



A general framework for evolutionary multiobjective optimization via manifold learning

Ke Li, Sam Kwong*

Department of Computer Science, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong



ARTICLE INFO

Article history:

Received 2 December 2013

Received in revised form

24 February 2014

Accepted 4 March 2014

Available online 15 July 2014

Keywords:

Regularity

Principal curve

Laplacian eigenmaps

Manifold learning

Evolutionary computation

Multiobjective optimization

ABSTRACT

Under certain mild condition, the Pareto-optimal set (PS) of a continuous multiobjective optimization problem, with m objectives, is a piece-wise continuous $(m-1)$ -dimensional manifold. This regularity property is important, yet has been unfortunately ignored in many evolutionary multiobjective optimization (EMO) studies. The first work that explicitly takes advantages of this regularity property in EMO is the regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA). However, its performance largely depends on its model parameter, which is problem dependent. Manifold learning, also known as nonlinear dimensionality reduction, is able to discover the geometric property of a low-dimensional manifold embedded in the high-dimensional ambient space. This paper presents a general framework that applies advanced manifold learning techniques in EMO. At each generation, we first use a principal curve algorithm to obtain an approximation of the PS manifold. Then, the Laplacian eigenmaps algorithm is employed to find the low-dimensional representation of this PS approximation. Afterwards, we identify the neighborhood relationship in this low-dimensional representation, which is also applicable for the original high-dimensional data. Based on the neighborhood relationship, we can interpolate new candidate solutions that obey the geometric property of the PS manifold. Empirical results validate the effectiveness of our proposal.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

Multiobjective optimization problems (MOPs), which naturally arise in many disciplines such as engineering [20], economics [26] and logistics [22], involve more than one objective to optimize simultaneously. Different from single objective optimization, an MOP, due to the conflicts of its objectives, does not have a unique solution. Instead, one can expect to find a set of trade-off solutions that compromise all these objectives. Evolutionary algorithm (EA), a nature-inspired and population-based optimization technique, has been widely accepted as a major approach for MOPs. Over the last two decades, much effort has been devoted to developing evolutionary multiobjective optimization (EMO) algorithms (e.g., [12–14,16,18]), many of which have been successfully applied to a wide range of problem domains [6,5,17]. There are two major operations in EMO algorithms: one is recombination, which considers generating offspring solutions, the other is selection, which determines the survival of elite solutions as the next parents. It is surprising that most, if not all, of the research in EMO focuses on the design and analysis of selection operators, while not much work has been done on recombination operators.

However, both these two issues are equally important. As pointed out in [10,15,25,30], many recombination operators of EMO algorithms are directly developed from single objective optimization, such as crossover and mutation, whereas the characteristics of MOPs have not been well utilized.

Under certain mild smoothness conditions, a continuous MOP has a so-called regularity property, induced from the Karush–Kuhn–Tucker conditions, that its Pareto-optimal set (PS) is a piece-wise continuous $(m-1)$ -dimensional manifold (m is the number of objectives) in the decision space [30]. That is to say, for a continuous MOP with two objectives, its PS is a piece-wise continuous one-dimensional curve and the PS of a continuous tri-objective MOP is a piecewise continuous two-dimensional surface, so on and so forth. As discussed in [10,30], although such regularity property has been utilized in the context of mathematical programming for approximating the Pareto-optimal front (PF) or PS [19,24], it has not been fully exploited in the context of EMO. The first work that explicitly exploits such regularity property is the regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA) [30]. Based on the piecewise continuous assumption, it employs the $(m-1)$ -dimensional local principal component analysis (PCA) [11] to divide the population into K (K is a user-defined parameter) disjoint clusters and identify the principal component for each of them. By making use of the solutions in each cluster, RM-MEDA builds a local model to

* Corresponding author.

E-mail address: cssamk@cityu.edu.hk (S. Kwong).

approximate a segment of the PS in the decision space. However, the major drawback of RM-MEDA just lies in its modeling technique, i.e., local PCA, which uses several linear models to approximate the nonlinear manifold. In this case, the number of local models, i.e., K , should have significant impacts on the algorithm performances [27]. But unfortunately, K is usually problem dependent.

