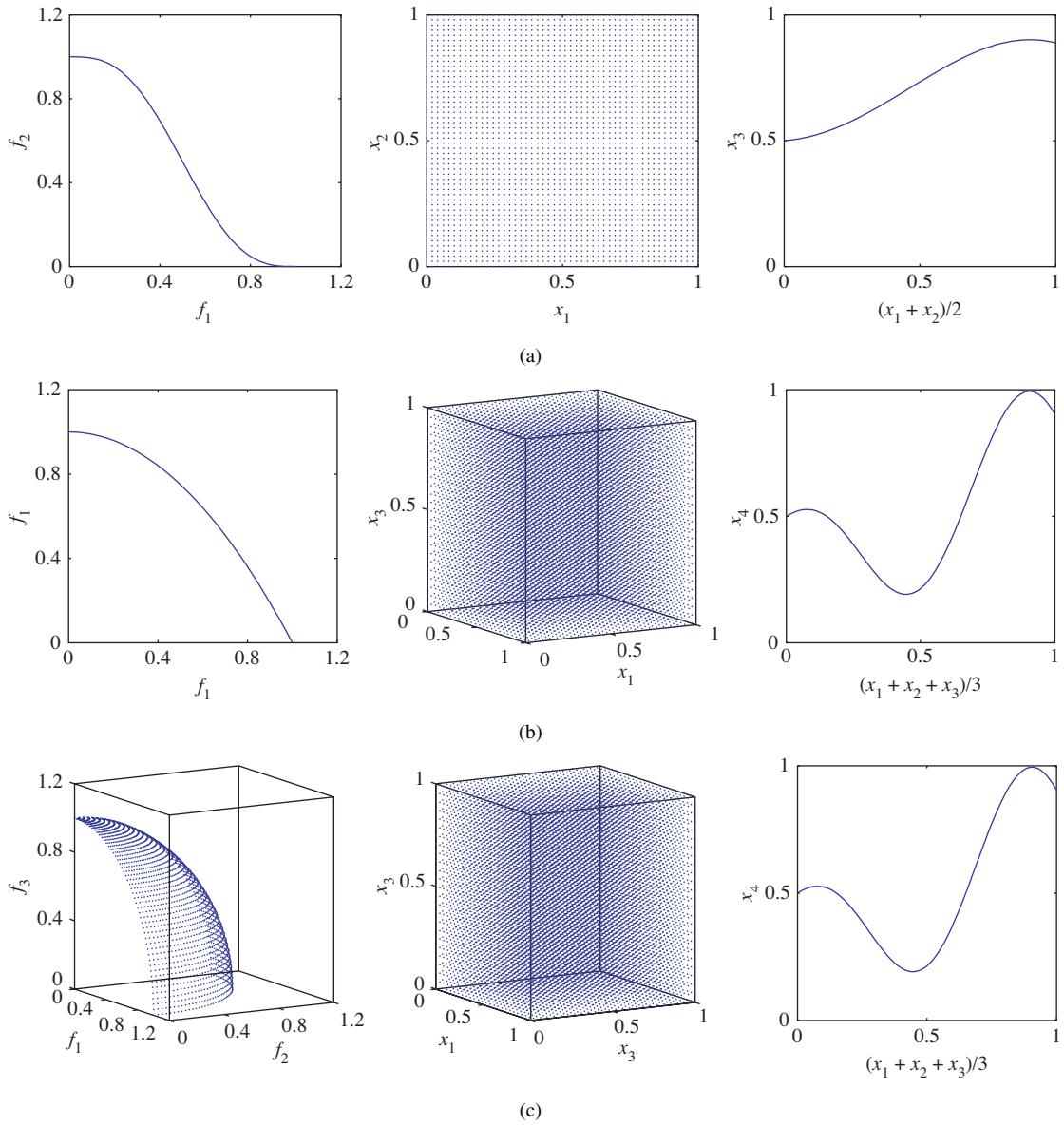


Fig. 3. PFs and the PSs for $F5-F7$. (a) $F5$, (b) $F6$, and (c) $F7$. Left: PFs in the objective space. Middle: the projections of the PSs onto the $x_1 - x_2$ space for $F5$, and onto the $x_1 - x_2 - x_3$ space for $F6$ and $F7$. Right: the projections of the PSs onto the $[(x_1 + x_2)/2] - x_3$ space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

and 5 give the distributions of the final populations with the lowest IGD values while, Table III shows the mean/std of the IGD and I_H^- values.

In terms of the $IGDX$ metric, Table III shows that MMEA significantly outperforms the three other algorithms. 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 the three other methods on $F1$ and $F2$: the distributions of the final population found by MMEA are more diverse and uniform than those obtained by the three others. These results indicate that MMEA could tackle MOPs with linear PSs like $F1$ and $F2$.

C. $F3-F7$

All these test instances have nonlinear PSs in the decision space. The dimensionality of the PSs of $F3-F5$ is 2 while that of $F6$ and $F7$ is 3.

$IGDF$ and IH^- are for measuring the approximation quality in the decision space. The t-test results in Table III suggest that on $F3-F7$, in terms of these two metrics, MMEA performs significantly better than KP1 and Omni-Optimizer, but does not always outperform RM-MEDA. It is confirmed to a certain extent by plots in Figs. 6–10: on $F3-F6$, that the final solutions with the lowest $IGDF$ values obtained by MMEA and RM-MEDA approximate the PFs very well while Omni-Optimizer and KP1 always miss part of the PFs; on $F7$, it is clear that MMEA and RM-MEDA provide better approximations than Omni-Optimizer and KP1, although none could approximate the PF very well. In terms of the $IGDX$ metric, it is evident from Table III that MMEA outperforms the three other algorithms on $F3-F7$, except RM-MEDA on $F7$. Figs. 6–9 also reveal that the solutions generated by MMEA, are distributed more uniformly in the decision space than those obtained by the three other ones. Table III shows

TABLE II
EXPERIMENTAL SETTINGS FOR $F1-F7$

	Number of variables	$n = 20$
	Population size for each algorithm	$N = 250 (F1-F5)$ $N = 500 (F6-F7)$
	Number of generations	500
	Number of runs for each algorithm	20
Omni-Optimizer KP1	Crossover parameter in SBX	$\eta_c = 20$
	Crossover rate	$P_c = 0.8$
	Parameter in polynomial mutation	$\eta_m = 20$
	Mutation rate	$P_m = 1/n$
RM-MEDA MMEA	Number of clusters	$K = 10$
	α in building a Utopian PF (Step 1.3)	$\alpha = 1.0$
	β in PCA and modeling (Step 5.5)	$\beta = 1.0$
	θ in PCA and modeling (Step 5.3)	$\theta = 0.8$
	K_{\max} in selecting reference points	$K_{\max} = 30$

TABLE III

STATISTICAL RESULTS ON $F1-F7$ ($mean \pm std.$) “+” IN PARENTHESIS: THE ONE-SIDE T-TEST INDICATES THAT THE METRIC VALUE IS LARGER THAN THAT OBTAINED BY MMEA AT THE 95% SIGNIFICANCE LEVEL. “−” IN PARENTHESIS MEANS A FAILURE IN THE T-TEST TO REJECT THE NULL HYPOTHESIS. IN THE ONE-SIDE T-TEST, THE NULL HYPOTHESIS IS THAT BOTH METRICS VALUES IN TEST ARE FROM THE SAME NORMAL DISTRIBUTION, AND THE ALTERNATIVE HYPOTHESIS IS THAT THE METRIC VALUE IN MMEA IS SMALLER THAN THAT OBTAINED BY ANOTHER ALGORITHM

