

ORIGINAL ARTICLE

A full variate Gaussian model-based RM-MEDA without clustering process

Meifeng Shi¹ · Zhongshi He¹ · Ziyu Chen¹ · Xin Liu²

Received: 20 November 2015 / Accepted: 28 March 2017
© Springer-Verlag Berlin Heidelberg 2017

Abstract A regularity model-based multi-objective estimation of distribution algorithm (RM-MEDA) is an excellent multi-objective estimation of distribution algorithm proposed in recent years. However, the performance of RM-MEDA is seriously affected by its clustering process. In order to avoid the influence of the clustering process, this paper presents a novel full variate Gaussian model-based (FGM-based) RM-MEDA without clustering process, named FRM-MEDA. In FRM-MEDA, the clustering process is removed from the original algorithm and the full variate Gaussian model (FGM) is introduced to keep the population diversity and make up the loss of the performance caused by removing the clustering process. Meanwhile, the introduction of FGM makes the FRM-MEDA faster and more stable when solving all the test instances. In addition, variable variance of FGM is presented to enhance the exploring ability of FRM-MEDA. The experiments demonstrate that the proposed algorithm significantly outperforms the RM-MEDA without clustering process and the RM-MEDA with K equal to AVE_K .

Keywords Estimation of distribution algorithm · Multi-objective optimization · Number of clusters · Full variate Gaussian model

1 Introduction

Multi-objective optimization problems (MOPs) comprise several mutual effected and conflicting objectives that should be satisfied simultaneously. Hence, no single solution can optimize all the objectives at the same time. Without loss of generality, all objective instances are assumed to be minimized, and then a general form of MOPs is given as

$$\begin{aligned} \min_x q &= (f_1(x), f_2(x), \dots, f_m(x)) \\ \text{subject to } &\left\{ \begin{array}{l} x \in X \subseteq R^n \\ q \in Q \subseteq R^m \end{array} \right. \end{aligned} \quad (1)$$

where R^n is the decision space, R^m is the objective space and $x = (x_1, x_2, \dots, x_n)^T$ is the decision variable vector. The goal of a multi-objective optimization algorithm is to present a solution set to make all the objectives as optimal as possible synchronously. MOPs have become a hotspot in the research field and are developed rapidly to solve many real-life applications [1–6].

Wide varieties of algorithms have been investigated to solve MOPs, where Estimation of Distribution Algorithms (EDAs) is one kind of those algorithms. EDAs are originally formed in 1996 and developed rapidly [7–13]. Up to now, EDAs become a hotspot in the research field of evolutionary computation and have been used to solve many engineering MOPs [7]. EDAs provide a novel macroscopically evolutionary paradigm in which an explicit probabilistic distribution model is built to describe the movements

✉ Meifeng Shi
shishating@163.com

Zhongshi He
zshe@cqu.edu.cn

Ziyu Chen
chenziyu@cqu.edu.cn

Xin Liu
liuxin_324@163.com

¹ College of Computer Science, Chongqing University, Chongqing, China

² School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing, China

of population and the population then evolves by iteratively learning and sampling the built model. EDAs have no evolutionary operators like crossover and mutation during the evolution. Generally, EDAs have the following advantages. Firstly, EDAs perform very well for problems with continuous and discrete decision variables. Then, they may easily incorporate prior information about the location of the optima. Finally, EDAs are easier to be analyzed theoretically. Therefore, in principle, the reasons for good or bad performance on a problem can be determined more easily, in comparison with other evolutionary algorithms. However, as a matter of fact there are still some drawbacks in EDAs. One of the biggest disadvantages of EDAs is that the more complex the problem to be solved is, the more expensive the probabilistic model used in EDAs is. Secondly, sometimes it is difficult to learn an adequate probabilistic model and in some cases it is possible to create a totally incorrect model for the problems. Thirdly, the implementation of an EDA is not straightforward. For these drawbacks of EDAs, many modified and new EDA algorithms are proposed in recent years, such as Boltzmann based EDA [14], copula-based EDA [15], Graph-based EDA [16], etc. Especially, the regularity model-based EDA and one of its improved versions are exactly algorithms we are interested in.

Under certain smoothness assumptions, it can be induced from the Karush–Kuhn–Tucker condition that a piecewise continuous ($m-1$)-dimensional manifold is defined by the Pareto set (PS) of a continuous MOP in its decision space [17–19]. Thus, the PS of a continuous bi-objective or tri-objectives optimization problem is a piecewise continuous curve or surface in R^n . Based on the above analysis of regularity, a regularity model-based multi-objective estimation of distribution algorithm named RM-MEDA has been proposed by Zhang et al. [20]. As a kind of EDAs [7], RM-MEDA describes the movements of populations by employing the ($m-1$)-dimensional local principal component analysis (($m-1$)-D local PCA) [21] to build the model over candidate solutions in the decision space. In RM-MEDA, the population is divided into a number of clusters to build the model. Furthermore, the number of clusters is fixed in the algorithm even when solving different kinds of problems. Nevertheless, Wang et al. found that the number of clusters K had a significant effect on the performance of RM-MEDA. There will be 3 situations when determining the value of K , that is to say, the given value is smaller, larger than, or equal to the required one. Ideally, if K is equal to the required one, the cluster C_i obtained by the model could approximate one of the pieces of the PS effectively. However, if K is not equal to the required value, what will happen? Wang et al. have given two crucial conclusions after their analysis. One conclusion is that the model built will be incorrect if the number of

clusters is smaller than required, and thus, result in a passive effect on the algorithm performance. Another one is that if the number of clusters is larger than required, there exist several incorrect or redundant manifolds though the PS approximated by the established model, which has a passive influence on both of the convergence speed and the convergence quality [16]. So they pointed out that the number of clusters was problem-dependent and cannot be fixed to a certain value. Motivated by the observation above, they improved the clustering process of RM-MEDA and present a reducing redundant cluster operator (RRCO) to establish a more precise model during the whole optimization process. By bringing RRCO into RM-MEDA, they proposed an improved version of RM-MEDA, named IRM-MEDA [22]. It is worth noting that in IRM-MEDA Wang et al. only coped with K larger than required since they usually initialized K with a higher value when solving MOPs. But how to deal with K smaller than required has not still been solved.

In this work, we try to solve the unsolved issue of IRM-MEDA by introducing the full variate Gaussian model (FGM) into the RM-MEDA in which the number of clusters is set to 1. In another word, the clustering process is removed from the RM-MEDA. Some researchers have pointed out that the inappropriate number of clusters always results in a passive influence on the performance of the algorithm, which is partly caused by the diversity loss of solutions in the case the number of clusters is smaller than required. Fortunately, the classical EDA with FGM can easily keep the diversity of solutions during the whole optimization process, which could make up the loss of the performance caused by removing the clustering process. Thus, a novel FGM-based RM-MEDA without clustering process, named FRM-MEDA, is presented in this paper.

The rest of this paper is organized as follows. Section 2 briefly reviews the RM-MEDA and its variant, IRM-MEDA. The full variate Gaussian model is presented in Sect. 3. Section 4 describes the proposed FRM-MEDA. The experimental results and analysis of FRM-MEDA in comparison with other algorithms on a set of MOPs are given in Sect. 5. Finally, Sect. 6 concludes the paper and outlines future research work.

2 Review of RM-MEDA and IRM-MEDA

Some algorithms related to this paper are introduced in this section.

2.1 RM-MEDA

In RM-MEDA, the solutions in the population are assumed as independent observations of a random vector $\xi \in R^n$

whose centroid is the Pareto set (PS) of Formula (1). As the centroid of the PS that is a piecewise continuous ($m-1$)-dimensional manifold, ξ can be naturally described as

$$\xi = \zeta + \epsilon. \quad (2)$$

where ζ is uniformly distributed over the PS, and ϵ is an n -dimensional zero mean noise vector. Figure 1 illustrates the basic idea [16].

As can be seen in Fig. 1, the population in the decision space will hopefully approximate and be uniformly scattered around the PS along with evolution. Modeling is a very important process in RM-MEDA. The modeling process of RM-MEDA comprises of two main steps. Firstly, RM-MEDA needs firstly to partition the population P_t into K clusters, C_1, \dots, C_K . Secondly, one model is built using the $(m-1)$ -D local PCA to estimate one manifold ψ_i with one cluster C_i . There are two points the RM-MEDA should maintain at each generation t :

1. A population P_t of N individuals: $P_t = \{x_1, \dots, x_N\}$;
2. Their fitness values: $F(x_1), \dots, F(x_N)$.

A general framework of RM-MEDA is shown in Algorithm 1.

Algorithm 1. RM-MEDA procedure

Step 0 Initialization: Set $t=0$. Generate an initial population P_0 and calculate the fitness value of each candidate solution in P_0

Step 1 Modeling: Build the probabilistic model using the $(m-1)$ -D local PCA for estimating the distribution of the candidate solutions in P_t

Step 2 Sampling: create an offspring population Q_t by sampling the established model. Evaluate the fitness value of each candidate solution in Q_t

Step 3 Selection: Select N solutions from Q_t and P_t to generate the population P_{t+1} for the next generation

Step 4 Stopping Condition: If stopping condition is met, stop and output the non-dominated individuals in P_t and their corresponding fitness vectors. All the fitness vectors form an approximation to the Pareto Front, otherwise set $t=t+1$ and go to Step2

The main purpose of modeling in RM-MEDA is to approximate one of the pieces of the PS using the solutions in one of the clusters. It is worth noting that RM-MEDA

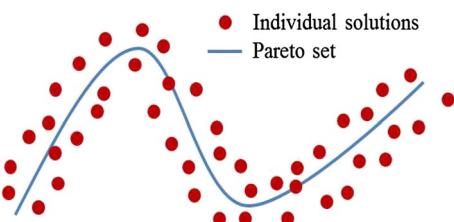


Fig. 1 The basic idea in RM-MEDA

adopts the non-dominated sorting-based selection approach proposed by Deb et al. [23] in the selection process.

2.2 IRM-MEDA

Motivated by the above observation that the number of clusters is problem-dependent and has a major impact on the performance of RM-MEDA, Wang et al. came up with an improved version of RM-MEDA, referred as IRM-MEDA [22]. Instead of adopting the fixed K during the evolution in RM-MEDA, IRM-MEDA dynamically adjusts K using the information extracted from the clustering results at each generation. A general framework of IRM-MEDA is presented in Algorithm 2.

Algorithm 2. IRM-MEDA procedure

Step 0 Initialization: Set $t=0$. Create an initial population P_0 and initialize number of clusters K

Step 1 Modeling: Build the probabilistic model using the $(m-1)$ -D local PCA for estimating the distribution of the candidate solutions in P_t

Step 2 Updating: Modify the value of K using RRCO

Step 3 Sampling: Generate an offspring population Q_t by sampling the established model. Evaluate the fitness value of each solution in Q_t

Step 4 Selection: Select N solutions from Q_t and P_t to create the population P_{t+1} for the next generation

Step 5 Stopping Condition: If the stopping criterion issatisfied, stop and output the fitness vectors of the non-dominated individuals of the final population, otherwise set $t = t + 1$ and go to Step 2.