Manifold learning, also known as nonlinear dimensionality reduction, tries to find a representation of the data, originally in a high-dimensional ambient space, in a relatively low-dimensional space, while preserving the structural information, especially the neighborhood structure. In the context of machine learning, manifold learning techniques have been widely accepted as a data preprocessing or feature extraction step, after which pattern recognition algorithms (e.g., clustering and classification) are applied. Many algorithms have been proposed to tackle this problem, such as Laplacian eigenmaps (LE) [1], principal curve (PC) [21,9], semi-definite embedding [28] and self-organizing map [29]. Since the dimensionality of the decision space is usually much larger than the number of objectives, the regularity property of a continuous MOP implies that the PS is a low-dimensional manifold embedded in the ambient decision space. Therefore, it is natural and reasonable to exploit advanced manifold learning techniques in the context of multiobjective optimization. There are some works along this direction, such as semi-definite embedding has been used for reducing the redundant objectives in many-objective optimization [23] and self-organizing map has been employed as a useful tool for visualizing the high dimensional PF [4] and generating offspring [3]. Nevertheless, the advantages of manifold learning in EMO have not been fully exploited. In this paper, we present a general framework that applies advanced manifold learning techniques to EMO in a systematic and rational manner. As a preliminary study along this direction, this paper only discusses the bi-objective continuous MOPs. At each generation, we first use the PC algorithm to project solutions, in the n -dimensional decision space, to a discrete curve that passes through their middle and provides an approximation of the PS manifold. Afterwards, we use the LE algorithm to reduce the dimensionality of these projected solutions to one, and find out the neighborhood relationship in a one-dimensional space. Correspondingly, such neighborhood relationship can be readily applicable to solutions in the original n -dimensional decision space, and it also reflects the geometric characteristic of the PS manifold. Finally, based on the neighborhood relationship, for each solution in the population, we will locate its neighboring solution and interpolate a new point as an offspring.

In the remainder of this paper, we first provide the problem formulation of MOP in Section 2. Then, we make some comments on RM-MEDA in Section 3. Afterwards, we present the technical details of our framework in Section 4. Some experiments are conducted in Section 5 to validate the effectiveness of our proposed algorithm, and finally Section 6 concludes this paper and provides some future directions.

2. Problem formulation

This paper considers the following continuous MOP:

$$\begin{aligned} &\text{minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ &\text{subject to } \mathbf{x} \in \Omega \end{aligned} \quad (1)$$

where $\Omega = \prod_{i=1}^n [a_i, b_i] \subseteq \mathbb{R}^n$ is the *decision (variable) space*, and a solution $\mathbf{x} = (x_1, \dots, x_n)^T \in \Omega$ is a vector of decision variables. $\mathbf{F}: \Omega \rightarrow \mathbb{R}^m$ constitutes m real-valued objective functions, and \mathbb{R}^m is called the *objective space*. The *attainable objective set* is defined as the set $\Theta = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in \Omega\}$. Due to the conflicting nature of MOP, only

partial ordering can be specified among solutions. In other words, for two solutions $\mathbf{x}^1, \mathbf{x}^2 \in \Omega$, it can so happen that $\mathbf{F}(\mathbf{x}^1)$ and $\mathbf{F}(\mathbf{x}^2)$ are incomparable. Some definitions related to MOP are given as follows in the context of minimization problems.¹

Definition 1. A solution \mathbf{x}^1 is said to *Pareto dominate* a solution \mathbf{x}^2 , denoted as $\mathbf{x}^1 \leq \mathbf{x}^2$, if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for every $i \in \{1, \dots, m\}$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for at least one index $j \in \{1, \dots, m\}$.

Definition 2. A solution $\mathbf{x}^* \in \Omega$ is said to be *Pareto-optimal* if there is no other solution $\mathbf{x} \in \Omega$ such that $\mathbf{x} \leq \mathbf{x}^*$.

Definition 3. The set of all Pareto-optimal solutions is called the PS. Accordingly, the set of all Pareto-optimal objective vectors, $PF = \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \mathbf{x} \in PS\}$, is called the PF.

3. Comments on RM-MEDA

Algorithm 1. Algorithm framework of RM-MEDA.

- Step 1. Initialization:** Initialize the population P_t by randomly sampling N solutions over Ω , evaluate their F -function values and set $t = 1$.
- Step 2. Stopping condition:** If the stopping condition is met, output P_t and stop.
- Step 3. Modeling:** Partition P_t into K disjoint subpopulations by $(m-1)$ -dimensional local PCA algorithm and build a probability model for each subpopulation.
- Step 4. Reproduction by sampling:** Generate a population of offspring solutions Q_t by sampling from the probability model built in **Step 3**, and add a Gaussian noise on each of them. Evaluate the F -function values of solutions in Q_t .
- Step 5. Selection:** Select N elite solutions from $P_t \cup Q_t$ to form P_{t+1} . Set $t = t + 1$ and go back to **Step 2**.