		OMNI	KP1	RM-MEDA	MMEA
$F1$	IH^-	$0.4653 \pm 0.0867 (+)$	$0.3364 \pm 0.0699 (+)$	$0.0598 \pm 0.0048 (+)$	0.0452 ± 0.0127
	$IGDF$	$0.0719 \pm 0.0150 (+)$	$0.0606 \pm 0.0119 (+)$	$0.0110 \pm 0.0008 (+)$	0.0079 ± 0.0016
	$IGDX$	$26.7926 \pm 4.6930 (+)$	$23.9950 \pm 7.4775 (+)$	$0.4967 \pm 0.0984 (+)$	0.2321 ± 0.0683
$F2$	IH^-	$0.2760 \pm 0.0931 (+)$	0.0288 $\pm 0.0012 (-)$	$0.0557 \pm 0.0041 (+)$	0.0298 ± 0.0044
	$IGDF$	$0.0356 \pm 0.0139 (+)$	$0.0108 \pm 0.0009 (+)$	$0.0101 \pm 0.0005 (+)$	0.0065 ± 0.0004
	$IGDX$	$0.3174 \pm 0.0336 (+)$	$0.2759 \pm 0.0325 (+)$	$0.4293 \pm 0.0415 (+)$	0.2159 ± 0.0430
$F3$	IH^-	$0.5412 \pm 0.0095 (+)$	$0.0842 \pm 0.0241 (+)$	0.0055 $\pm 0.0039 (-)$	0.0059 ± 0.0042
	$IGDF$	$0.5609 \pm 0.0154 (+)$	$0.0776 \pm 0.0290 (+)$	0.0026 $\pm 0.0018 (-)$	0.0029 ± 0.0019
	$IGDX$	$1.0458 \pm 0.0134 (+)$	$0.4078 \pm 0.0752 (+)$	$0.2497 \pm 0.0195 (+)$	0.1073 ± 0.0100
$F4$	IH^-	$0.4220 \pm 0.0708 (+)$	$0.1920 \pm 0.1213 (+)$	$0.0760 \pm 0.0631 (+)$	0.0408 ± 0.0685
	$IGDF$	$0.3310 \pm 0.1038 (+)$	$0.1146 \pm 0.1040 (+)$	$0.0304 \pm 0.0319 (-)$	0.0179 ± 0.0392
	$IGDX$	$0.8415 \pm 0.1613 (+)$	$0.4421 \pm 0.2019 (+)$	$0.2795 \pm 0.0544 (+)$	0.1293 ± 0.0892
$F5$	IH^-	$0.3661 \pm 0.1045 (+)$	$0.0297 \pm 0.0042 (+)$	$0.0122 \pm 0.0084 (+)$	0.0080 ± 0.0043
	$IGDF$	$0.2677 \pm 0.0973 (+)$	$0.0300 \pm 0.0032 (+)$	$0.0145 \pm 0.0072 (+)$	0.0097 ± 0.0045
	$IGDX$	$0.6291 \pm 0.1176 (+)$	$0.2512 \pm 0.0278 (+)$	$0.2149 \pm 0.0322 (+)$	0.0918 ± 0.0105
$F6$	IH^-	$0.2381 \pm 0.0891 (+)$	$0.1757 \pm 0.0291 (+)$	0.0025 $\pm 0.0003 (-)$	0.0068 ± 0.0016
	$IGDF$	$0.1366 \pm 0.0707 (+)$	$0.0887 \pm 0.0217 (+)$	0.0012 $\pm 0.0001 (-)$	0.0039 ± 0.0009
	$IGDX$	$0.5760 \pm 0.1501 (+)$	$0.4484 \pm 0.0512 (+)$	$0.3405 \pm 0.0240 (+)$	0.1616 ± 0.0275
$F7$	IH^-	$0.6055 \pm 0.0816 (+)$	$0.9028 \pm 0.0000 (+)$	0.0866 $\pm 0.0046 (-)$	0.1055 ± 0.0056
	$IGDF$	$0.6073 \pm 0.0869 (+)$	$0.7451 \pm 0.0000 (+)$	0.0476 $\pm 0.0022 (-)$	0.0620 ± 0.0028
	$IGDX$	$1.0669 \pm 0.0361 (+)$	$1.3061 \pm 0.0223 (+)$	0.2138 $\pm 0.0059 (-)$	0.2387 ± 0.0092

that RM-MEDA slightly outperforms MMEA in terms of all the performance metrics on $F7$. One cannot tell, however, any big difference in approximation quality between RM-MEDA and MMEA from Fig. 10. The reason why MMEA could not outperform RM-MEDA on $F7$ might be that neither of these two algorithms converges and has not yet started refining their solutions within the given number of function evaluations.

KP1 and Omni-Optimizer promote the population diversity in their selection operators and mainly use the SBX, which was originally proposed for single objective optimization,

to generate new solutions. RM-MEDA assumes that the PS dimensionality is $m - 1$. MMEA estimates the dimensionality and the 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 their 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 an MOP with

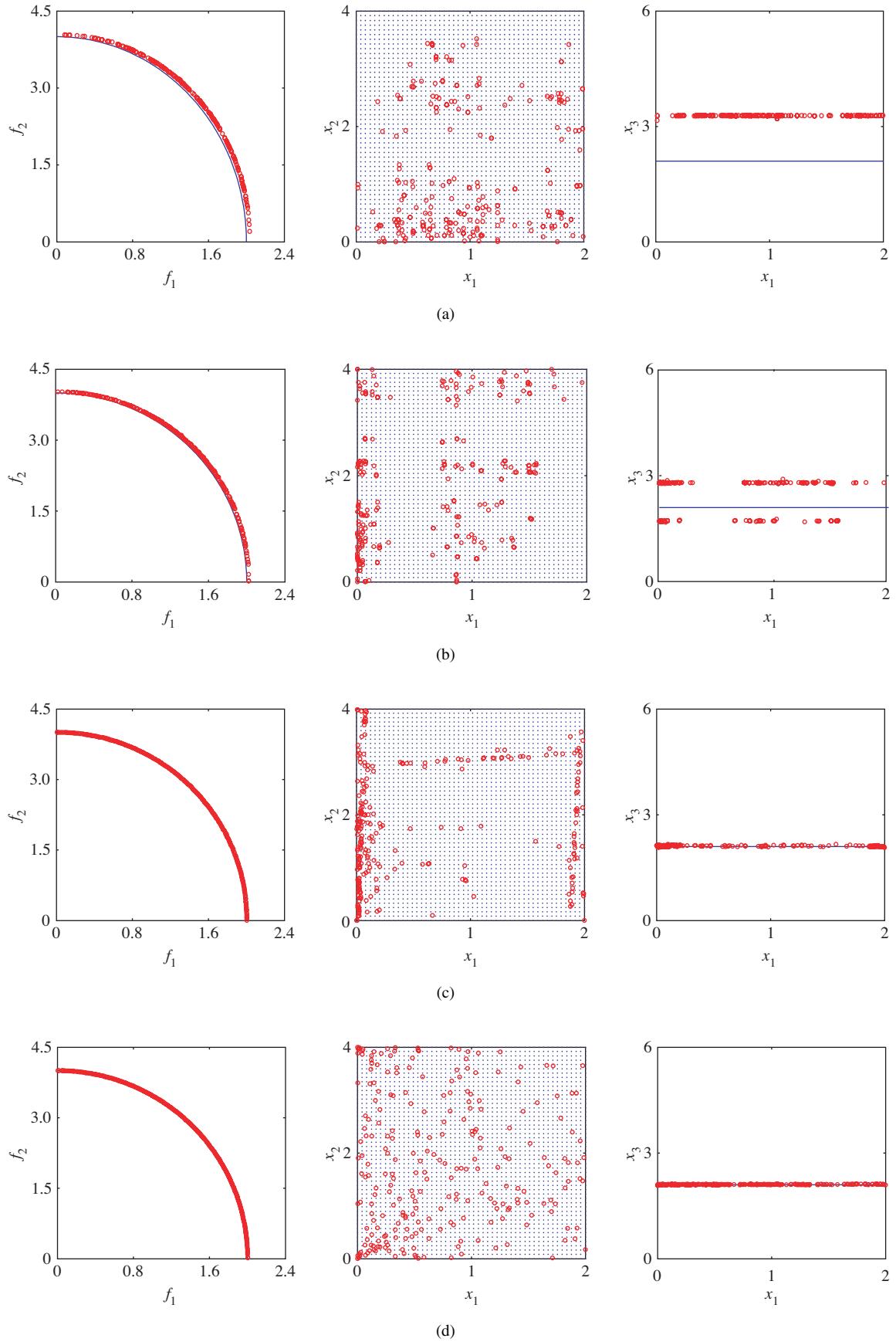


Fig. 4. Best approximations obtained by four algorithms for F1: (a) Omni-Optimizer, (b) KP1, (c) RM-MEDA, and (d) MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest *IGDF* values by four respective algorithms. Middle: the distributions of 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.