Note that here the steps of initialization, modeling, sampling, and selection in IRM-MEDA are exactly in line with those in RM-MEDA

As mentioned therein before, the inappropriate number of clusters always results in a passive influence on the performance of the algorithm. IRM-MEDA copes with the clustering number greater than required only. If the model is built with the clustering number smaller than required, it will be incorrect. The more complex the MOPs are, the more obvious the passive influence is. Without loss of generality, the number of clusters is set to 1 under this situation. As test instances, F1 and F5 with convex PF are variants of ZDT1 [24], and F2 and F6 with concave PF are variants of ZDT2 [24]. F5 and F6 are more complex than F1 and F2, so F5 and F6 in [11] are used as the test instances. The testing code is the original code of RM-MEDA. The population size, dimensions of individual and iterations are 100, 50 and 200 respectively. The results are shown in Figs. 2 and 3.

As is seen from Figs. 2 and 3, the RM-MEDA can approach the Pareto front quickly at the beginning of optimization process both on F5 with convex PF and F6 with concave PF. But in the convergence phase, the obtained solutions

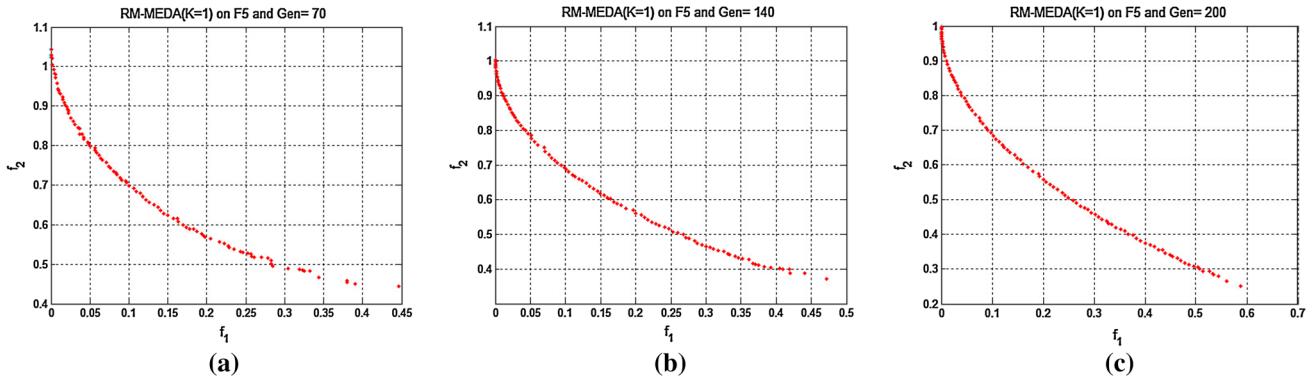


Fig. 2 Typical Pareto fronts obtained for **a** 70, **b** 140 and **c** 200 iterations on F5

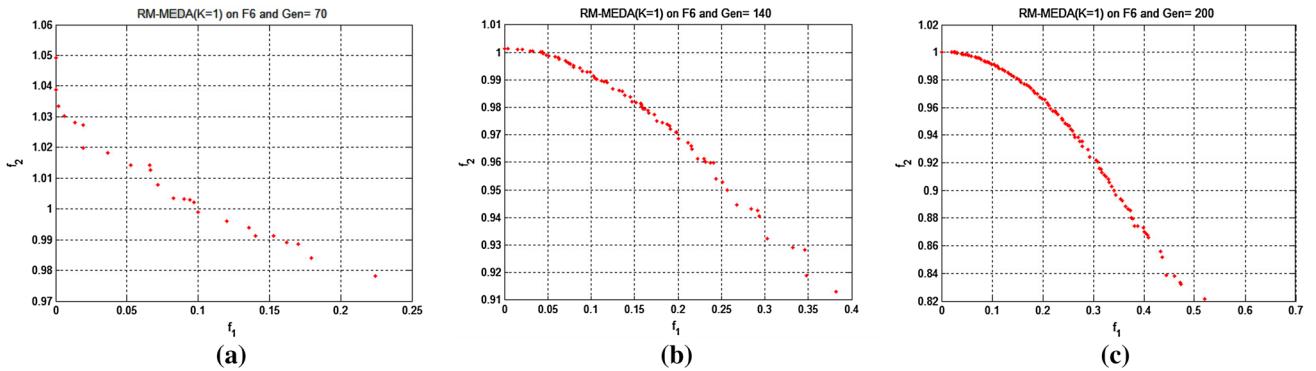


Fig. 3 Typical Pareto fronts obtained for **a** 70, **b** 140 and **c** 200 iterations on F6

are concentrated around a part of solutions, which indicates the number of clusters smaller than required leads to the diversity loss of solutions. In addition, the passive influence caused by the K value will be more obvious if test instances are more complex. The related analysis is presented in Sect. 5

3 Full variate Gaussian model

Gaussian model is the most famous of all physicists and, indeed, the Gaussian probability distribution function is also referred to as normal for it is usually supposed that errors are ‘normally’ distributed in the light of this function. So in theory the model can describe all the distribution. To keep good diversity of the solutions, the full variate Gaussian model (FGM) is brought into the proposed algorithm. The FGM assumes that the variables are all related. Assume $X = (X_1, X_2, \dots, X_n)$ follows multivariate Gaussian distribution. The normal distribution of FGM is given by formulation (3):

$$f(x_1, x_2, \dots, x_n) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} \exp(-1/2(x - \mu)^T \Sigma^{-1}(x - \mu)) \quad (3)$$

where $X = (X_1, X_2, \dots, X_n)$ denotes the population of each generation, μ and Σ are the free model parameters, and can be calculated as

$$\Sigma = \begin{bmatrix} \sigma_{11} & \dots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{n1} & \dots & \sigma_{nn} \end{bmatrix}. \quad (4)$$

In order to analyze the performance of FGM in multi-objective optimization, the classical estimation of distribution algorithm with FGM is applied to F1 and F2 in [19] with just two objectives. The algorithm is implemented by use of the Matlab toolbox for EDAs (MatEDA) [25]. The population size, dimensions of individual and iterative times are 100, 30 and 100, respectively, for all instances. In each generation, 50% of the solutions based on non-dom-inations in the population are selected for modeling (i.e., $\tau=0.5$).

Since the PF of test instances in the original paper for RM-MEDA [20] are all with convex and concave, F1 with convex and F2 with concave PF are chosen for the verification test. Figure 4 presents the typical fronts obtained by

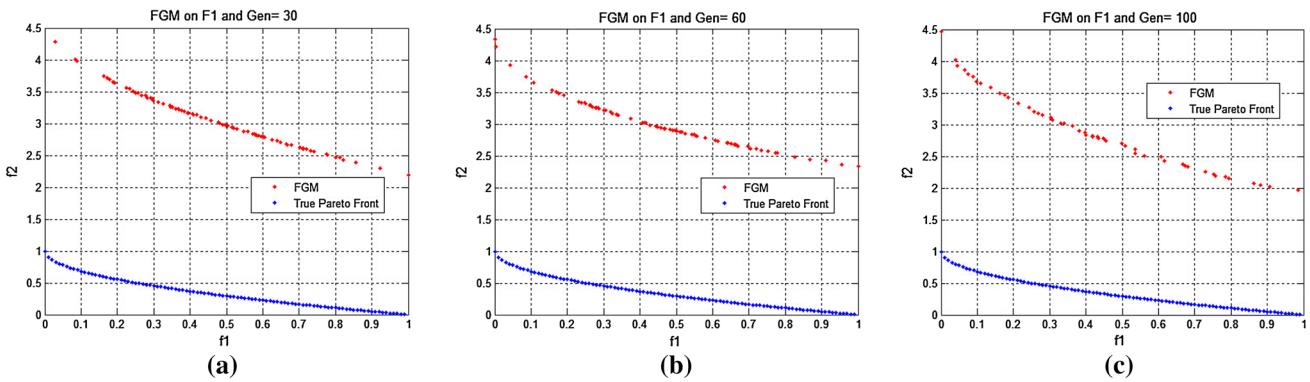


Fig. 4 Typical Pareto fronts obtained for **a** 30, **b** 60 and **c** 100 iterations on F1

the classical EDA with FGM on the test instances F1 for 30, 60 and 100 iterations, respectively.

We can see from Fig. 4 that even though the classical EDA with FGM is too simple to describe the more complex problems, it can easily keep the good diversity of solutions during the whole optimization process on F1 with convex PF. But it is also obvious that the algorithm has little capacity to make the solution set quickly approach the Pareto front.

Figure 5 presents the typical fronts obtained by the classical EDA with FGM on the test instances F2 for 30, 60 and 100 iterations, respectively.

Similarly, as is seen from Fig. 5, the classical EDA with FGM can also easily keep the good diversity of solutions during the whole optimization process on F2 with concave PF. According to the analysis above, even though the RM-MEDA with smaller K value than required can approach the Pareto front quickly during the optimization process, it loses the solution diversity. Fortunately, the FGM exhibits good performance in diversity maintenance. Consequently, the performance of RM-MEDA without clustering process could be improved by appropriately combining the FGM.

In addition, the easy implementation of FGM is another reason for us to choose it as the probabilistic model. Generally, EDAs give a probabilistic model for modeling promising solutions in the decision space ground on statistical information extracted from the previous search and then sample some new trial solutions from the built model. A very crucial issue is which kind of model should be adopted for such a task. A wonderful model should be easy to build and sample, and be in a position to depict promising areas with high fidelity [16–28]. If we want to improve the performance of RM-MEDA without clustering process by combining a probabilistic model with RM-MEDA, the chosen model should be simple, universal and good at keeping the solution diversity. The FGM is a variate related Gaussian model, so it is very easily implemented. The FGM has a very low computational cost in comparison with other probabilistic models like Bayesian Optimization Algorithm (BOA), Markovianity Optimization Algorithm (MOA) etc. Meanwhile, the Gaussian pdf is also commonly regarded as normal because it is usually assumed that its errors are ‘normally’ distributed. Thus, the Gaussian models can describe all the distribution in theory.