First of all, we briefly review the underlying mechanism of RM-MEDA, whose implementation is given in Algorithm 1. The principle of RM-MEDA is to build a probability model by exploiting the regularity information from the current population. New offspring solutions are thus obtained by sampling from this model. As show in Fig. 1, RM-MEDA assumes that solutions in the population are independent observations of a random vector $\xi \in \Omega$ whose centroid is the PS manifold. According to the regularity property of continuous MOP, ξ can be described as follows:

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

where ζ is uniformly distributed over a piecewise continuous $(m-1)$ -dimensional manifold, and ϵ is an n -dimensional zero-mean noise vector. In order to model the nonlinear PS manifold, by using the local PCA, RM-MEDA partitions P_t into K subpopulations, C^1, \dots, C^K , each of which is used to approximate a linear manifold (a $(m-1)$ -dimensional hyper-rectangle) Ψ^i , where $i \in \{1, \dots, K\}$. In this case, ζ is uniformly randomly sampled from Ψ^i and ϵ is a Gaussian noise $N(0, \sigma^i I)$, where I is a $n \times n$ identity matrix and $i \in \{1, \dots, K\}$. Therefore, the modeling task of RM-MEDA is transformed to estimate Ψ^i and σ^i for each subpopulation. For simplicity, we only give the formulations of Ψ^i and σ^i as follows,

¹ Due to the duality principle, maximization of a scalar objective function f_i is the same as minimization of $-f_i$, the produced results in this paper can be naturally generalized for maximization problem.

interested readers can refer to [30] for details:

$$\Psi^i = \left\{ \mathbf{x} \in \Omega | \mathbf{x} = \bar{\mathbf{x}}^i + \sum_{j=1}^{m-1} \alpha_j U_j^i, a_j - 0.25(b_j^i - a_j^i) \leq \alpha_j \leq b_j^i + 0.25(b_j^i - a_j^i) \right\} \quad (3)$$

where $i \in \{1, \dots, K\}$, $\bar{\mathbf{x}}^i = (1/|C^i|) \sum_{\mathbf{x} \in C^i} \mathbf{x}$, U_j^i is a unity eigenvector associated with the j th largest eigenvalue of the covariance matrix of solutions in C^i , $a_j^i = \min_{\mathbf{x} \in C^i} (\mathbf{x} - \bar{\mathbf{x}}^i)^T U_j^i$ and $b_j^i = \max_{\mathbf{x} \in C^i} (\mathbf{x} - \bar{\mathbf{x}}^i)^T U_j^i$:

$$\sigma^i = \frac{1}{n-m+1} \sum_{j=m}^n \lambda_j^i \quad (4)$$

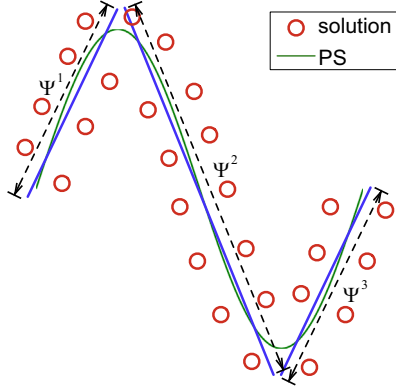


Fig. 1. Illustration of the basic idea of RM-MEDA. Solutions should be scattered around the PS in the decision space. Three manifolds Ψ^1 , Ψ^2 and Ψ^3 are used to jointly approximate the PS.

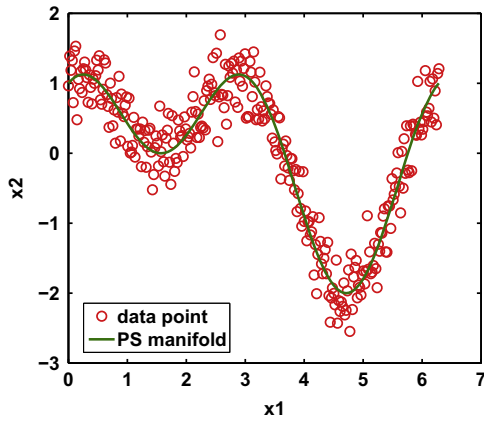


Fig. 2. An example of a nonlinear one-dimensional PS manifold.