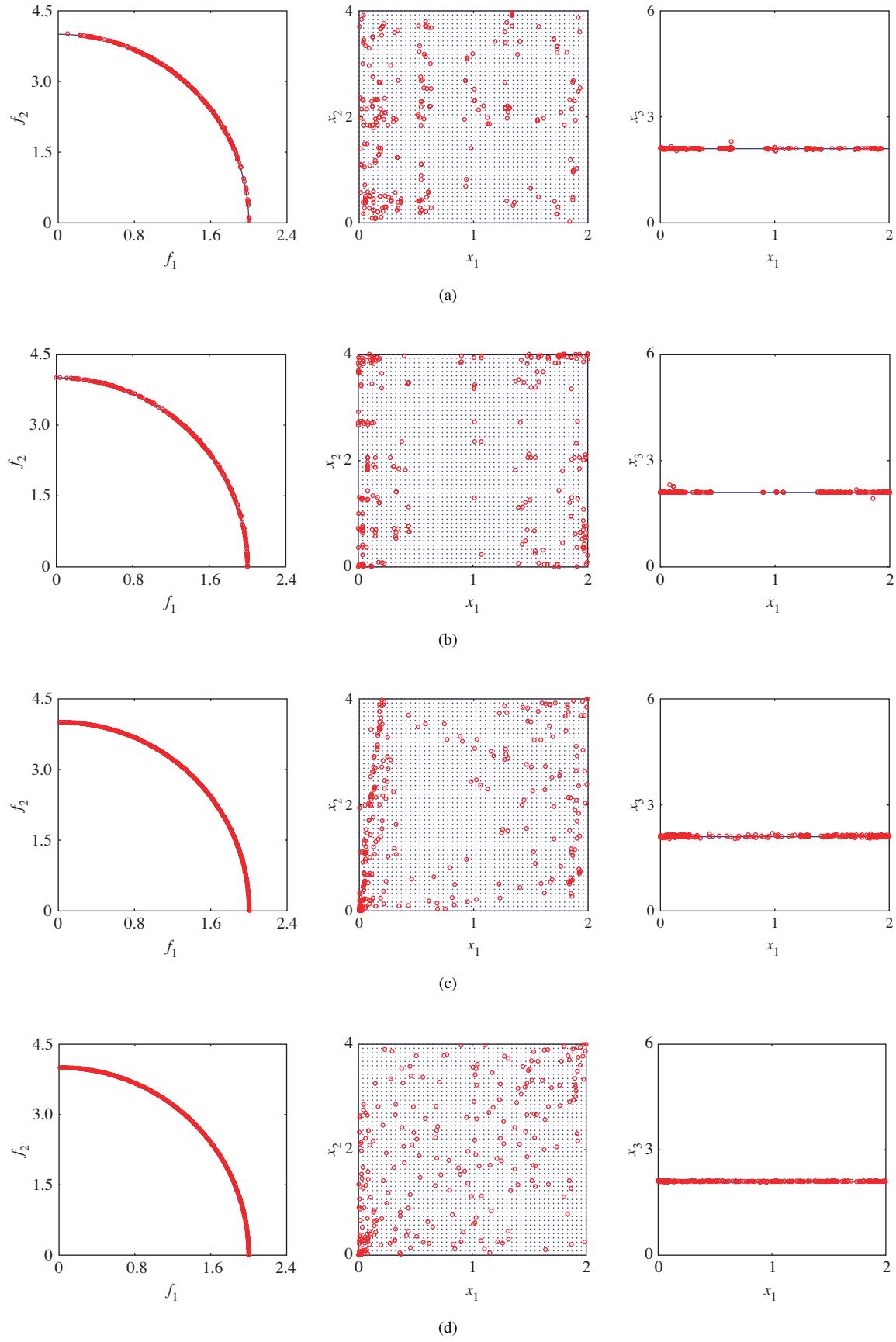


Fig. 5. Best approximations obtained by four algorithms for $F2$: (a) Omni-Optimizer, (b) KP1, (c) RM-MEDA, and (d) MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by four respective algorithms. Middle: the distributions of 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.

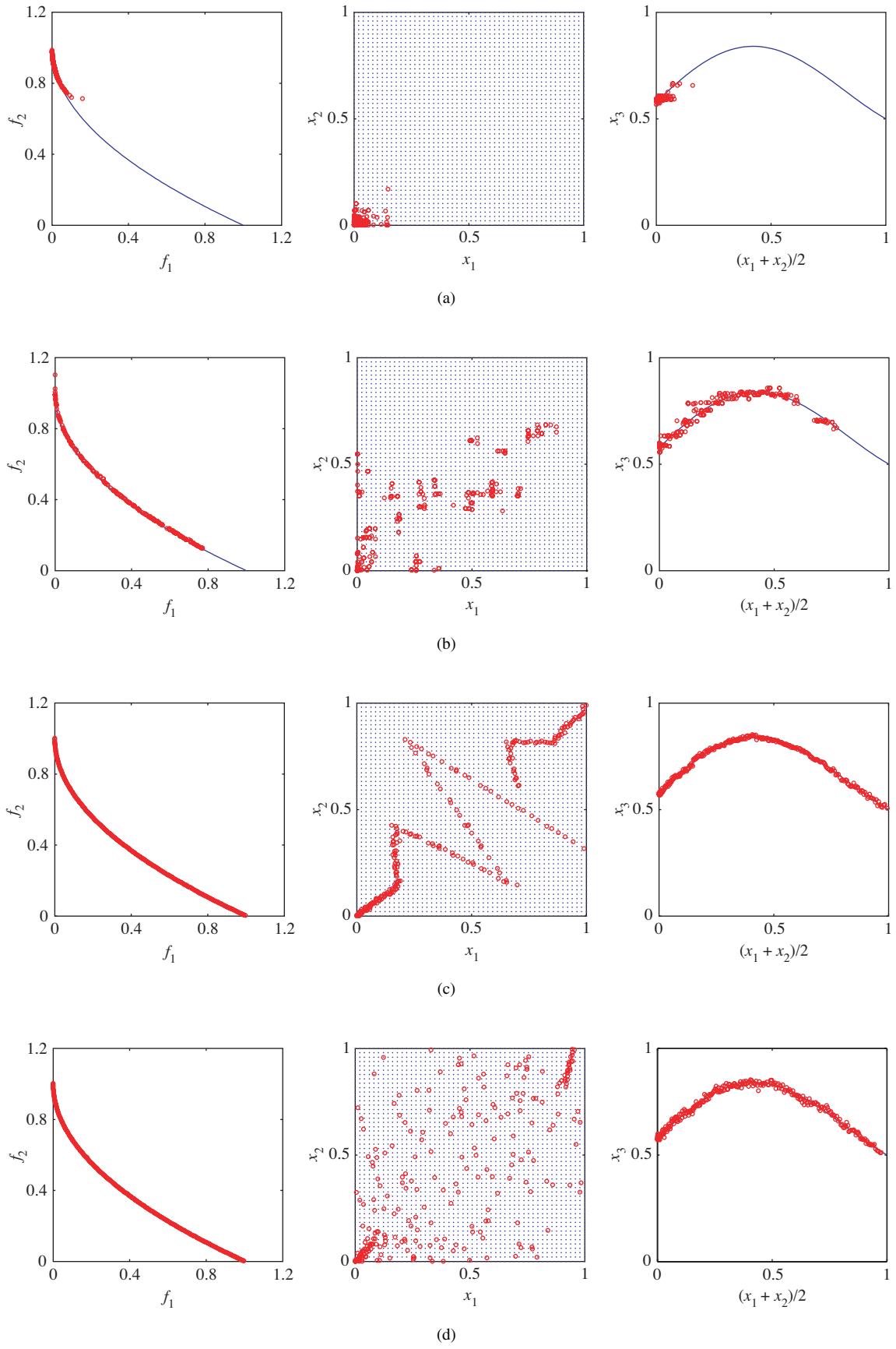


Fig. 6. Best approximations obtained by four algorithms for $F3$: (a) Omni-Optimizer, (b) KP1, (c) RM-MEDA, and (d) MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest $IGDF$ values by three respective algorithms. Middle: the distributions 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 $[(x_1 + x_2)/2] - x_3$ space obtained in the runs with the lowest $IGDX$ values.

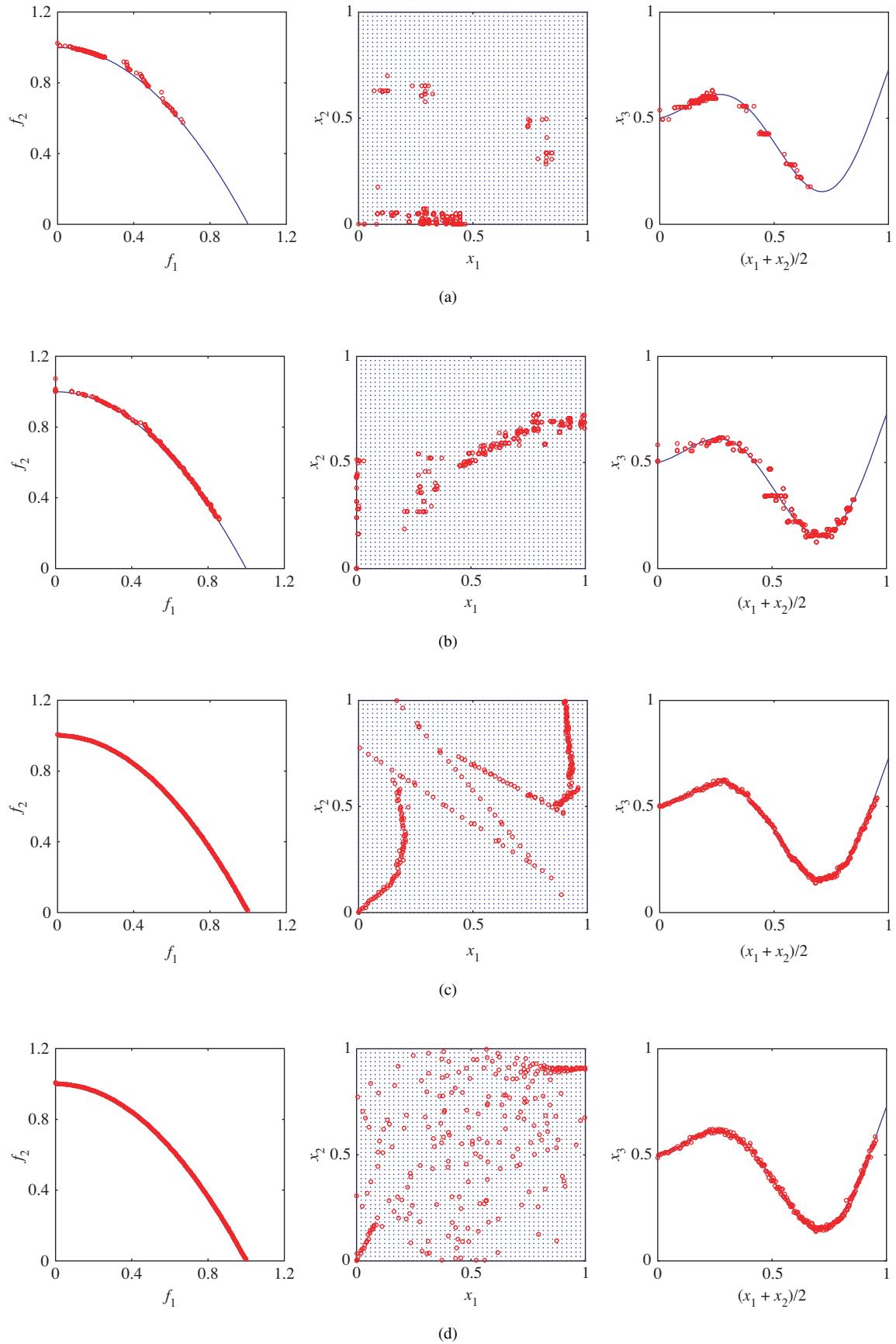


Fig. 7. Best approximations obtained by four algorithms for F4: (a) Omni-Optimizer, (b) KP1, (c) RM-MEDA, and (d) MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest *IGDF* values by three respective algorithms. Middle: the distributions of 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 $[(x_1 + x_2)/2] - x_3$ space obtained in the runs with the lowest *IGDX* values.