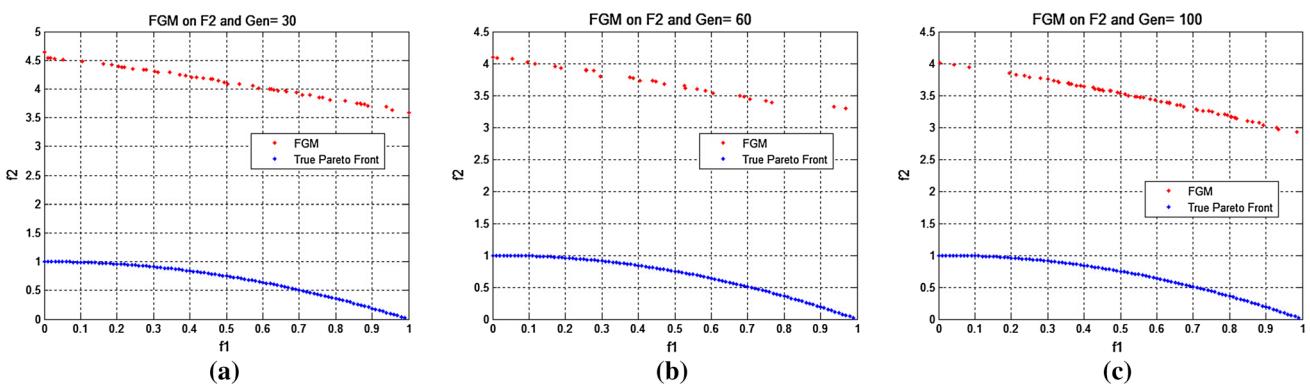


Fig. 5 Typical Pareto fronts obtained for **a** 30, **b** 60 and **c** 100 iterations on F2

Based on the conclusion above, Sect. 4 presents a novel FGM-based RM-MEDA without clustering process, referred as FRM-MEDA.

4 FRM-MEDA: the proposed algorithm

This section shows the proposed FRM-MEDA algorithm in details, where the FGM is introduced into the RM-MEDA to eliminate the influence caused by the numbers of clusters in the original algorithm. Moreover, FRM-MEDA can deal well with the issue of the number of clusters K smaller than required in RM-MEDA. The proposed FRM-MEDA works as follows:

Algorithm 3. FRM-MEDA procedure

Step 0 Initialization: Set $t=0$, $K=1$ (without clustering process).

Create an initial population P_0

Step1 Modeling: Establish the probabilistic model by the $(m-1)$ -D local PCA and select elitist solutions (10% of the solutions in the paper) to build the FGM for estimating the distribution of the candidate solutions in P_t

Step 2 Sampling: Create the offspring population Q_{t1} by sampling the $(m-1)$ -D PCA and Q_{t2} by sampling the FGM. Calculate the fitness value of each candidate solution in Q_{t1} and Q_{t2}

Step 3 Selection: Select N solutions from Q_{t1} , Q_{t2} and P_t to create the population P_{t+1} for the next generation

Step 4 Stopping Condition: If the stopping criterion is met, stop and return the fitness vectors of the non-dominated solutions of the obtained population, otherwise set $t = t + 1$ and then go to Step 2

For the better description of the FRM-MEDA, a general framework of FRM-MEDA is given as in Fig. 6.

In FRM-MEDA, the number of clusters K is fixed at 1 during the evolution. The steps of modeling, sampling and selection related to $(m-1)$ -D PCA are conducted exactly as in RM-MEDA. In addition, 10% of the solutions based on non-domination sorting in the population are selected to build the FGM. The sampling of FGM is to generate a random sequence x and $x \sim (\mu, \Sigma)$.

5 Experimental study

The Matlab toolbox for EDAs (MatEDA) [25] is a commonly used code platform proposed by Santana et al. for EDAs in 2010. The modeling and sampling methods of FGM are implemented by use of the MatEDA, in which we firstly calculate the parameters (μ, Σ) for FGM according to the current population and then sample the FGM by adopting related full Gaussian variable sampling method.

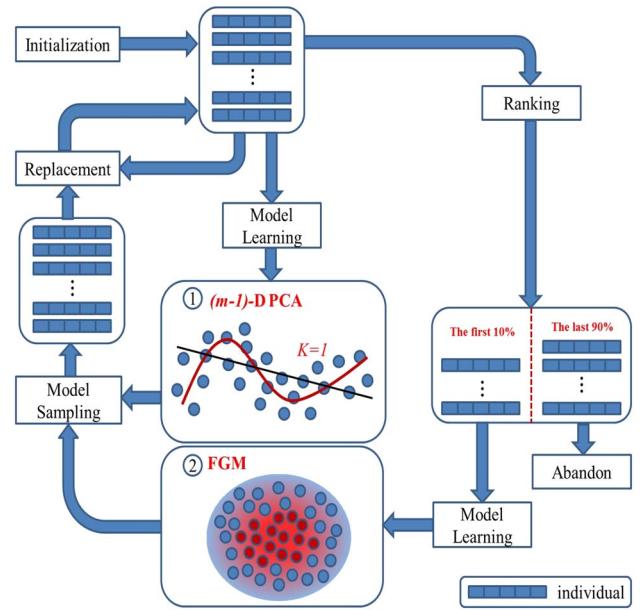


Fig. 6 The general framework of FRM-MEDA

5.1 Test instances

For the sake of rationality of the comparison result, nine test instances of the original paper about RM-MEDA (F1–F9 in [20]) are adopted to verify the effectiveness of FRM-MEDA, and its behavior is analyzed. Table 1 shows all the test instances.

According to the form of variable linkages, all the test instances could be split into two categories: test instances with linear variable linkages and those with nonlinear variable linkages, respectively. The former includes F1–F4 and the latter includes F5–F9.

5.2 General experimental setting

The experimental setting is given in this subsection. Firstly, setting K to 1 is the extreme case that the number of clusters is smaller than required, which directly revealing the performance of RM-MEDA without clustering process. So the proposed FRM-MEDA is compared with RM-MEDA with K equal to 1 and the RM-MEDA with K equal to its average number. For K proposed in the original paper is set to 5 for all instances, the average of K is set to 2 for 2 objectives and 3 for 3 objectives in this paper. Then, FRM-MEDA and RM-MEDA have the same population size for all test instances to keep consistence. The quantity of new trial individuals sampled at each generation is the same as the population size. The quantity of new individuals sampled at each generation in all the three algorithms is set to be 100 for all the bi-objective test instances and 200 for those with three objectives. The

Table 1 Test instances

Instance	Variables	Objectives	Characteristics
F1	$[0, 1]^n$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})[1 - \sqrt{f_1(\vec{x})/g(\vec{x})}]$ $g(\vec{x}) = 1 + 9[\sum_{i=2}^n (x_i - x_1)^2]/(n - 1)$	Convex PF Linear variable linkage $n = 50$
F2	$[0, 1]^n$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})[1 - (f_1(\vec{x})/g(\vec{x}))^2]$ $g(\vec{x}) = 1 + 9[\sum_{i=2}^n (x_i - x_1)^2]/(n - 1)$	Concave PF Linear variable linkage $n = 50$
F3	$[0, 1]^n$	$f_1(\vec{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(\vec{x}) = g(\vec{x})[1 - (f_1(\vec{x})/g(\vec{x}))^2]$ $g(\vec{x}) = 1 + 9[\sum_{i=2}^n (x_i - x_1)^2]/9^{0.25}$	Concave PF Non-uniformly distributed Linear variable linkage $n = 50$
F4	$[0, 1]^n$	$f_1(\vec{x}) = \cos(0.5\pi x_1) \cos(0.5\pi x_2)(1 + g(\vec{x}))$ $f_2(\vec{x}) = \cos(0.5\pi x_1) \sin(0.5\pi x_2)(1 + g(\vec{x}))$ $f_3(\vec{x}) = \sin(0.5\pi x_1)(1 + g(\vec{x}))$ $g(\vec{x}) = \sum_{i=3}^n (x_i - x_1)^2$	Concave PF Linear variable linkage 3 Objectives $n = 50$
F5	$[0, 1]^n$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})[1 - \sqrt{f_1(\vec{x})/g(\vec{x})}]$ $g(\vec{x}) = 1 + 9[\sum_{i=2}^n (x_i^2 - x_1)^2]/(n - 1)$	Convex PF Nonlinear variable linkage $n = 50$
F6	$[0, 1]^n$	$f_1(\vec{x}) = \sqrt{x_1}$ $f_2(\vec{x}) = x_1$ $f_3(\vec{x}) = g(\vec{x})[1 - (f_1(\vec{x})/g(\vec{x}))^2]$ $g(\vec{x}) = 1 + 9[\sum_{i=2}^n (x_i^2 - x_1)^2]/(n - 1)$	Concave PF Nonlinear variable linkage $n = 50$
F7	$[0, 1]^n$	$f_1(\vec{x}) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(\vec{x}) = g(\vec{x})[1 - (f_1(\vec{x})/g(\vec{x}))^2]$ $g(\vec{x}) = 1 + 9[\sum_{i=2}^n (x_i^2 - x_1)^2]/9^{0.25}$	concave PF non-uniformly distributed nonlinear variable linkage $n = 50$
F8	$[0, 1]^n$	$f_1(\vec{x}) = \cos(0.5\pi x_1) \cos(0.5\pi x_2)(1 + g(\vec{x}))$ $f_2(\vec{x}) = \cos(0.5\pi x_1) \sin(0.5\pi x_2)(1 + g(\vec{x}))$ $f_3(\vec{x}) = \sin(0.5\pi x_1)(1 + g(\vec{x}))$ $g(\vec{x}) = \sum_{i=3}^n (x_i^2 - x_1)^2$	Concave PF Nonlinear variable linkage 3 Objectives $n = 50$
F9	$[0, 1] \times [0, 10]^{n-1}$	$f_1(\vec{x}) = x_1$ $f_2(\vec{x}) = g(\vec{x})[1 - (\sqrt{f_1(\vec{x})}/g(\vec{x}))]$ $g(\vec{x}) = 1/4000 \sum_{i=2}^n (x_i^2 - x_1)^2$ $- \prod_{i=2}^n \cos((x_i^2 - x_1)/\sqrt{i-1}) + 2$	Concave PF Nonlinear variable linkage Multimodal with Griewank function $n = 50$

dimension of decision variables in all algorithms is set to be the same number 50 for all instances. Each algorithm runs 20 times independently for each test instance. And the method recommended by Nebro et al. [29] is used to determinate the specific stopping criterion, in which once the hyper-volume (*HV*) [30, 31] of P attains or goes over 98% of the *HV* value of P^* , we think that a reasonable approximation of the true PF has been achieved. The detail of *HV* will be given in the next subsection. The maximal function evaluations in each algorithm are 10,000 for F1, F2 and F4, 20,000 for F5, F6 and F8, 100,000 for F3 and F7, and 50,000 for F9 and F10, respectively. The feasible solution space is a hyper-rectangle in all the instances. And a new trial solution x will be reset to be a set of random values within the boundary using the repairing function given in the MatEDA if x is out of the boundary.

5.3 Performance metric

Three performance indicators, hyper-volume (HV), generational distance (GD) and Δ metric, are used to measure the convergence speed, convergence quality and solution diversity of each algorithm, respectively. HV, GD and Δ metric are standard evaluation indicators in multi-objective optimization field.

5.3.1 Convergence speed

For measuring the convergence speed, the stopping criterion HV is recommended by Nebro et al. Let the prescribed P^* be a set of points distributed uniformly along the true PF, and the P a set of points which are the image of the non-dominated

solutions of the current population in the objective space. A reasonable approximation of the true PF is obtained if the HV value of P attains or goes over 98% of the HV value of P^* . This indicator can effectively avoid the forced termination of algorithm when using maximum iterations. The stopping criterion is determined by Eq. (5):

$$\frac{|HV(P) - HV(P^*)|}{HV(P) + HV(P^*)} \leq 0.02 \quad (5)$$

where $HV(P)$ and $HV(P^*)$ are the HV values calculated for P^* and P , respectively. The size of P^* is set to 100,000 for all instances. Once an algorithm terminates each run, the number of function evaluations (FES) will be recorded. The mean and standard derivation of FES among 20 independent runs is used to measure the convergence speed.