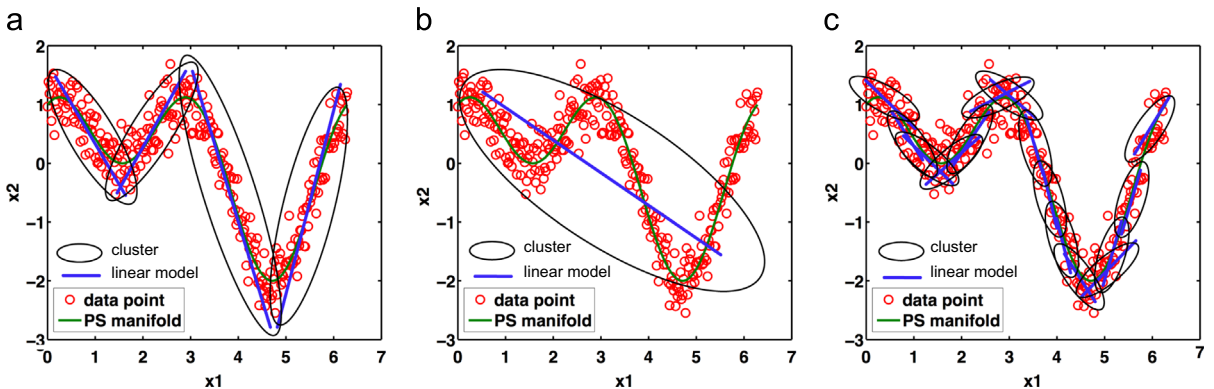


Fig. 3. Comparison of the effects of different models in different granularity: (a) nice model, (b) not enough local models and (c) too many local models.

where λ_j^i is the j th largest eigenvalue of the covariance matrix of solutions in C^i .

According to the aforementioned modeling process of RM-MEDA, one question naturally arises: how to determine the value of K ? It is set as a constant (i.e., $K=5$) in the original paper of RM-MEDA. However, different PS shapes might require different number of linear models, and even worse, the search landscapes are usually unknown a priori. Let us consider an example shown in Fig. 2, whose function form is as follows:

$$x_i = \sin(x_1) + \cos(2x_1) \quad (5)$$

where $0 \leq x_1 \leq 2\pi$, $i \in \{2, \dots, n\}$. For simplicity, we only plot the coordinates x_1 and x_2 in Fig. 2, where circles denote the solutions that scatter around the PS.

Intuitively, as for the example shown in Fig. 2, four local models (i.e., $K=4$) can effectively approximate this nonlinear manifold. Fig. 3(a) shows the clustering result of the $(m-1)$ -dimensional local PCA with $K=4$. These four local models nicely depict the geometric characteristics of this PS manifold. However, if the number of local models is not enough, it would result in a poor approximation of the PS manifold, e.g., the example shown in Fig. 3(b). In this case, the performance of RM-MEDA will be significantly degraded, and it can hardly converge to the true PS. On the other hand, if we use too many local models to approximate the PS manifold, e.g., the example shown in Fig. 3(c), it would result in a waste of computational budgets for building redundant models. Even worse, the convergence rate of RM-MEDA will be deteriorated by sampling from those redundant models.

4. Proposed framework

In this section, we present a general framework, as shown in Algorithm 2, that applies manifold learning techniques for approximating the PS manifold and facilitating offspring generation. Several remarks are given in the following paragraphs to elaborate each step of Algorithm 2.

Algorithm 2. Offspring generation via manifold learning technique.

Step 1. Use the PC algorithm to project the data points

$$S = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^N\} \text{ into a nonlinear manifold } P = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}.$$

Step 2. Use the LE algorithm to obtain the low-dimensional representations of data points in P , denoted as

$$\hat{P} = \{\tilde{\mathbf{x}}^1, \dots, \tilde{\mathbf{x}}^N\}.$$

Step 3. Determine the neighborhood relationship of each data point in \hat{P} .

Step 4. For each solution $\tilde{\mathbf{x}}^i \in P$, where $i \in \{1, \dots, N\}$, we find out its neighboring solution according to the neighborhood relationship obtained in Step 3 and interpolate a new

offspring population $Q = \{\tilde{\mathbf{x}}^1, \dots, \tilde{\mathbf{x}}^N\}$ that scattered around the approximated PS manifold.