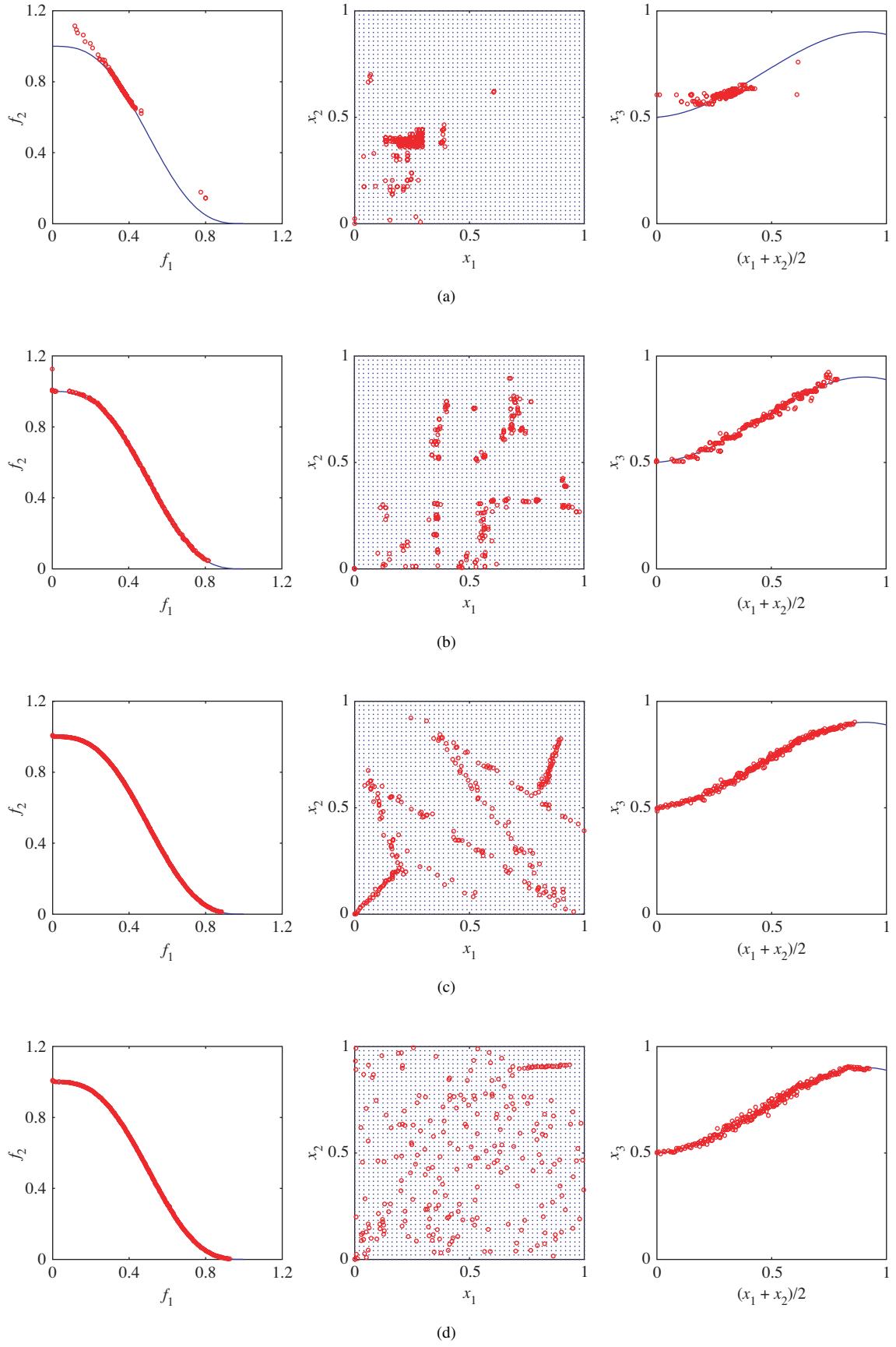


Fig. 8. Best approximations obtained by four algorithms for F5: (a) Omni-Optimizer, (b) KP1, (c) RM-MEDA, and (d) MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest *IGDF* values by three respective algorithms. Middle: the distributions of 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 $[(x_1 + x_2)/2] - x_3$ space obtained in the runs with the lowest *IGDX* values.

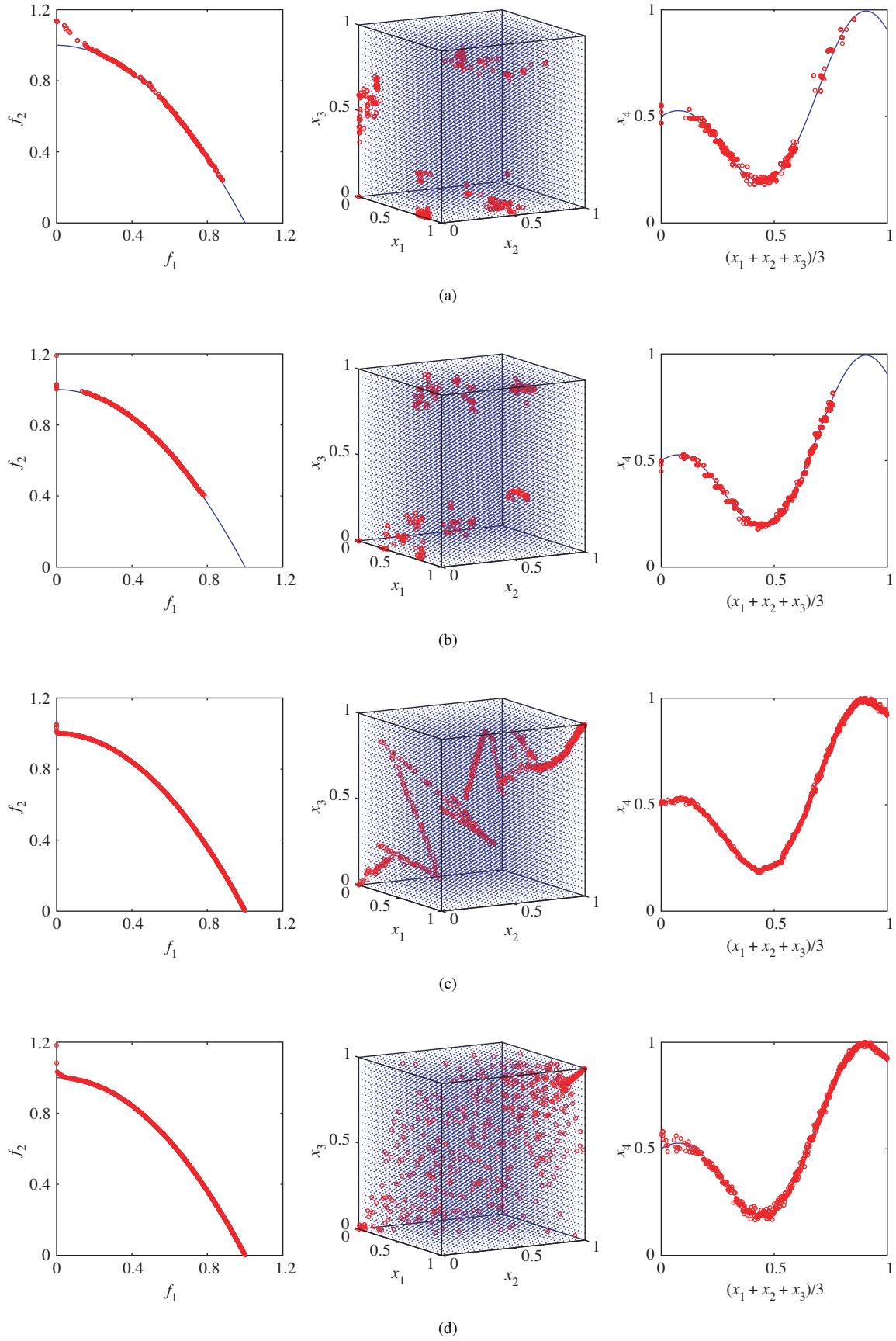


Fig. 9. Best approximations obtained by four algorithms for F6: (a) Omni-Optimizer, (b) KP1, (c) RM-MEDA, and (d) MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest *IGDF* values by three respective algorithms. Middle: the distributions of 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 $[(x_1 + x_2 + x_3)/3] - x_4$ space obtained in the runs with the lowest *IGDX* values.