In addition, the speed indicator AR proposed by Wang et al. [19] is also used to measure the convergence speed. In their paper, the AR related to FES shows the increase of the speed when comparing IRM-MEDA with RM-MEDA. Here, we compare FRM-MEDA with RM-MEDA. The AR is set as in formulation (6):

$$AR = \frac{AVE_{RM-MEDA} - AVE_{FRM-MEDA}}{AVE_{RM-MEDA}} \quad (6)$$

where $AVE_{RM-MEDA}$ and $AVE_{FRM-MEDA}$ stand for the average number of FES of RM-MEDA and FRM-MEDA, respectively.

5.3.2 Convergence quality

The GD proposed by Van Veldhuizen et al. [32] is employed to measure the convergence quality. The generational distance between P and P^* is defined by formulation (7):

$$GD = \frac{\sqrt{\sum_{i=1}^{|P|} d(P_i, P^*)^2}}{|P|} \quad (7)$$

where $d(P_i, P^*)$ denotes the minimum Euclidean distance of $P_i - P^*$.

GD shows the mean distance of the points in an approximation to the true PF. To obtain a low GD, P has to be extremely close to the PF. The best case is that GD equals to 0, i.e., all the points in P are on the true PF. The size of P^* is set to 500 for bi-objective instances and 1000 for 3 objectives.

5.3.3 Population diversity

The Δ -metric [17] is proposed to measure solution diversity. The Δ metric can be described as follows:

$$\Delta = \frac{d_f + d_l + \sum_i^N |d_i - \bar{d}|}{d_f + d_l + (N - 1)\bar{d}} \quad (8)$$

where N is the number of non-dominated solutions. d_f and d_l are distances of the extreme solutions that need to be calculated firstly by adjusting a curve parallel to that of the PF. d_i is a Euclidean distance between consecutive solutions in the obtained non-dominated set and \bar{d} is the mean value of these distances.

The maximum value of Δ can be greater than 1 according to the formulation (8). If the non-dominated set gets an excellent distribution, it would make all distances d_i equal to \bar{d} and meanwhile make d_f and d_l equal to 0 respectively. In this case, Δ equal to 0. The lower the Δ is, the better the solution diversity is.

The comparison of results between the algorithms is based on their statistical analysis. Thus, the Wilcoxon signed ranks test [33] is used to check the statistical differences on the performance of the algorithms. Let the null hypothesis be that all the algorithms have equivalent performance. When the null hypothesis that all the algorithms have an equal average rank is rejected for a specific problem configuration with a p-value less than 0.05, the entry related to the algorithm with the best Wilcoxon signed rank is shown in bold in the following subsections.

5.4 Experimental results and analysis

5.4.1 Convergence speed

The maximum number of FES can be found in the experimental setting. And as is stated previously, the mean and standard deviation of FES have been recorded in each run. The specific data of the performance indicators and their statistical results are shown in Table 2. “Mean FES” and “Std Dev” represent the mean and standard deviation of FES of 20 independent runs. “AR” indicates the acceleration rate.

It is clear in Table 2 that both FRM-MEDA and RM-MEDA with clustering process perform better than RM-MEDA without clustering process on all the test instances. Furthermore, FRM-MEDA saves about 72 and 21% FES in comparison to the RM-MEDA without clustering process and RM-MEDA with clustering process, respectively. The convergence speed of RM-MEDA without clustering process is too slow.

5.4.2 Convergence quality

The convergence quality is measured by the performance indicator GD. In this subsection, we set the same maximum of FES in solving the same instance for all algorithms. K is set to the same number as is stated in the previous subsection. According to the results in Table 2, the convergence speed of RM-MEDA without clustering process is too slow. Thus, to obtain 98% of the HV value of the true PF for all

Table 2 Results of convergence speed over 20 independent runs

Instance	FRM-MEDA Mean FES \pm Std Dev ($\times 10^3$)	RM-MEDA (with different number of K)	
		Mean FES \pm Std Dev ($\times 10^3$)/ AR	
		$K=1$	$K=AVE_K$
F1	5.450 \pm 1.7864	22.070 \pm 2.7172/75%	6.965 \pm 0.4271/22%
F2	8.110 \pm 0.7122	36.940 \pm 5.8770/78%	9.015 \pm 0.6792/11%
F3	110.610 \pm 7.2813	318.060 \pm 26.647/65%	171.325 \pm 20.935/35%
F4	17.220 \pm 1.6389	159.960 \pm 3.1828/89%	26.000 \pm 8.0498/34%
F5	10.400 \pm 3.3179	47.365 \pm 5.6276/78%	11.690 \pm 3.1611/11%
F6	13.180 \pm 1.7668	35.510 \pm 4.9184/63%	16.010 \pm 1.7897/18%
F7	144.450 \pm 15.443	394.190 \pm 35.324/63%	158.610 \pm 14.189/9%
F8	42.240 \pm 12.085	168.200 \pm 24.623/75%	65.550 \pm 16.283/36%
F9	23.400 \pm 3.9318	61.090 \pm 10.381/62%	26.510 \pm 3.4371/12%
Mean of AR		72%	21%

The performance of the algorithm with bold data is superior to other algorithms

algorithms, the maximum of FES for every instance is set at the mean value of FES required by RM-MEDA with K equal to AVE_K . So the maximum of FES are set to be 10,000 for F1 and F2, 200,000 for F3 and F7, 30,000 for F4 and F9, 20,000 for F5 and F6 and 100,000 for F8. The results of all 20 independent runs for all the instances in GD are given in Table 3. “Mean GD” and “Std Dev” represent the mean and standard deviation of GD of 20 independent runs, respectively.

Table 3 shows that the FRM-MEDA provide evidently lower GD than RM-MEDA without clustering process and RM-MEDA with K equal to AVE_K for all test instances, although it is worse than RM-MEDA with K equal to AVE_K in test instances F9. To assure the results of convergence quality with statistical confidence, the Wilcoxon signed ranks test at a 0.05 significance level is utilized and the actual concrete data of the statistical analysis is presented in Table 4. “ $R+$ ” is the sum of ranks for the problems in

which the first algorithm outperforms the second with the “ p -value” associated, and “ $R-$ ” for the opposite.

The results from Table 4 demonstrate that FRM-MEDA significantly outperforms RM-MEDA without clustering process for all the instances and has the same performance with RM-MEDA with K equal to AVE_K .

5.4.3 Population diversity

In this subsection, the performance indicator Δ metric is adopted to measure the diversity of the obtained population. K and FES are set to the same value as shown in the previous subsection. The results are given in Table 5. “Mean Δ ” and “Std Dev” represent the mean and standard deviation of Δ of 20 independent runs, respectively.

It can be seen from Table 5 that FRM-MEDA is significantly superior to RM-MEDA without clustering process in terms of GD, although it is worse than RM-MEDA with K

Table 3 Results of convergence quality over 20 independent runs

Instance	The number of FES	FRM-MEDA Mean GD \pm Std Dev ($\times 10^{-3}$)	RM-MEDA	
			Mean GD \pm Std Dev ($\times 10^{-3}$)	
			$K=1$	$K=AVE_K$
F1	10,000	0.2842 \pm 0.1771	5.1438 \pm 0.0707	0.3304 \pm 0.1788
F2	10,000	0.1159 \pm 0.0067	3.5333 \pm 0.0992	0.1169 \pm 0.0149
F3	200,000	3.8790 \pm 0.4103	98.6847 \pm 39.0190	11.2200 \pm 2.5227
F4	30,000	1.3956 \pm 0.0685	414.6459 \pm 66.0754	5.5573 \pm 8.8936
F5	20,000	0.6813 \pm 0.0637	0.9077 \pm 0.4840	0.8542 \pm 0.9804
F6	20,000	0.7331 \pm 0.0839	1.2104 \pm 0.4882	1.1249 \pm 1.5780
F7	200,000	21.0767 \pm 9.6560	54.0953 \pm 364.4290	39.1499 \pm 2.0044
F8	100,000	3.5265 \pm 0.0303	298.9802 \pm 100.6249	22.4138 \pm 28.7962
F9	30,000	1.7818 \pm 0.8570	4.8436 \pm 1.9067	1.5835 \pm 0.6869

The performance of the algorithm with bold data is superior to other algorithms

Table 4 Wilcoxon signed ranks test results of convergence quality with a level of significance $\alpha=0.05$

Instance	Comparison ($R+$, $R-$ and p -value) on GD					
	FRM-MEDA vs RM-MEDA ($K=1$)			FRM-MEDA vs RM-MEDA ($K=AVE_K$)		
	R^+	R^-	p value	R^+	R^-	p value
F1	20	0	0.000	12	8	0.279
F2	20	0	0.000	10	10	0.970
F3	20	0	0.000	20	0	0.000
F4	20	0	0.000	18	2	0.000
F5	18	2	0.001	3	17	0.927
F6	19	1	0.000	9	11	0.709
F7	17	3	0.002	20	0	0.000
F8	20	0	0.000	14	6	0.015
F9	18	2	0.000	9	11	0.654

The bold data of “ p -value” demonstrate that the performance of FRM-MEDA significantly outperfrom the compared algorithm

Table 5 Results of population diversity

Instance	The number of FES	FRM-MEDA		RM-MEDA	
		Mean $\Delta \pm Std Dev$		Mean $\Delta \pm Std Dev$	
		$K=1$	$K=AVE_K$	$K=1$	$K=AVE_K$
F1	20,000	0.1862 ± 0.0124		0.3716 ± 0.0341	0.1891 ± 0.0142
F2	20,000	0.1858 ± 0.0086		0.3335 ± 0.0991	0.1907 ± 0.0096
F3	100,000	0.5864 ± 0.0073		0.9469 ± 0.1242	0.6091 ± 0.0099
F4	30,000		0.7027 ± 0.0220	0.7705 ± 0.0462	0.5889 ± 0.0268
F5	20,000		0.1851 ± 0.0113	0.4898 ± 0.0876	0.1800 ± 0.0244
F6	20,000	0.1934 ± 0.0182		0.5280 ± 0.2933	0.1936 ± 0.0675
F7	200,000		0.8137 ± 0.0985	1.1135 ± 0.2503	0.7082 ± 0.0976
F8	30,000	0.6466 ± 0.0246		1.0928 ± 0.0965	0.7928 ± 0.3157
F9	100,000		0.2112 ± 0.0694	0.5581 ± 0.3974	0.2105 ± 0.0371

The performance of the algorithm with bold data is superior to other algorithms

Table 6 Wilcoxon signed ranks test results of population diversity with a level of significance $\alpha=0.05$