Remark 1. PC can be regarded as a nonlinear generalization of PCA. Comparing to the linear manifold constructed by PCA, the goal of constructing a PC is to project the data points onto a nonlinear manifold directly. Fig. 4 shows the projection result obtained by a PC algorithm on the same data set used in Fig. 2. Comparing to a linear line, the projection result obtained by PCA in Fig. 3(b), the projection obtained by the PC algorithm obviously captures the geometric characteristics of the data set. Here we apply the PC algorithm proposed in [21], which redefines principal curves and surfaces in terms of the gradient and Hessian of the probability density estimate (e.g., kernel density estimation). Since the algorithm procedure is rather complicated, we do not intend to elaborate it here. Interested readers are recommended to refer to [21] for more details.

Remark 2. The projection obtained in Step 1 is an approximation of the PS manifold. In order to interpolate new solutions that obey the geometric characteristics of the manifold, we should have a good knowledge about the neighborhood relationship of solutions along the manifold. It is worth noting that the concept of neighborhood along the manifold is different from that in the Euclidean space. As shown in Fig. 5, B is the nearest neighboring point of A in the Euclidean space (as $\|AB\|^2 < \|AC\|^2$, where $\|\bullet\|^2$ indicates the ℓ_2 -norm), whereas C is the nearest neighboring point

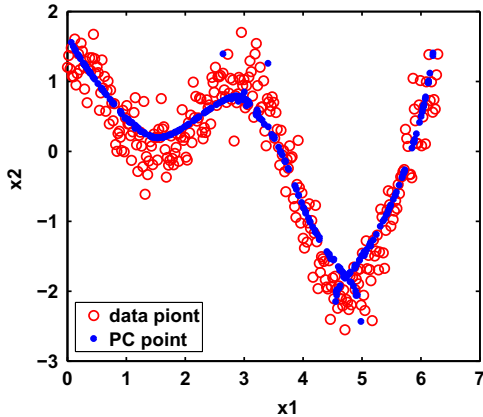


Fig. 4. Illustration of the projection result of PC algorithm on the same data set used in Fig. 2.

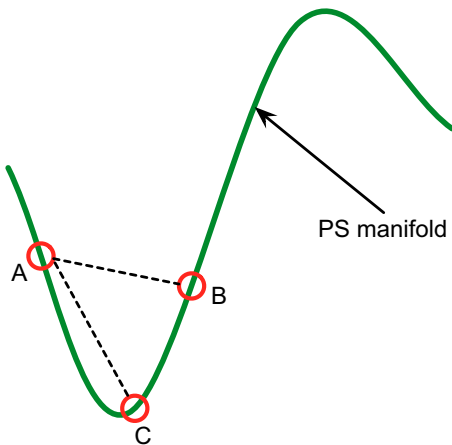


Fig. 5. Illustration of different neighboring solution concepts.

of A along this PS manifold (as $\|\widehat{AB}\|_{arc} < \|\widehat{AC}\|_{arc}$, where $\|\bullet\|_{arc}$ means the arc length). However, it is far from trivial to search for the neighboring points in the high-dimensional decision space. In order to obtain the neighborhood relationship that reflects the geometric characteristics of the approximated PS manifold, Step 2 uses a classic manifold learning technique, i.e., LE algorithm, to preprocess the data points in P . In particular, LE algorithm finds a set of points $\hat{P} = \{\hat{\mathbf{x}}^1, \dots, \hat{\mathbf{x}}^N\}$ in \mathbb{R}^{m-1} that can “represent” points of P in Ω . The basic idea of LE algorithm is to construct a weighted graph with N nodes, each of which is a point in P . A set of edges is generated to connect neighboring points. By computing the eigenvectors of the graph Laplacian, we can obtain the embedded mapping. The pseudocode of LE algorithm is shown in Algorithm 3. Interested reader can refer to [1] for more details.

Algorithm 3. Procedure of LE.

- Step 1. [Constructing the adjacency graph]** Put an edge between nodes i and j if \mathbf{x}^i and \mathbf{x}^j are close, i.e., $\|\mathbf{x}^i - \mathbf{x}^j\|^2 < \epsilon$.
Step 2. [Choosing the weights] If nodes i and j are connected, put $W_{ij} = e^{-\|\mathbf{x}^i - \mathbf{x}^j\|^2}$, otherwise put $W_{ij} = 0$.
Step 3. [Eigenmaps] Compute eigenvalues and eigenvectors for the generalized eigenvector problem:

$$L\mathbf{f} = \lambda D\mathbf{f} \quad (6)$$

where D is diagonal weight matrix, its entries are column (or row, since W is symmetric) sums of w , $D_{ii} = \sum_j W_{ji}$. $L = D - W$ is the Laplacian matrix. Let $\mathbf{f}^0, \dots, \mathbf{f}^{n-1}$ be the solutions of equation (6), ordered according to their eigenvalues

$$\begin{aligned} L\mathbf{f}^0 &= \lambda^0 D\mathbf{f}^0 \\ L\mathbf{f}^1 &= \lambda^1 D\mathbf{f}^1 \\ &\dots \\ L\mathbf{f}^{n-1} &= \lambda^{n-1} D\mathbf{f}^{n-1} \\ 0 &= \lambda^0 \leq \lambda^1 \leq \dots \leq \lambda^{n-1} \end{aligned}$$