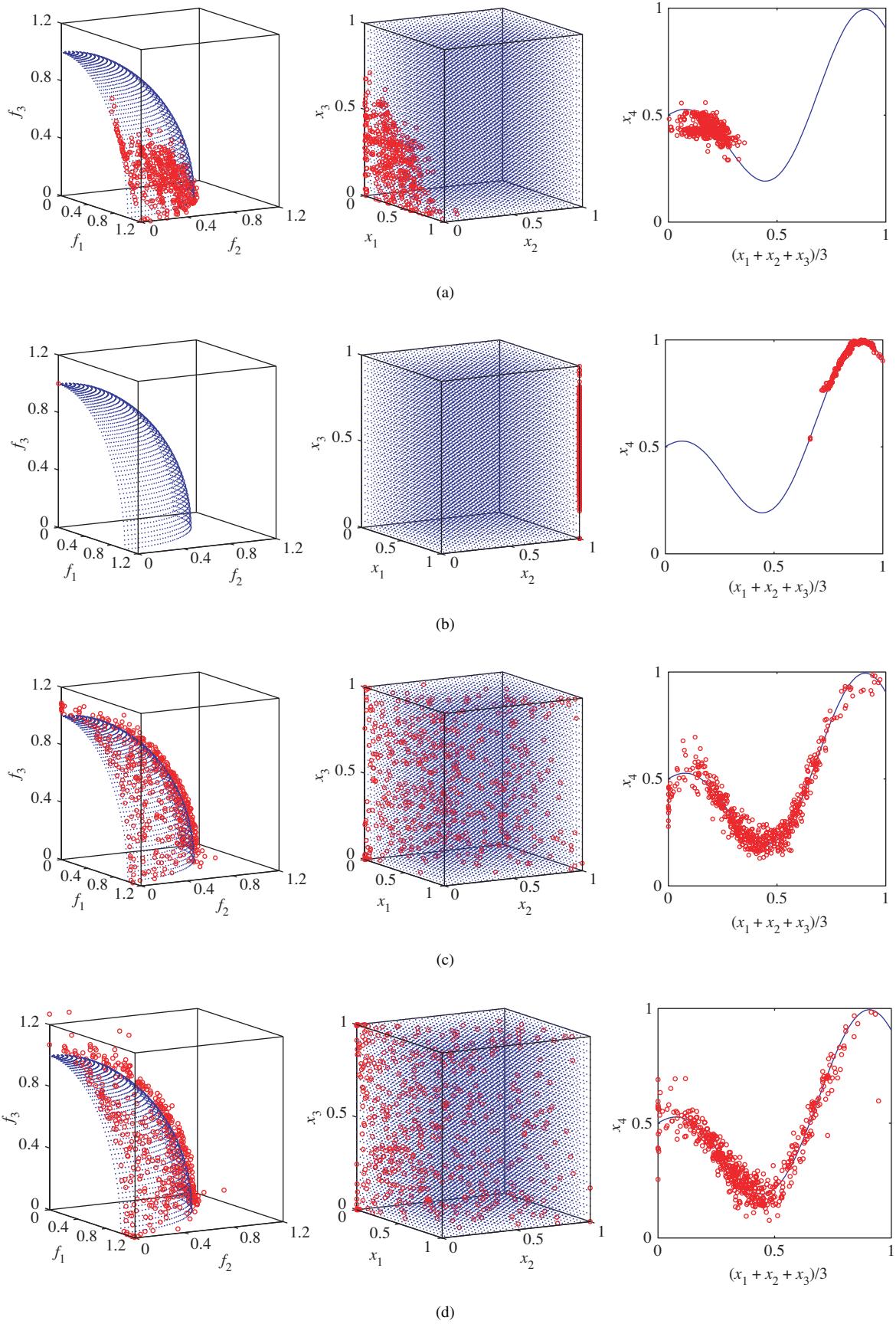


Fig. 10. Best approximations obtained by four algorithms for F7: (a) Omni-Optimizer, (b) KP1, (c) RM-MEDA, and (d) MMEA. Left: the distributions of the final solutions in the objective space obtained in the runs with the lowest *IGDF* values by three respective algorithms. Middle: the distributions of 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 $[(x_1 + x_2 + x_3)/3] - x_4$ space obtained in the runs with the lowest *IGDX* values.

TABLE IV
REGULAR TEST INSTANCES. THEIR PSS AND PFs ARE WITH THE SAME DIMENSIONALITY

Instance	Variables	Objectives	Remarks
ZZJ08-F1	$[0, 1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^n (x_i - x_1)^2)/(n - 1)$	Convex PF Linear variable linkage
ZZJ08-F2	$[0, 1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^n (x_i - x_1)^2)/(n - 1)$	Concave PF Linear variable linkage
ZZJ08-F3	$[0, 1]^n$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9[\sum_{i=2}^n (x_i - x_0)^2/9]^{0.25}$	Concave PF Nonuniformly distributed Linear variable linkage
ZZJ08-F4	$[0, 1]^n$	$f_1(x) = \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)(1 + g(x))$ $f_2(x) = \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2)(1 + g(x))$ $f_3(x) = \sin(\frac{\pi}{2}x_1)(1 + g(x))$ $g(x) = \sum_{i=3}^n (x_i - x_1)^2$	Concave PF Linear variable linkage Three objectives
ZZJ08-F5	$[0, 1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{x_1/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^n (x_i^2 - x_1)^2)/(n - 1)$	Convex PF Nonlinear variable linkage
ZZJ08-F6	$[0, 1]^n$	$f_1(x) = \sqrt{x_1}$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^n (x_i^2 - x_1)^2)/(n - 1)$	Concave PF Nonlinear variable linkage
ZZJ08-F7	$[0, 1]^n$	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9[\sum_{i=2}^n (x_i^2 - x_0)^2/9]^{0.25}$	Concave PF Nonuniformly distributed Nonlinear variable linkage
ZZJ08-F8	$[0, 1]^n$	$f_1(x) = \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)(1 + g(x))$ $f_2(x) = \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2)(1 + g(x))$ $f_3(x) = \sin(\frac{\pi}{2}x_1)(1 + g(x))$ $g(x) = \sum_{i=3}^n (x_i^2 - x_1)^2$	Concave PF Nonlinear variable linkage Three objectives

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. Since RM-MEDA tries to use an $(m - 1)$ -D manifold to approximate the PS, the final solutions obtained by it for biobjective test instances, as shown in Figs. 6–9, are distributed along 1-D manifolds in the decision space.

V. MORE DISCUSSIONS

A. Can MMEA Solve Regular MOPs?

MMEA was designed for solving an MOP of Class II, in which the dimensionality of its PS is not lower than the number of the objectives and unknown. In fact, MMEA uses the PCA technique to estimate the PS dimensionality before modeling the PS. Now a question arises whether MMEA can effectively solve a regular MOP in which the PS is an $(m - 1)$ -D continuous manifold in the decision space. To address this issue, we have compared MMEA with RM-MEDA on a set of regular MOP test instances with linear

and nonlinear variable linkages introduced in [14], which are given in Table IV.

The experimental setting are the same as in [14]:

- 1) the number of variables $n = 30$;
- 2) the population size for each algorithm $N = 100$ for two objective instances, and 200 for three objective instances;
- 3) the number of generations is 100 for ZZJ08-F1, ZZJ08-F2, ZZJ08-F5, and ZZJ08-F6; 1000 for ZZJ08-F3 and ZZJ08-F7; and 200 for ZZJ08-F4 and ZZJ08-F8;
- 4) the number of clusters in RM-MEDA $K = 5$;
- 5) α , β , θ and K_{\max} in MMEA are the same as in Table II.

All the following results are based on 20 independent runs of each algorithm on each test instance. In our experiments, 1000 points, in which x_1 taking 1000 equidistant values from their lower bounds to their upper bounds, are selected from the respective PFs of ZZJ08-F1 – ZZJ08-F3 and ZZJ08-F5 – ZZJ08-F7 to form P^* . $50 \times 50 = 2500$ points in the PF of ZZJ08-F4 and ZZJ08-F8 with $x_1, x_2 =$

TABLE V

STATISTICAL RESULTS ON REGULAR MOPs (*mean* \pm *std.*). “+” IN PARENTHESIS: THE ONE-SIDE T-TEST INDICATES THAT THE METRIC VALUE IN RM-MEDA IS LARGER THAN THAT OBTAINED BY MMEA AT THE 95% SIGNIFICANCE LEVEL. “–” IN PARENTHESIS MEANS A FAILURE IN THE T-TEST TO REJECT THE NULL HYPOTHESIS

Instance	<i>IGDF</i>		<i>IH</i> [–]	
	RM-MEDA	MMEA	RM-MEDA	MMEA
ZZJ08-F1	0.0049 \pm 0.0001 (+)	0.0044 \pm 0.0003	0.0116 \pm 0.0011 (+)	0.0070 \pm 0.0017
ZZJ08-F2	0.0056 \pm 0.0002 (+)	0.0044 \pm 0.0004	0.0190 \pm 0.0029 (+)	0.0084 \pm 0.0032
ZZJ08-F3	0.0094 \pm 0.0037 (+)	0.0047 \pm 0.0005	0.0382 \pm 0.0593 (+)	0.0078 \pm 0.0011
ZZJ08-F4	0.0723 \pm 0.0036 (+)	0.0527 \pm 0.0023	0.1596 \pm 0.0113 (+)	0.0803 \pm 0.0054
ZZJ08-F5	0.0079 \pm 0.0036 (–)	0.0076 \pm 0.0011	0.0234 \pm 0.0130 (+)	0.0135 \pm 0.0026
ZZJ08-F6	0.0136 \pm 0.0183 (–)	0.0093 \pm 0.0034	0.0644 \pm 0.0708 (+)	0.0271 \pm 0.0232
ZZJ08-F7	0.0981 \pm 0.0137 (+)	0.0582 \pm 0.0079	0.1019 \pm 0.0138 (+)	0.0774 \pm 0.0073
ZZJ08-F8	0.0694 \pm 0.0029 (–)	0.0684 \pm 0.0280	0.1211 \pm 0.0096 (+)	0.1026 \pm 0.0188

0/49, 1/49, $S, \dots, 49/49$, are taken to form P^* for experiments on ZZJ08-F4 and ZZJ08-F8. The bounded reference point is chosen to be $(1.5, 1.5)^T$ for F1–F3 and F5–F7, and $(1.5, 1.5, 1.5)^T$ for F4 and F8 in calculating I_H^- values.