Instance	Comparison ($R+$, $R-$ and p -value) on Δ					
	FRM-MEDA vs RM-MEDA ($K=1$)			FRM-MEDA vs RM-MEDA ($K=AVE_K$)		
	R^+	R^-	p value	R^+	R^-	p value
F1	20	0	0.000	11	9	0.502
F2	20	0	0.000	14	6	0.093
F3	20	0	0.000	19	1	0.000
F4	17	3	0.000	0	20	1.000
F5	20	0	0.000	5	15	0.865
F6	17	3	0.001	6	14	0.855
F7	19	1	0.000	4	16	0.995
F8	20	0	0.000	8	12	0.668
F9	18	2	0.001	10	10	0.502

The bold data of “ p -value” demonstrate that the performance of FRM-MEDA significantly outperfrom the compared algorithm

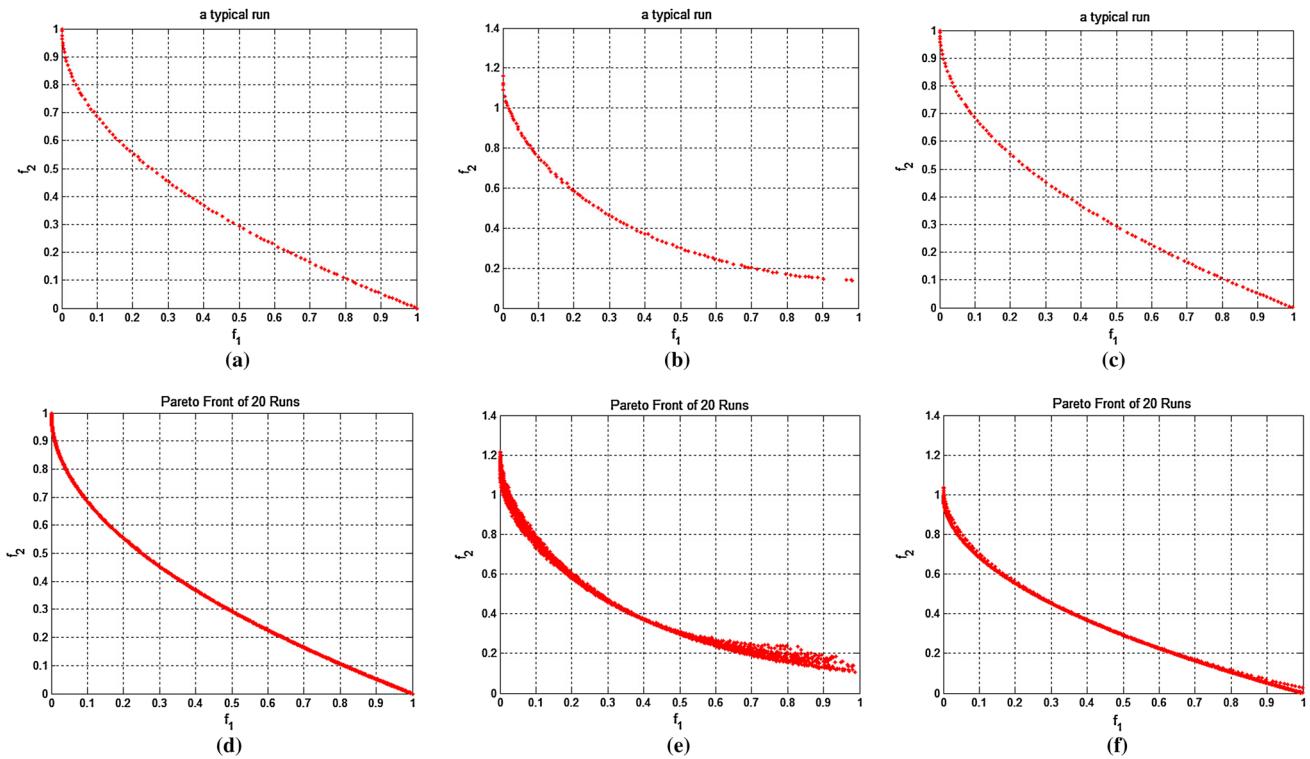


Fig. 7 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on F1. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

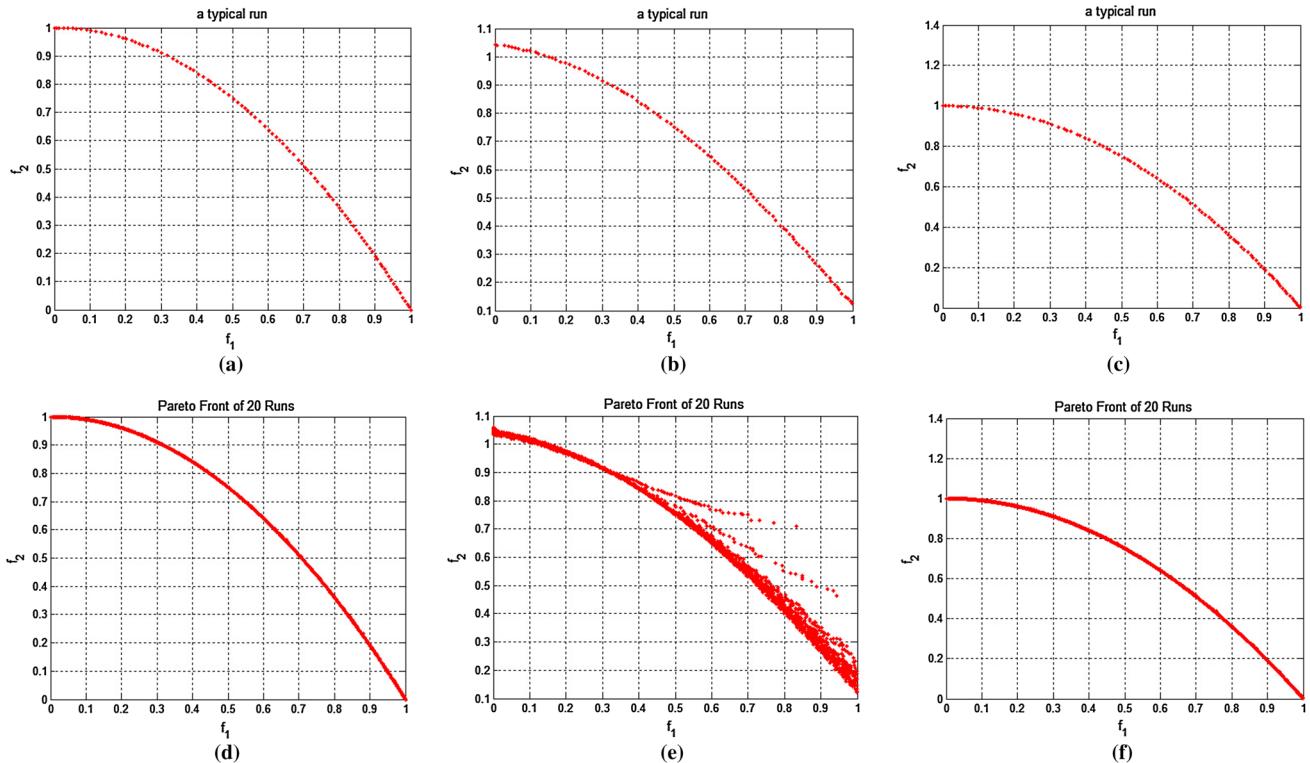


Fig. 8 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on F2. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

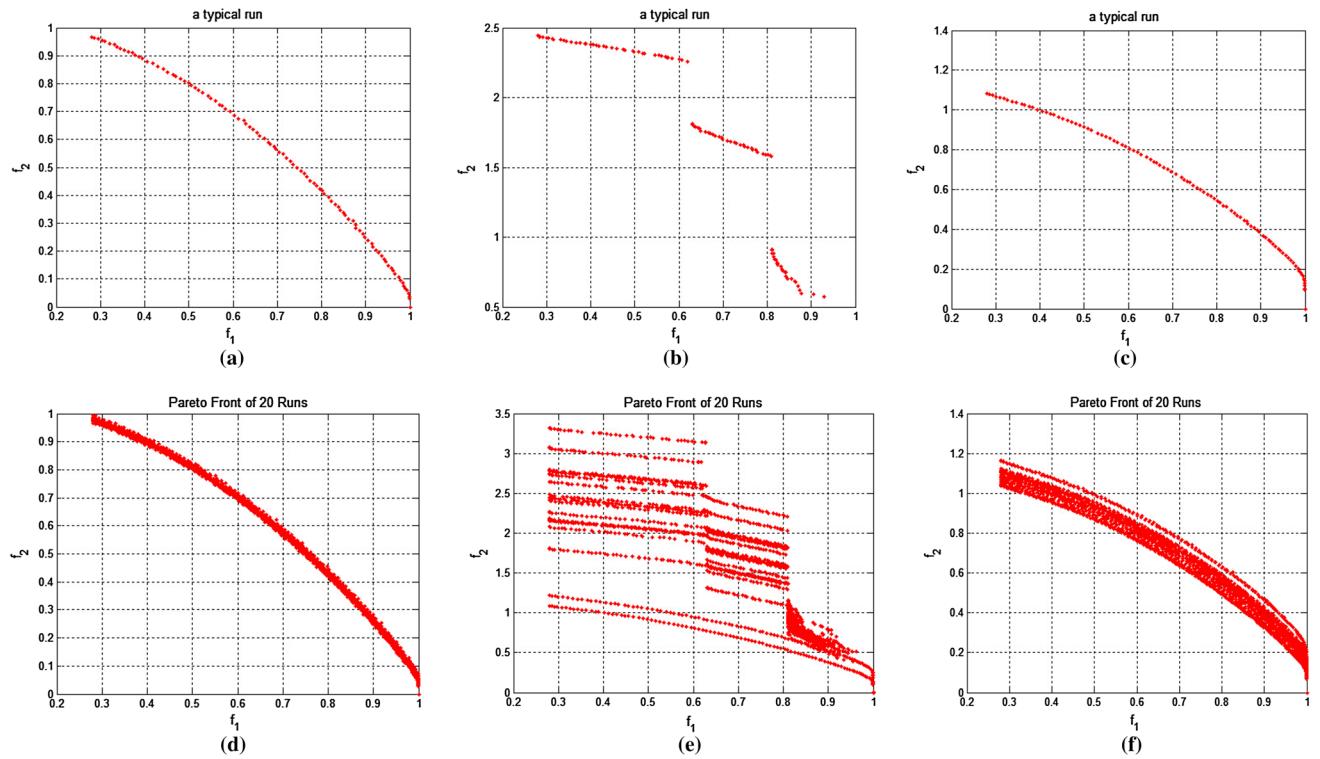


Fig. 9 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on F3. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