Leave out the eigenvector \mathbf{f}^0 corresponding to eigenvalue 0 and use the next $m-1$ eigenvalues for embedding in $(m-1)$ -dimensional Euclidean space, i.e., $\mathbf{x}^i \Rightarrow \hat{\mathbf{x}}^i = (f_i^1, \dots, f_i^{m-1})^T$.

Remark 3. Since we only consider the bi-objective MOPs here, according to the regularity property of continuous MOPs, the dimensionality of solutions in \hat{P} is one after the preprocess of the LE algorithm. Therefore, the neighborhood relationship of solutions in \hat{P} can be obtained by a simple sorting procedure. Due to the locality preserving characteristic of the LE algorithm [1], we assume that the neighborhood relationship obtained in \hat{P} can be directly applied to P , while reflecting the geometric characteristics of the manifold. To validate this assumption, we apply the LE algorithm to the data points as shown on the left-hand side of Fig. 6, and obtain the low-dimensional representation as shown on the right-hand side of Fig. 6. The green and red circles highlight two neighboring points in the low-dimensional representation. Correspondingly, we find that they are also the neighboring solutions in the original data set along the manifold.

Remark 4. Based on the neighborhood relationship obtained in Step 3, the task of Step 4 is to interpolate new solutions that conform to the geometric characteristics of the approximated PS manifold. For simplicity, we try to interpolate new solutions between two neighboring solutions. Specifically, for a solution \mathbf{x}^i , $i \in \{1, \dots, N\}$, let its neighboring solution be \mathbf{x}^r , a new solution is

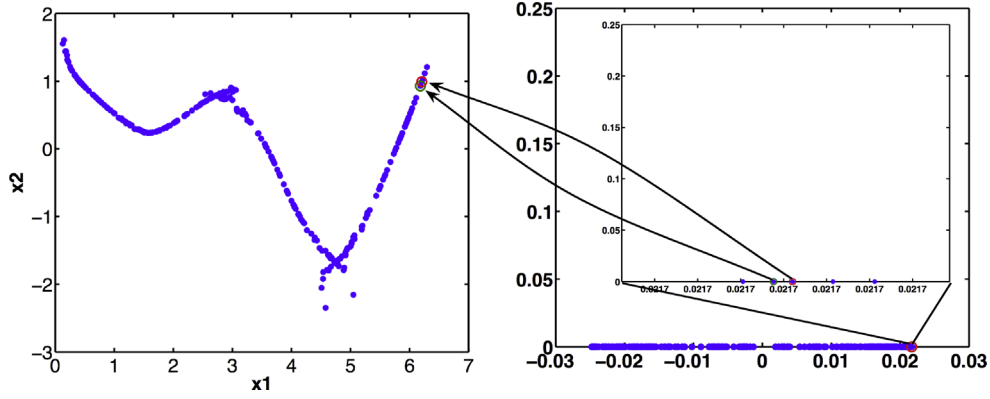


Fig. 6. Illustration of locality preserving characteristic of LE algorithm. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

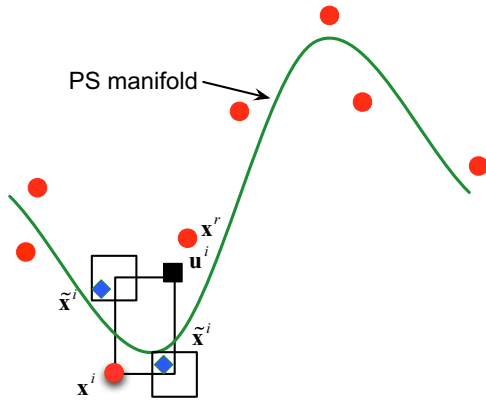


Fig. 7. Illustration on generating offspring solutions. In principle, a new offspring solution \tilde{x}^i can be any point that scattered around the PS manifold.