Table V gives the means and standard deviations of the I_H^- and *IGDF* values of the 20 final populations obtained by RM-MEDA and MMEA for all the eight test instances. Figs. 11 and 12 show, in the objective space, the distribution of the final solutions obtained in the runs with the lowest *IGDF* values of the two algorithms for these test instances, respectively.

It is clear from Table V that MMEA is significantly better than RM-MEDA on all the instances in terms of the I_H^- metric, and MMEA significantly outperforms or is not worse than RM-MEDA in terms of *IGDF* metric on these eight instances. The difference between the best approximations of RM-MEDA and those of MMEA in Figs. 11 and 12 can hardly be visually distinguished. These results imply that although MMEA is designed for Class II problems, its performance is not worse than RM-MEDA on regular MOPs.

B. Can MMEA Deal With an MOP of Class I?

An MOP of Class II has a continuous PS of dimensionality larger than $m - 1$, while the PS of an 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 tested MMEA on DT05-F4.4 [25], in which the two objectives to be minimized are as follows:

$$f_1(x) = \sum_{i=1}^n \sin(\pi x_i)$$

$$f_2(x) = \sum_{i=1}^n \cos(\pi x_i)$$

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

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

and its PS consists of 3^n disconnected parts, each of which 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 1000 as in [25]. All the other parameter settings are the same as in Section IV. Fig. 13 presents the final population obtained in the run with the lowest *IGDF* value among 20 independent runs. Clearly, MMEA has not produced a satisfactory approximation to the PS. This could be attributed to the fact that population clustering in MMEA is based on the distance in the objective space, which prevents it from distinguishing the different parts of the DT05-F4.4 PS in the decision space and thus cannot find a good approximation to its PS.

C. Sensitivity of Control Parameters

In the following, taking F3 as an example, we investigate the sensitivity of the four control parameters in MMEA.

1) *The Effect of α :* We have tried different values of α : 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0 for MMEA on F3. The settings of the other control parameters are the same as in Section IV. Fig. 14 shows the average *IGDF* and *IGDX* metrics versus the different values of α , respectively. It is clear that MMEA works very well when $0.4 \leq \alpha \leq 2.0$. It also indicates that MMEA performs poorly if α is too small. This could be due to the fact that a small value of α might reduce the exploration ability of MMEA.

2) *The Effect of β :* We have tried different values of β : 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, and 2.0 for MMEA on F3. The settings of the other control parameters are the same as in Section IV. Fig. 15 shows the average *IGDF* and *IGDX* metrics v.s. the different values of β , respectively. Clearly, *IGDX* value is more sensitive to β than *IGDF*. This is not a surprise since β is mainly for extending the search in the decision space. It is also evident that MMEA works well in terms of both *IGDX* and *IGDF* metrics if $0.8 \leq \beta \leq 2$.

3) *The Effect of θ :* We have tried different values of θ : 0.5, 0.6, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, and 1.0 on F3. The settings of the other parameters are the same as in Section IV.

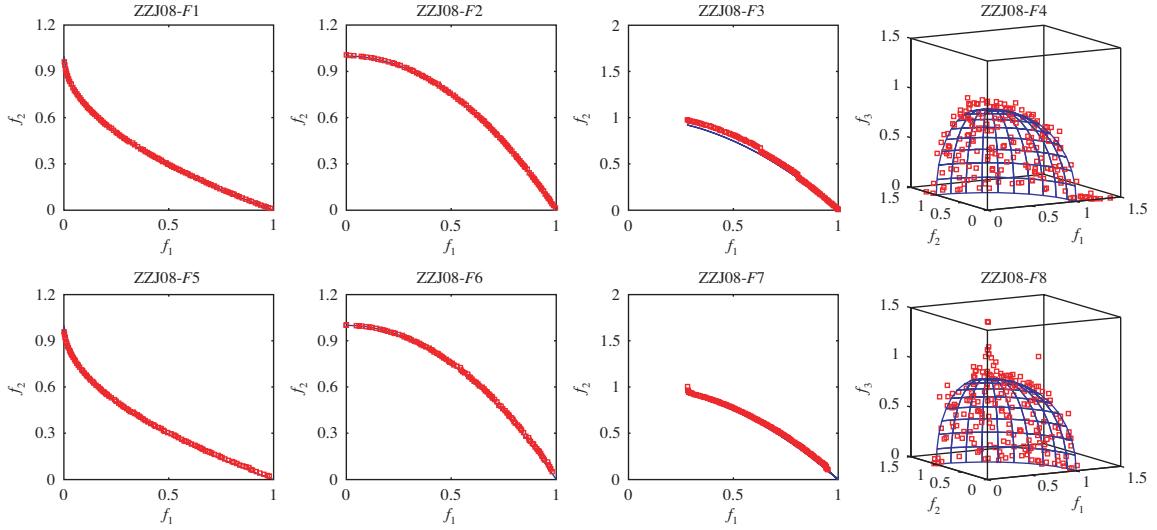


Fig. 11. Approximations with the lowest *IGDF* values obtained by RM-MEDA on regular MOPs.

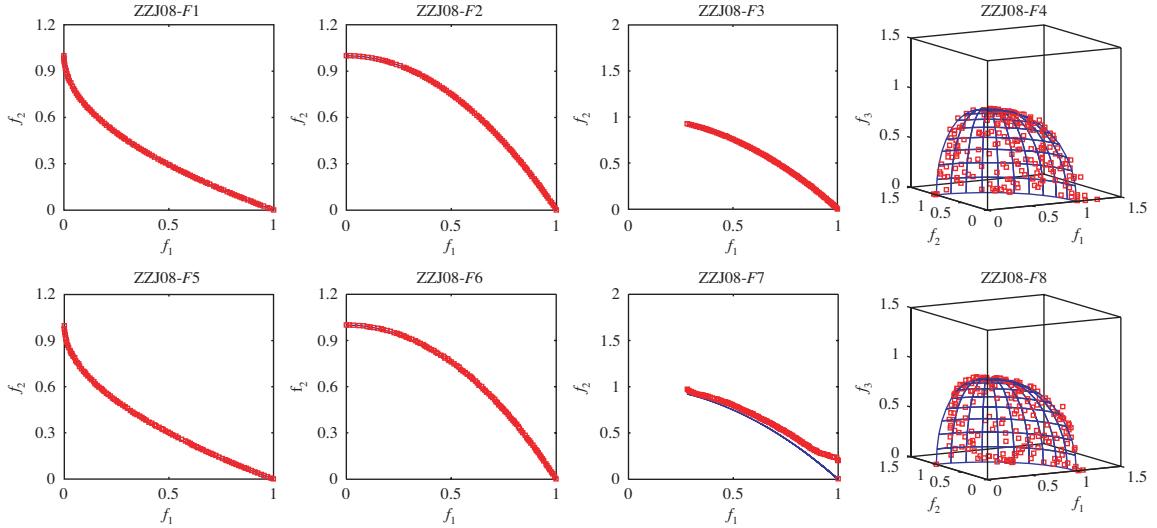


Fig. 12. Approximations with the lowest *IGDF* values obtained by MMEA on regular MOPs.

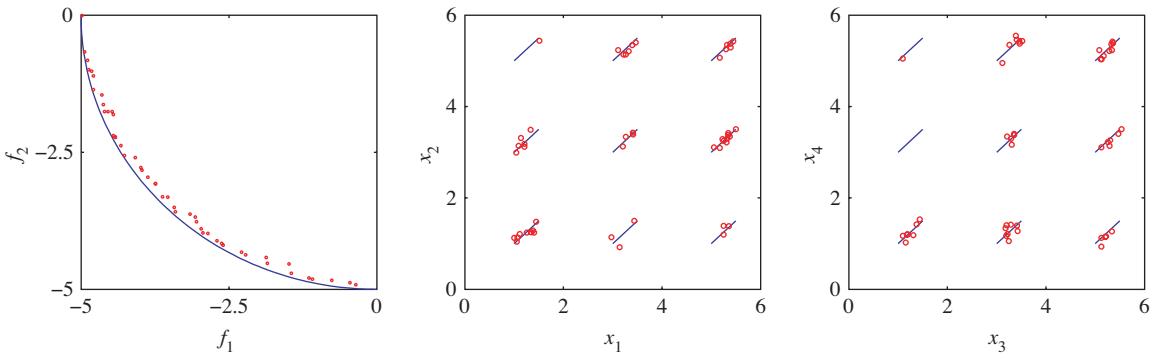


Fig. 13. Best approximation obtained in the 20 runs with the lowest *IGDF* value by MMEA for DT05-F4.4. Left: the distribution of the final solutions in the objective space. Middle: the distribution of the final solutions in the $x_1 - x_2$ space. Right: the distribution of the final solutions in the $x_3 - x_4$ space.

Fig. 16 shows the average *IGDF* and *IGDX* metrics versus the different values of θ , respectively. It is clear from this figure that MMEA can approximate the PF of *F3* if $0.5 \leq \theta \leq 0.95$ and it can approximate the PS of *F3* if $0.75 \leq \theta \leq 0.95$.