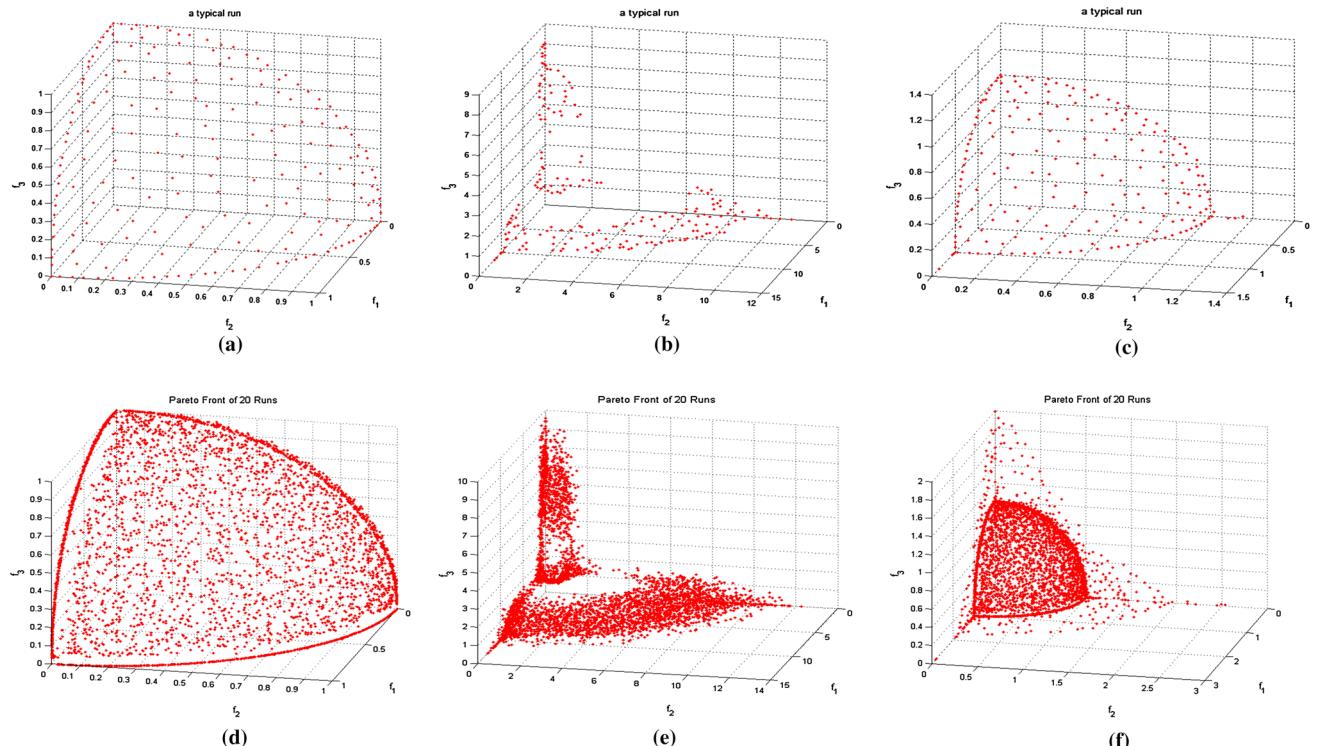


Fig. 10 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 3 in the objective space on F4. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

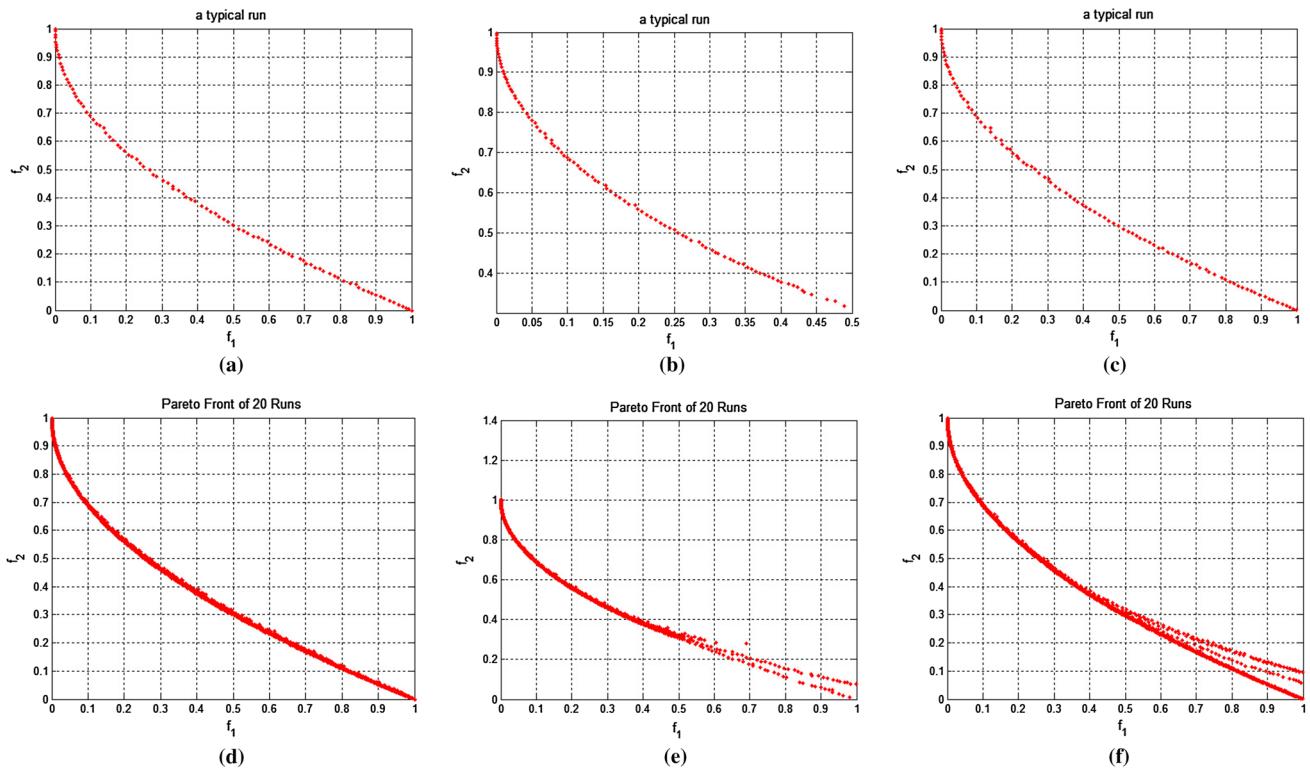


Fig. 11 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on F5. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

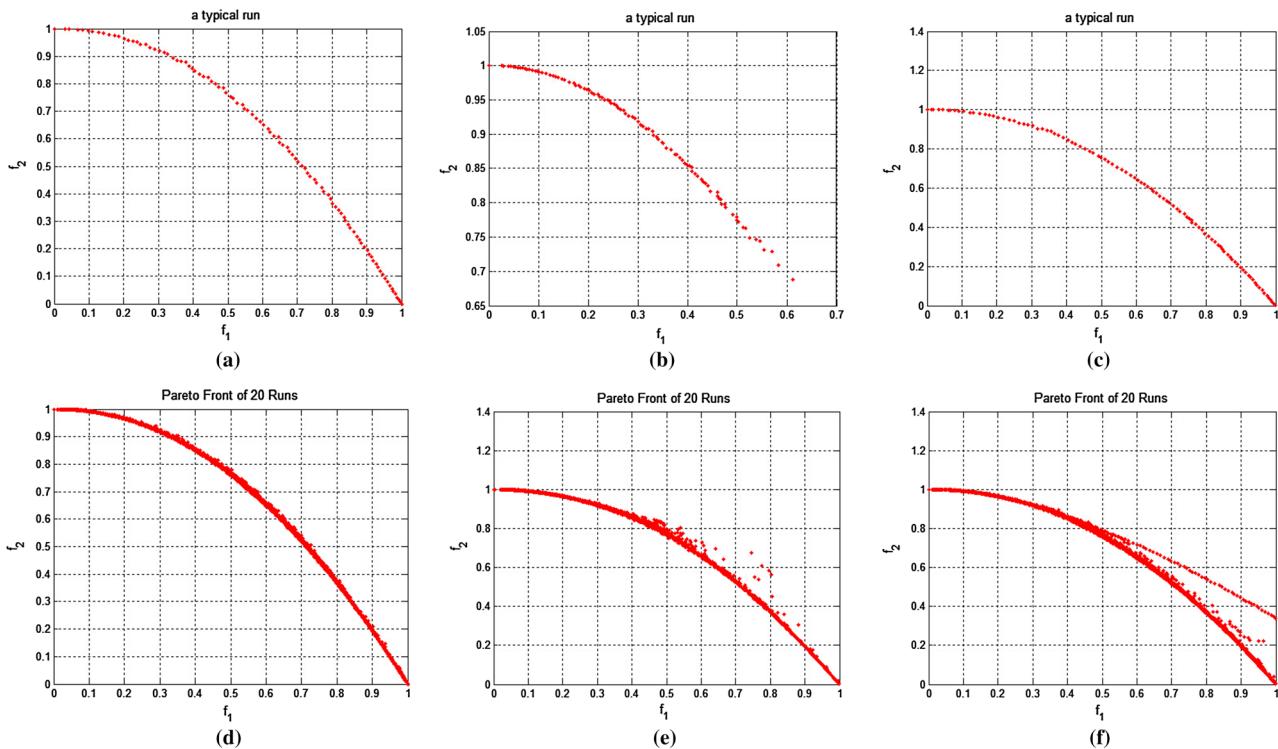


Fig. 12 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on F6. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

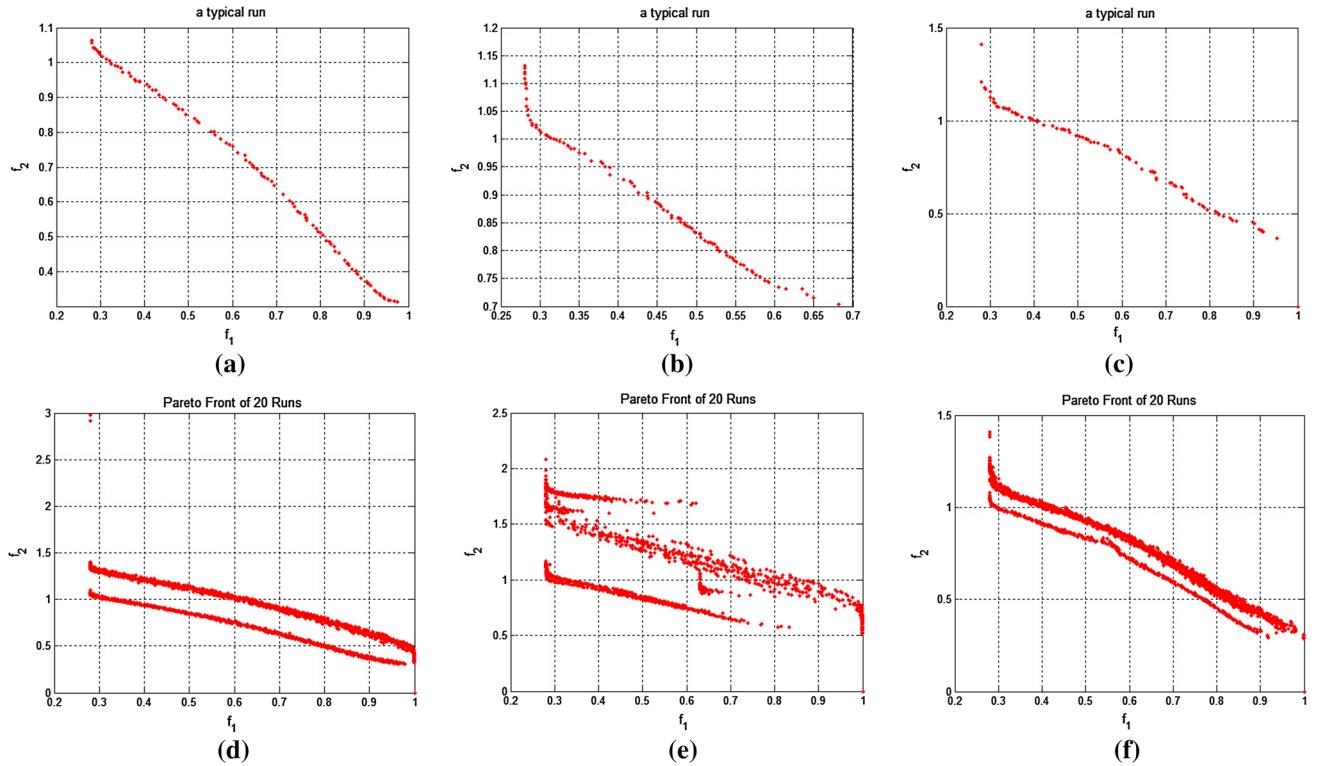


Fig. 13 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on F7. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