generated as

$$u_j^i = \begin{cases} x_j^i + F \times (x_j^r - x_j^i) & \text{if } rand < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \quad (7)$$

where $j \in \{1, \dots, n\}$, $rand \in [0, 1]$, $F \in [-2, 3]$ and j_{rand} is a random integer uniformly chosen from 1 to n . Then, the polynomial mutation [7] is applied on each u^i to obtain \tilde{x}^i :

$$\tilde{x}_j^i = \begin{cases} u_j^i + \sigma_j \times (b_j - a_j) & \text{if } rand < p_m \\ u_j^i & \text{otherwise} \end{cases} \quad (8)$$

with

$$\sigma_j = \begin{cases} (2 \times rand)^{1/(\eta+1)} - 1 & \text{if } rand < 0.5 \\ 1 - (2 - 2 \times rand)^{1/(\eta+1)} & \text{otherwise} \end{cases} \quad (9)$$

where the distribution index η and the mutation rate p_m are two control parameters. a_j and b_j are the lower and upper bounds of the j th decision variable. For simplicity, the violated decision variable is set to its nearer boundary value. Fig. 7 presents an illustration on generating new solutions.

5. Empirical studies

In principle, the proposed framework can be incorporated into any EMO algorithms, by replacing their recombination operators. As a preliminary study, we replace **Step 3** and **Step 4** of **Algorithm 1** with our proposed modeling technique in **Algorithm 2**. Moreover, we use the selection operator proposed in [18] in **Step 5** of **Algorithm 1**. This section presents the performance comparisons

of the resulted algorithm, termed as RM-MEDA/ML with RM-MEDA.

5.1. Test instances

Ten unconstrained bi-objective MOP test instances are used in our empirical studies, including F1, F2, F3, F5, F6 from [30] and UF1, UF2, UF3, UF4, UF7 from CEC2009 MOEA competition [31]. All these test instances are with variable linkages, and UF-series test instances are with complicated PSs in the decision space. The number of decision variables is constantly set to 10 for all test instances.

5.2. Performance metrics

No unary performance metric can give a comprehensive assessment on the performance of an EMO algorithm [34]. In our empirical studies, we employ the following two widely used performance metrics.

1. *Inverted Generational Distance* (IGD) metric [2]: Let P^* be a set of points uniformly sampled along the PF, and S be the set of solutions obtained by an EMO algorithm. The IGD value of S is calculated as

$$IGD(S, P^*) = \frac{\sum_{x \in P^*} \text{dist}(x, S)}{|P^*|} \quad (10)$$

where $\text{dist}(x, S)$ is the Euclidean distance between the solution x and its nearest point in S , and $|P^*|$ is the cardinality of P^* . The PF of the underlying MOP is assumed to be known a priori when using the IGD metric. In our empirical studies, 1000 uniformly distributed points are sampled along the PF for IGD calculation.

2. *Hypervolume* (HV) metric [33]: Let $z^r = (z_1^r, \dots, z_m^r)^T$ be a reference point in the objective space that is dominated by all Pareto-optimal objective vectors. HV metric measures the size of the objective space dominated by the solutions in S and bounded by z^r :

$$HV(S) = \text{Vol} \left(\bigcup_{x \in S} [f_1(x), z_1^r] \times \dots [f_m(x), z_m^r] \right) \quad (11)$$

where $\text{Vol}(\cdot)$ is the Lebesgue measure. In our empirical studies, the reference point is constantly set as $z^r = (2.0, 2.0)^T$ for all test instances.

Both IGD and HV metrics can measure the convergence and diversity of S . The lower is the IGD value (the larger is the HV

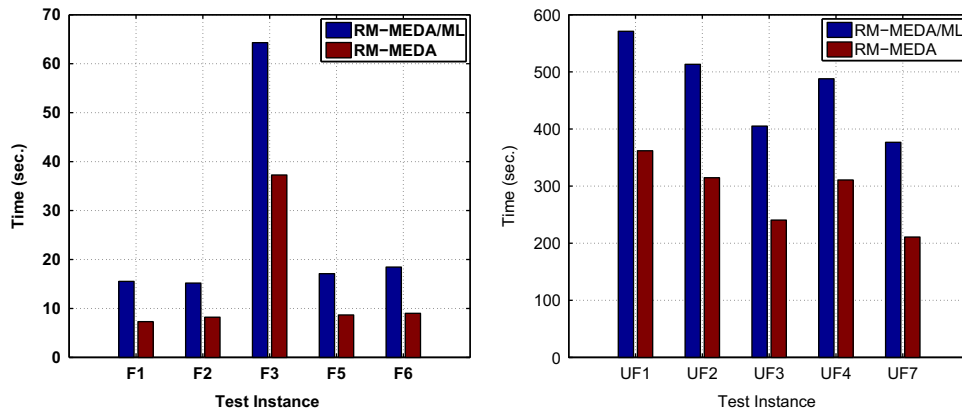


Fig. 8. Average CPU-time costs comparisons of RM-MEDA/ML and RM-MEDA.