When $\theta = 1.0$, the performance of MMEA becomes very poor. The reason might be that in such a case, MMEA implicitly assumes that the dimensionality of the PS is n and set σ^k to be zero in **Step 5.6** in PCA and Modeling;

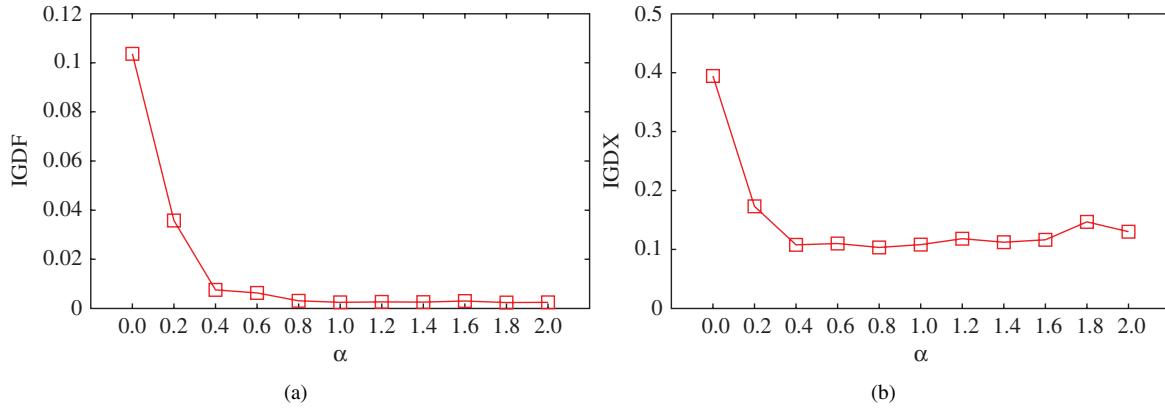


Fig. 14. Average IGD metrics of the 20 final approximations versus α in MMEA on F_3 .

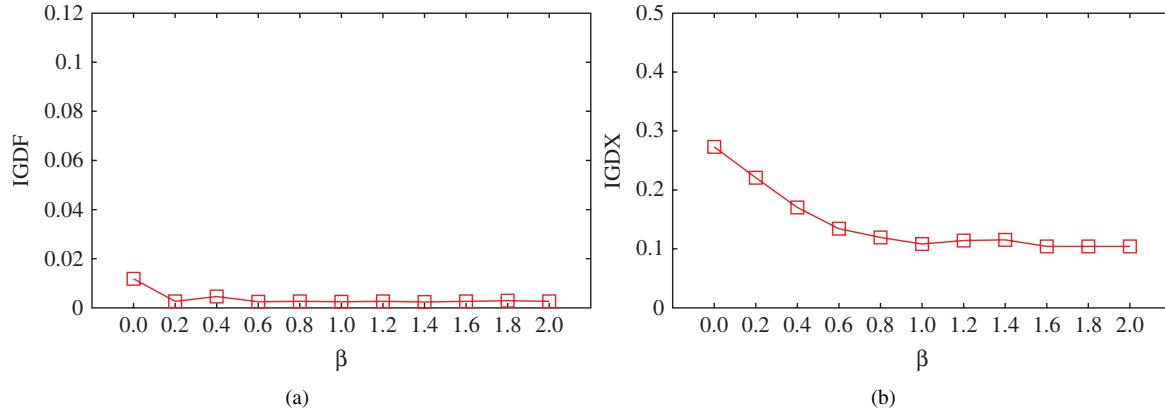


Fig. 15. Average IGD metrics of the 20 final approximations versus β in MMEA on F_3 .

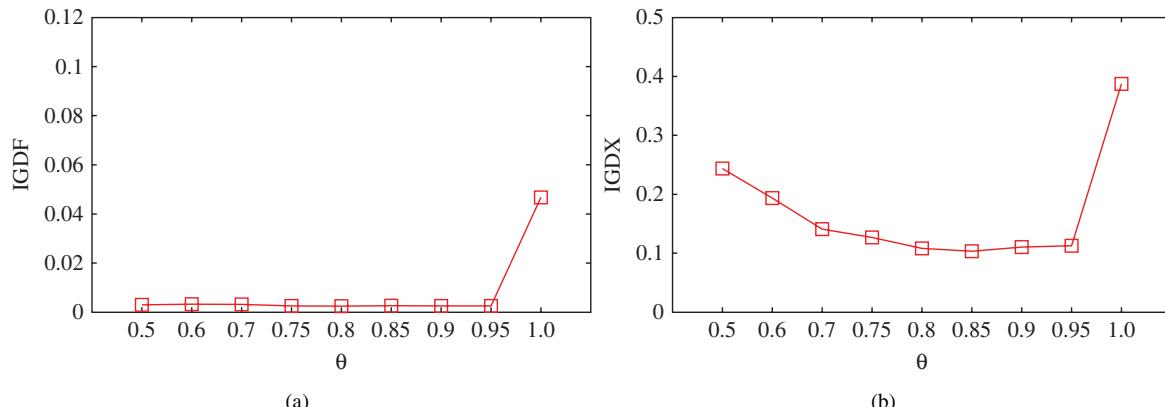


Fig. 16. Average IGD metrics of the 20 final approximations versus θ in MMEA on F_3 .

as a result, the search ability of MMEA has been reduced significantly.

4) The Effect of K_{\max} : In our experiment, we have tried the different values of K_{\max} : 10, 15, 20, 25, 30, 35, 40, 45, and 50. It is evident from Fig. 17 that MMEA could approximate the PF of F_3 very well when $K_{\max} \geq 15$. However, when $K_{\max} > 35$, the approximation quality in the decision space will decrease. It implies that with too many subpopulations, MMEA could not correctly estimate the shape of the PS.

From the above experiments, we can also conclude that for each of these four control parameters, there is a reasonably large range such that MMEA works well.

D. CPU Time Cost

We have also recorded the CPU time used by each algorithm on $F_1 - F_7$. The average CPU time³ used by the four

³The four algorithms are implemented in C++, and they are executed in Thinkpad X60s with Intel Core Duo CPU L2400 1.66GHz, 2GB RAM, and Windows Vista.

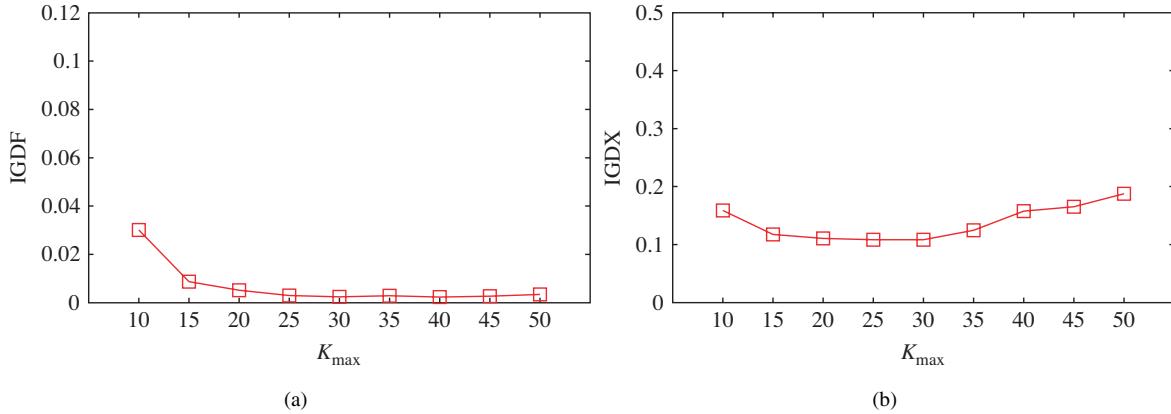
Fig. 17. Average IGD metrics of the 20 final approximations versus K_{\max} in MMEA on F_3 .

TABLE VI
THE AVERAGE CPU TIME (IN SECONDS) USED BY FOUR ALGORITHMS ON F_1 – F_7

	F_1	F_2	F_3	F_4	F_5	F_6	F_7
KP1	33.88	50.48	18.76	16.43	21.45	62.81	153.37
Omni-Optimizer	113.21	134.33	101.56	79.06	86.64	311.14	458.74
RM-MEDA	1339.91	1168.92	973.18	952.44	904.29	2743.44	2675.40
MMEA	136.18	165.22	128.66	125.10	120.80	413.86	470.12

algorithms are given in Table VI. Clearly, RM-MEDA needs much more CPU time than the three others. KP1 is the fastest in terms of CPU times. MMEA and Omni-Optimizer require about the same CPU time. The reason why MMEA is faster than RM-MEDA might be that the PCA used in MMEA requires much less CPU time than the local PCA used in RM-MEDA.

VI. CONCLUSION

A good approximation to both the PS and the PF of an MOP might be required in some real-world applications. A good approximation to the PF of an MOP might not represent 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 an MOP of Class I. This paper represents a first attempt to do so for an MOP of Class II.

MMEA proposed in this paper generalizes the idea used in RM-MEDA to an MOP of Class II for approximating its PS and PF simultaneously. In the modeling phase of MMEA, the population is clustered into a number of subpopulations based on their distribution in the objective space, the PCA technique is then used to estimate the dimensionality of the PS manifold in each subpopulation, and then a probabilistic model is built for modeling the distribution of promising solutions in the decision space. We argue that such a modeling procedure could promote the population diversity in both the decision and objective spaces. New solutions are sampled from the model thus built. The population for the next generation is selected by the NDS selection. The comparison between MMEA and the three other algorithms, KP1, Omni-Optimizer, and RM-MEDA, on seven test instances, five of which were proposed

in this paper, has been made in this paper. Our empirical results have clearly indicated that MMEA has a big advantage over the three other algorithms in approximating both the PS and the PF of an MOP of Class II. We have investigated the ability of MMEA to deal with a regular MOP and an MOP of Class I. We have also studied the sensitivity of control parameters in MMEA.