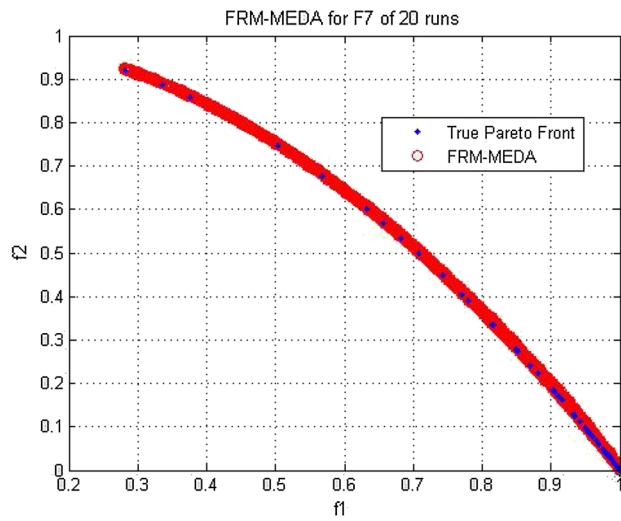


Fig. 14 Plots of the non-dominated individuals of 20 runs obtained by the FRM-MEDA based on new strategies

equal to AVEK in test instances F4, F5, F7 and F9. Wilcoxon signed ranks are performed to ensure the results of population diversity with statistical confidence. The actual concrete data of the statistical analysis on Δ are presented in Table 6.

The results in Table 6 indicate that FRM-MEDA significantly outperforms RM-MEDA without clustering process for all test instances. But FRM-MEDA is inferior to the RM-MEDA with K equal to AVEK in their statistical indicators in terms of Δ metric.

5.4.4 The typical fronts

The typical fronts of 20 runs obtained for all algorithms are also exhibited as follows. Figures 7, 8, 9, 10, 11, 12, 13, 14, 15 plot the non-dominated solutions of the populations obtained finally by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on all test instances, respectively. In Figs. 7, 8, 9, 10, 11, 12, 13, 14, 15, figure (a), (b) and (c) are in a typical run, and (d), (e) and (f) are in 20 runs, respectively. Both the convergence quality and the population diversity can be directly observed from the figures.

Figures 7 and 8 show that both FRM-MEDA and RM-MEDA with K equal to AVEK can obtain good non-dominated fronts in a typical run and 20 independent runs. The model built by RM-MEDA without the clustering process might not be suitable for solving F1 and F2 because the number of clusters is smaller than required, which has a

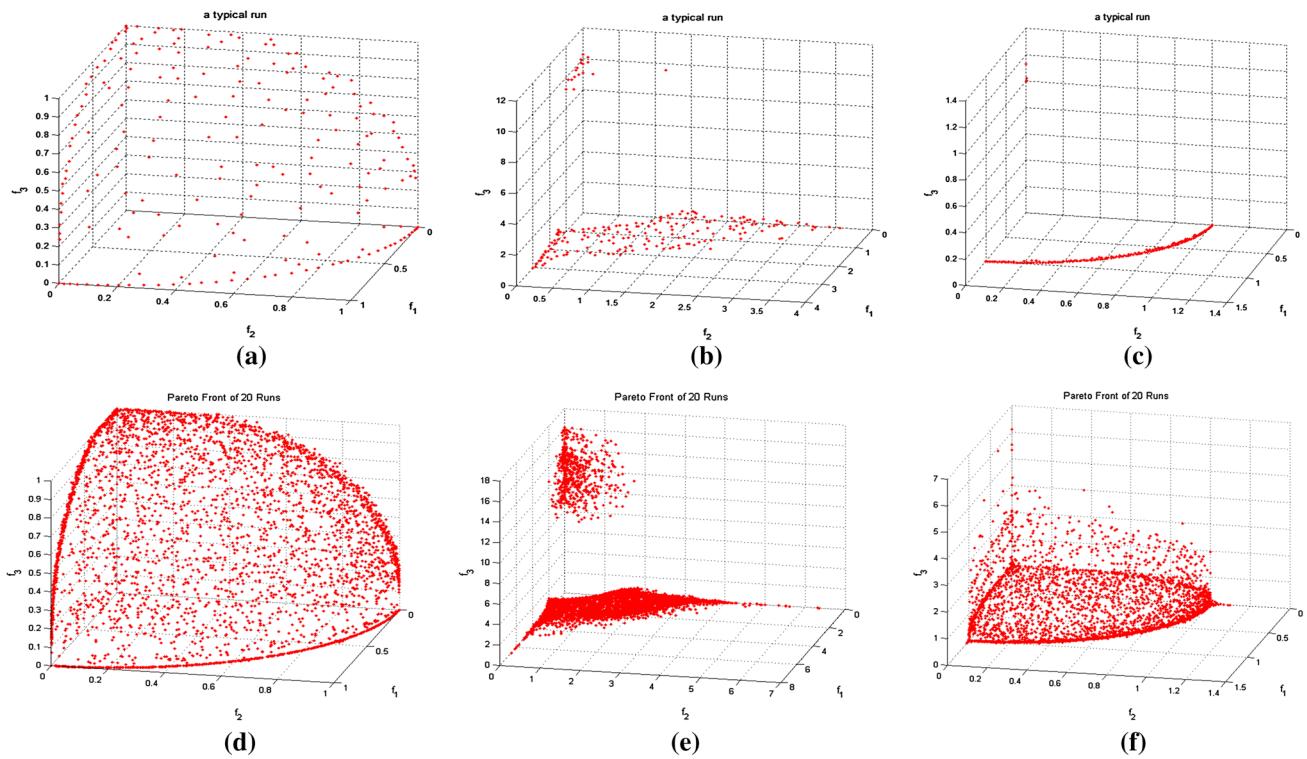


Fig. 15 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 3 in the objective space on F8. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

passive effect on the convergence speed, convergence quality and diversity of the population.

As can be seen from Figs. 9, 10, 11 and 12, all the figures indicate that FRM-MEDA can obtain good non-dominated fronts both in a typical run and 20 independent runs. The performance of RM-MEDA with K equal to AVEK is similar to FRM-MEDA except its lower convergence speed. As for RM-MEDA without clustering process, its built model will sometimes be incorrect when the number of clusters is smaller than required, which would directly reduce the convergence speed, seriously influence the convergence quality and the population diversity as shown in subplot (b) and (e) of Figs. 9, 10, 11 and 12.

As is shown in Fig. 13, FRM-MEDA and RM-MEDA with K equal to 2 have acceptable performance in terms of population diversity even though the performance of FRM-MEDA is not significantly outperforming the RM-MEDA with K equal to 2. However, the performance of all the algorithms on F7 is relatively worse than that on other instances. One reason is that the dimension of individuals in this paper ($n=50$) is set to be much larger than that in the original paper ($n=30$), making F7 more complicated. Another reason is that the exploring ability of FRM-MEDA and RM-MEDA is not strong enough. When the dimension of individuals increases dramatically, the interaction of each dimension has a strong

impact on the exploring ability of algorithms. Both these two algorithms are unable to explore new trial individuals. When sampling the full variate Gaussian model, a very low variance will lead to the stagnation of FRM-MEDA. In order to promote the activity of FRM-MEDA, we zoom in the scaling for the covariance values. Meanwhile, an accelerating factor $a = [1/MI, 2/MI, \dots, 1]^T$ is brought into the algorithm to improve its exploring ability. MI is the maximum number of iterations. The plot of non-dominated individuals of the final population is presented in Fig. 14. The result shows that the performance of FRM-MEDA with new strategies is significantly better than that of RM-MEDA with K equal to 2. In addition, RM-MEDA without clustering process performs badly in all indicators.

It is clear as in Fig. 15 that RM-MEDA without clustering process cannot converge to the PF and RM-MEDA with K equal to 3 also easily misses some parts of the PF both in a typical run and 20 independent runs due to their wrongly built model. By contrast, FRM-MEDA significantly outperforms the RM-MEDA in terms of the three indicators.

It can be observed from Fig. 16 that all the algorithms have the similar performance on F9 in terms of all the indicators. But RM-MEDA without clustering process has a lower convergence rate than FRM-MEDA and

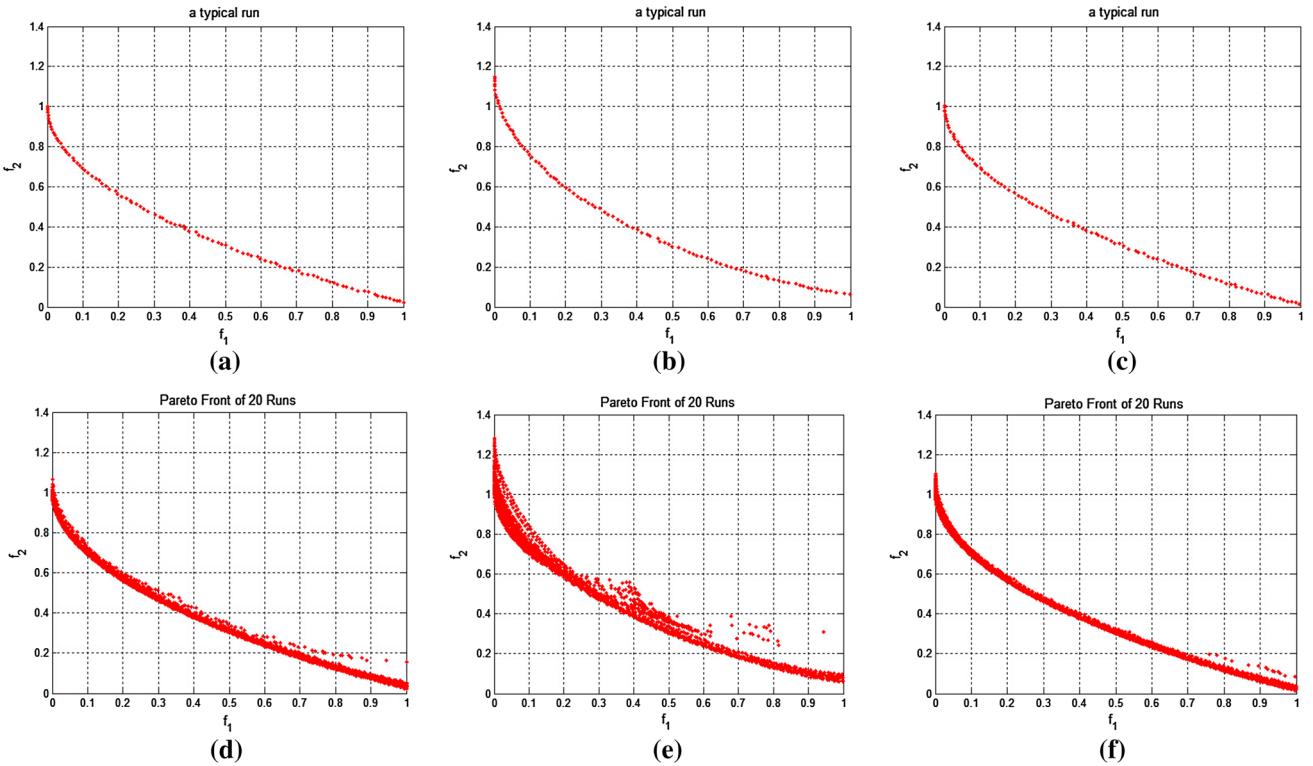


Fig. 16 Plots of the non-dominated individuals of the final populations obtained by the FRM-MEDA, RM-MEDA without clustering process and RM-MEDA with K equal to 2 in the objective space on F9. The figure **a–c** are in a typical runs and **d–f** are in 20 runs, respectively

RM-MEDA with K equal to 2, and sometimes will not converge.

5.4.5 The application of FRM-MEDA

The proposed FRM-MEDA can be used to solve many problems in the areas like image processing, natural language processing, and engineering application and so on. Actually, the proposed FRM-MEDA has been applied to the registration project about images of the Dazu Rock Carvings. Here, a brief example of image registration using FRM-MEDA is presented as follows.

The aim of image registration is to find the transformation matrix. To insure the accuracy of the transformation matrix, one objective of the multi-objective model is to minimize the mean distance \bar{d} between the obtained H and the corresponding data set. The first objective can be formulated as:

$$\text{minimize } f_1(X) = \bar{d}(H, P) \quad (9)$$

Another objective of the multi-objective model is to minimize the threshold distance d_T which can maximize the number of inliers on the basis of the given \bar{d} . This objective is set to improve the robustness of the transformation

matrix as much as possible. The second objective can be formulated as:

$$\text{minimize } f_2(X) = \arg \max(N_{in}(dT, \bar{d})) \quad (10)$$

In formulation (2), N_{in} is the function of d_T to count the number of inliers. The N_{in} is affected by the d in f_1 .

Thus, the image registration problem is described as a bi-objectives optimization problem.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (11)$$

The matrix above is just the transformation matrix that we need to get all the eight unknown parameters h_1 – h_8 . As is seen in the formulation (11), there are nine parameters in total. Therefore, each variable in EDA is set to nine dimensions and each dimension corresponds to a parameter in the transformation matrix. The variable is given as:

$$x = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, 1) \quad (12)$$

Under the variable description, the unknown parameters of perspective transformation are obtained by using the FRM-MEDA. Figure 17 shows the registration results



Fig. 17 **a, b** Are two input images obtained under different situations. **c** Is the aligned image obtained by FRM-MEDA

which are acceptable even though there are still some flaws like the mis-alignment in the eye of the Buddha.

6 Conclusion

In combination with FGM, a novel FGM-based RM-MEDA without clustering process is presented in this paper. The proposed algorithm, FRM-MEDA, is an improvement of the well-known multi-objective estimation of distribution algorithm RM-MEDA.

The FRM-MEDA is put forward to deal with the situation where the number of clusters K is smaller than required. We analyze the case and find the great loss of population diversity under the situation. In this regard, the full variate Gaussian model (FGM) is brought into the

algorithm to make up for the deficiency. A crucial nature of FGM is that it can easily keep good diversity of solutions during the entire optimization process. The results of exhaustive experiments in the instances of F1–F9 show that in comparison with RM-MEDA without clustering process and RM-MEDA with K equal to $AVEK$, the FRM-MEDA is the fastest and the most stable algorithm for all the test instances in terms of the convergence speed, convergence quality and population diversity.

However, it is necessary to notice that although the FRM-MEDA significantly outperforms the RM-MEDA without clustering process, its superiority is still not great when compared with the RM-MEDA with K equal to $AVEK$ on Δ metric according to the statistical analysis. That is because the FGM is too simple to explore some

parts of solution space and cannot fully describe population of more complex MOPs.

One of the future studies related to this work is to propose a scheme to improve the population diversity. The division of solution space and the combination of a variety of sampling methods will also be considered in our future work. Then, a variable variance will be considered into our work to make the algorithm break its stagnation state and improve the exploring ability. Furthermore, some excellent characteristics found in this work can be applied together with other strategies to improve the performance of EDAs in solving more complex high-dimensional multi-objective optimization problems.

Acknowledgements This work has been funded by the Project of Science and Technology for Graduate Students with No. CDJXS12180003, and the Scientific and Technological Research Program of Chongqing Municipal Education Commission under Grant No. KJ1400409.

References

- Taormina R et al (2015) Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and extreme learning machines. *J Hydrol* 529(3):1617–1632
- Zhang J et al (2009) Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization. *J Univers Comput Sci* 15(4):840–858
- Wang WC et al (2015) Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. *Water Resour Manag* 29(8):2655–2675
- Zhang SW et al (2009) Dimension reduction using semi-supervised locally linear embedding for plant leaf classification. *Lect Notes Comput Sci* 5754:948–955
- Wu CL et al (2009) Methods to improve neural network performance in daily flows prediction. *J Hydrol* 372(1–4):80–93
- Chau KW et al. (2010) A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. *J Hydroinf* 12(4):458–473
- Larrañaga P, Lozano JA (2002) Estimation of distribution algorithms: a new tool for evolutionary computation [M]. Kluwer Press, Boston
- Bengoetxea E, Larrañaga P, Bloch I et al (2001) Estimation of distribution algorithms: a new evolutionary computation approach for graph matching problems[C]. Energy minimization methods in computer vision and pattern recognition. Springer, Berlin, pp 454–469
- Pelikan M, Goldberg DE, Lobo FG (2002) A survey of optimization by building and using probabilistic models [J]. *Comput Optim Appl* 21(1):5–20
- Laumanns M, Ocenasek J (2002) Bayesian optimization algorithms for multi-objective optimization [M]. Parallel Problem Solving from Nature-PPSN VII. Springer, Berlin, pp 298–307
- Pelikan M, Sastry K, Goldberg DE (2005) Multi-objective hBOA, clustering, and scalability[C]. In: Proceedings of the 2005 conference on Genetic and evolutionary computation ACM, pp 663–670
- Sastry K, Goldberg DE, Pelikan M (2005) Limits of scalability of multi-objective estimation of distribution algorithms[C]. In: Evolutionary Computation, 2005. The 2005 IEEE Congress on IEEE 3: pp 2217–2224
- Wang X, Dong L, Yan J (2012) Maximum ambiguity based sample selection in fuzzy decision tree induction. *IEEE Trans Knowl Data Eng* 24(8):1491–1505
- Valdez SI, Hernández A, Botello S (2013) A Boltzmann based estimation of distribution algorithm [J]. *Inf Sci* 236(1):126–137
- Gonzalez-Fernandez Y, Soto M (2014) Copulaedas: an R package for estimation of distribution algorithms based on copulas [J]. *J Stat Softw* 58(9):1–34
- Maezawa K, Handa H (2015) Memetic algorithms of graph-based estimation of distribution algorithms [C]. In: The 18th Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES 2014). Springer International Publishing 2, pp 647–656
- Miettinen K (1999) Nonlinear multi-objective optimization [M]. Springer, Berlin
- Ehrgott M (2005) Multicriteria optimization [M]. Springer, Berlin
- Schütze O, Mostaghim S, Dellnitz M et al (2003) Covering Pareto sets by multilevel evolutionary subdivision techniques [C]. Evolutionary multi-criterion optimization. Springer, Berlin, pp 118–132
- Zhang Q, Zhou A, Jin Y (2008) RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm [J]. *Evol Comput IEEE Trans* 12(1):41–63
- Kambhatla N, Leen TK (1997) Dimension reduction by local principal component analysis [J]. *Neural Comput* 9(7):1493–1516
- Wang Y, Xiang J, Cai Z (2012) A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator [J]. *Appl Soft Comput* 12(11):3526–3538
- Deb K, Pratap A, Agarwal S et al. (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II [J]. *Evol Comput IEEE Trans* 6(2):182–197
- Zitzler E, Deb K, Thiele L (2000) Comparison of multi-objective evolutionary algorithms: empirical results [J]. *Evol Comput* 8(2):173–195
- Santana R, Bielza C, Larrañaga P et al. (2010) Mateda-2.0: estimation of distribution algorithms in MATLAB [J]. *J Stat Softw* 35(7):1–30
- Zhang Q (2004) On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm [J]. *Evol Comput IEEE Trans* 8(1):80–93
- Zhang Q, Muhlenbein H (2004) On the convergence of a class of estimation of distribution algorithms [J]. *Evol Comput IEEE Trans* 8(2):127–136
- Zhang Q (2004) On the convergence of a factorized distribution algorithm with truncation selection [J]. *Complexity* 9(4):17–23
- Nebro AJ, Durillo JJ, Coello CAC et al (2008) A study of convergence speed in multi-objective metaheuristics[M]. Parallel problem solving from nature-PPSN X. Springer, Berlin, pp 763–772
- Charnetski JR, Soland RM (1978) Multiple-attribute decision making with partial information: the comparative hypervolume criterion [J]. *Naval Res Logist Q* 25(2):279–288
- Zitzler E, Thiele L (1999) Multi-objective evolutionary algorithms: a comparative case study and the strength Pareto approach [J]. *Evol Comput IEEE Trans* 3(4):257–271
- Van Veldhuizen DA, Lamont GB (2000) On measuring multi-objective evolutionary algorithm performance. In: Proceedings of the 2000 IEEE Congress on Evolutionary Computation, CEC 2000, vol 1. IEEE Service Center, Piscataway, New Jersey, pp 204–211
- Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18