The future research topics along this line may include:

- 1) extension of MMEA to constrained MOPs, and MOPs under dynamic and/or noisy environment for approximating both their PS and PF [39]–[41];
- 2) study of the scalability of MMEA to the numbers of decision variables and objectives [42]–[44];
- 3) use of other machine learning methods in MMEA [45]–[47];
- 4) combination of other techniques, particularly, traditional mathematical programming methods and new ideas in MOEAs, with MMEA for improving the algorithm performance.

The C++ code of MMEA can be downloaded from Q. Zhang's homepage: <http://dces.essex.ac.uk/staff/qzhang/>.

REFERENCES

- [1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA: Kluwer Academic, 1999.
- [2] K. Deb, *Multiobjective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
- [3] C. A. Coello Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multiobjective Problems*. Norwell, MA: Kluwer, 2002.
- [4] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for Pareto multiobjective optimisation," in *Proc. Congr. Evol. Comput.*, vol. 1. 1999, pp. 98–105.

- [5] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, 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 Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, Jun. 2003.
- [7] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.
- [8] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature (PPSN VIII)*, LNCS vol. 3242, Birmingham, U.K.: Springer-Verlag, 2004, pp. 832–842.
- [9] C. Gil, A. Márquez, R. Banos, M. G. Montoya, and J. Gómez, "A hybrid method for solving multiobjective global optimization problems," *J. 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 Trans. Evol. Comput.*, vol. 10, no. 5, pp. 527–549, Oct. 2006.
- [11] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multiobjective optimization," *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, 2007.
- [12] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.
- [13] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [14] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 41–63, Feb. 2008.
- [15] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 269–283, Jun. 2008.
- [16] K. I. Smith, R. M. Everson, J. E. Fieldsend, C. Murphy, and R. Misra, "Dominance-based multiobjective simulated annealing," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 323–342, 2008.
- [17] H. P. Benson and S. Sayin, "Optimization over the efficient set: Four special cases," *J. Optimization Theory Applicat.*, vol. 80, no. 1, pp. 3–18, 1994.
- [18] P. T. Thach, H. Konno, and D. Yokota, "Dual approach to minimization on the set of pareto-optimal solutions," *J. Optimization Theory Applicat.*, vol. 88, no. 3, pp. 689–707, 1996.
- [19] R. Horst and N. V. Thoai, "Utility function programs and optimization over the efficient set in multiple-objective decision making," *J. Optim. Theory Appl.*, vol. 92, no. 3, pp. 605–631, 1997.
- [20] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*. New York: Springer-Verlag, 2008.
- [21] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, 2000.
- [22] Y. Jin. (2002). *"Effectiveness of weighted sum of the objectives for evolutionary multiobjective optimization: Methods, analysis and applications."* [Online]. Available: <http://www.soft-computing.de/edwa2002.pdf>
- [23] G. Rudolph, B. Naujoks, and M. Preuss, "Capabilities of EMOA to detect and preserve equivalent Pareto subsets," in *Proc. 4th Int. Conf. Evol. Multicriterion Optimization (EMO '07)*, 2007, LNCS vol. 4403, pp. 36–50.
- [24] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.
- [25] K. Deb and S. Tiwari, "Omni-Optimizer: A procedure for single and multiobjective optimization," in *Proc. 3rd Int. Conf. Evol. Multicriterion Optimization (EMO '05)*, 2005, LNCS vol. 3410, pp. 41–65.
- [26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [27] K. P. Chan and T. Ray, "An evolutionary algorithm to maintain diversity in the parametric and the objective space," in *Proc. 3rd Int. Conf. Comput. Intell., Robotics Autonomous Syst. (CIRAS '05)*.
- [28] M. Preuss, B. Naujoks, and G. Rudolph, "Pareto set and EMOA behavior for simple multimodal multiobjective functions," in *Parallel Problem Solving from Nature (PPSN IX)*, LNCS vol. 4193, Reykjavík, Iceland: Springer-Verlag, 2006, pp. 513–522.
- [29] L. Devroye, *Non-Uniform Random Variate Generation*. New York: Springer-Verlag, ch. XI, 1986, pp. 568–570.
- [30] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural Comput.*, vol. 9, no. 7, pp. 1493–1516, 1997.
- [31] M. Reyes Sierra and C. A. Coello Coello, "A study of fitness inheritance and approximation techniques for multiobjective particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC '05)*, 2005, pp. 65–72.
- [32] J. D. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," Comput. Eng. Networks Laboratory, ETH Zurich, Zurich, Switzerland, Tech. Rep. 214, 2006.
- [33] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms a comparative case study," in *Parallel Problem Solving from Nature (PPSN V)*, LNCS vol. 1498, Amsterdam, The Netherlands: Springer-Verlag, 1998, pp. 292–304.
- [34] J. D. Knowles and D. W. Corne, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 100–116, Apr. 2003.
- [35] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evol. Methods Design, Optimisation Control*, Barcelona, Spain: CIMNE, 2002, pp. 95–100.
- [36] H. Li and Q. Zhang, "Comparison between NSGA-II and MOEA/D on a set of multiobjective optimization problems with complicated Pareto sets," *IEEE Trans. Evol. Comput.*, vol. 12, no. 2, pp. 284–302, 2009.
- [37] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
- [38] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inform.*, vol. 26, no. 4, pp. 30–45, 1996.
- [39] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments a survey," *IEEE Trans. Evol. Comput.*, vol. 9, no. 3, pp. 303–317, Jun. 2005.
- [40] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multiobjective optimization," in *Proc. 4th Int. Conf. Evol. Multicriterion Optimization (EMO '07)*, 2007, LNCS vol. 4403, pp. 832–846.
- [41] C. K. Goh and K. C. Tan, "An investigation on noisy environments in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 3, pp. 354–381, Jun. 2007.
- [42] C. W. Ahn and R. S. Ramakrishna, "On the scalability of real-coded bayesian optimization algorithm," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 307–322, Jun. 2008.
- [43] L. Martí, J. García, A. Berlanga, and J. M. Molina, "Scalable continuous multiobjective optimization with a neural network based estimation of distribution algorithm," in *Proc. Applicat. Evol. Comput.: EvoWorkshops 2008*, LNCS vol. 4974, pp. 535–544.
- [44] Y. Jin, A. Zhou, Q. Zhang, B. Sendhoff, and E. Tsang, "Modeling regularity to improve scalability of model-based multiobjective optimization algorithms," in *Multiojective Problem Solving*, New York: Springer-Verlag, 2008, pp. 331–355.
- [45] R. S. Michalski, "Learnable evolution model: Evolutionary processes guided by machine learning," *Mach. Learn.*, vol. 38, no. 1–2, pp. 9–40, 2000.
- [46] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Monte Carlo Simulation, Randomized Optimization and Machine Learning*, New York: Springer-Verlag, 2004.
- [47] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, *Toward a New Evolutionary Computation: Advances in Estimation of Distribution Algorithms*, Secaucus, NJ: Springer-Verlag, 2006.



Aimin Zhou received the B.Sc. and M.Sc. degrees in computer science from Wuhan University, Wuhan, China, in 2001 and 2003, respectively, and the Ph.D. degree in computer science from the University of Essex, Colchester, U.K., in 2009.

He is currently a Lecturer in the Department of Computer Science and Technology, East China Normal University, Shanghai, China. His main research areas are evolutionary computation, multiobjective optimization, and metaheuristics.



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

He is currently a Professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K. From 1994 to 2000, he was with the National University of Defense Science and Technology, China, Hong Kong

Polytechnic University, Hong Kong, the German National Research Center for Information Technology (now Fraunhofer-Gesellschaft), Germany, and the University of Manchester Institute of Science and Technology, U.K. His main research areas are evolutionary computation, optimization, neural networks, data analysis, and their applications. He has published more than 70 research papers.

Prof. Zhang is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B, and an Editorial Board Member of three other international journals in his area of expertise. He has been a Guest Editor of three special issues of international journals, including one on evolutionary algorithms based on probabilistic models in the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION. He won the Unconstrained Multiobjective Optimization Algorithm Competition in the Congress of Evolutionary Computation in 2009 and was awarded the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award.



Yaochu Jin (M'98–SM'02) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr-Universität Bochum, Bochum, Germany, in 2001.

He is currently a Principal Scientist at Honda Research Institute Europe, Offenbach, and a Member of Scientific Coordination, CoR-Lab Graduate School, Bielefeld University, Bielefeld, both in Germany. His research interests are computational approaches to a system-level understanding of evolution,

learning and development in biology, and bio-inspired methodologies for complex systems design, spanning the research fields of computational intelligence, artificial life, computational systems biology, and computational neurosciences. He is the Co-Editor of *Fuzzy Systems in Bioinformatics and Computational Biology*, *Multiobjective Machine Learning*, and *Knowledge Incorporation in Evolutionary Computation*, the author of the book *Advanced Fuzzy Systems Design and Applications*, all published by Springer-Verlag, and the author/coauthor of many journal and conference papers.

Dr. Jin is currently an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART C, IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, AND BIOSYSTEMS, and an Editorial Board Member of Soft Computing and Memetic Computing. He has been a Guest (co)-Editor of six special issues of international journals, including one on “Evolutionary Optimization in the Presence of Uncertainties” in the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.