Table 1

Performance comparisons on IGD and HV metrics.

Test instances	IGD		HV	
	RM-MEDA	RM-MEDA/ML	RM-MEDA	RM-MEDA/ML
F1	5.982E−3(1.38E−3) ^a	3.882E−3(3.20E−6)	3.6491(1.05E−2) ^a	3.6613(5.48E−5)
F2	6.803E−3(3.41E−3) ^a	3.819E−3(5.01E−9)	3.2882(5.06E−2) ^a	3.3282(6.04E−5)
F3	7.426E−3(4.25E−3) ^a	7.099E−2(1.54E−2)	3.0246(1.30E−2) ^a	3.0599(7.89E−2)
F5	1.452E−2(1.03E−2) ^a	5.158E−3(1.28E−4)	3.5996(4.73E−2) ^a	3.6581(2.42E−4)
F6	1.870E−1(1.18E−1) ^a	5.084E−3(2.92E−4)	2.5068(3.41E−1) ^a	3.3242(7.22E−4)
UF1	4.583E−2(1.34E−2) ^a	6.770E−3(6.65E−4)	3.5295(3.74E−2) ^a	3.6476(6.08E−3)
UF2	2.357E−2(8.71E−3) ^a	9.472E−3(1.28E−3)	3.5720(4.97E−2) ^a	3.6450(1.03E−2)
UF3	1.103E−1(5.22E−2) ^a	4.289E−3(5.18E−4)	3.2802(2.08E−1) ^a	3.6592(1.81E−3)
UF4	8.687E−2(3.04E−3) ^a	6.401E−2(8.65E−2)	3.0717(1.13E−2) ^a	3.4510(6.76E−1)
UF7	1.560E−2(2.24E−3) ^a	4.642E−3(2.54E−4)	3.4494(1.69E−2) ^a	3.4902(2.07E−3)

Wilcoxon's rank sum test at a 0.05 significance level is performed between RM-MEDA/ML and RM-MEDA. The best mean is highlighted in boldface. The performance of RM-MEDA is significantly better than that of RM-MEDA/ML.

^a The performance of RM-MEDA is significantly worse than that of RM-MEDA/ML.

value), the better is the quality of S for approximating the whole PF. In the comparison tables of the following paragraphs, the best mean metric values are highlighted in bold face. In order to have statistically sound conclusions, Wilcoxon's rank sum test at a 5% significance level is conducted to compare the significance of difference between the metric values of two algorithms.

5.3. General parameter settings

All these algorithms are implemented in MATLAB.² The parameters of RM-MEDA are set the same as in [30], while the parameter settings of our proposed RM-MEDA/ML are summarized as follows:

1. *Parameters of polynomial mutation*: The mutation probability $p_m = 1/n$ and its distribution index $\eta_m = 20$ [7].
2. *Population size*: $N = 100$ for all test instances.
3. *Number of runs and stopping condition*: Each algorithm is run 20 times independently on each test instance. The algorithm stops after 10,000 function evaluations for F1, F2, F5 and F6. 50,000 for F3. 300,000 for UF1 to UF4 and UF7.

² The source code of RM-MEDA is downloaded from <http://cswwww.essex.ac.uk/staff/zhang/>. As for RM-MEDA/ML, the implementation of PC is downloaded from <http://indigo.ece.neu.edu/erdogmus/pubs.html>, and the implementation of LE is downloaded from <http://www.cse.ohio-state.edu/mbelkin/algorithms/algorithms.html>.

4. *Parameters of manifold learners*: The parameters of PC and LE algorithms used in RM-MEDA/ML are set the same as recommended in their original papers [21,1].

5.4. Empirical results

The performance comparisons of RM-MEDA and RM-MEDA/ML, in terms of IGD and HV metrics, are presented in Table 1. Figs. 9–18 plot the non-dominated fronts, including the one obtained in the run with the best IGD value and all 20 fronts together, by RM-MEDA and RM-MEDA/ML, respectively, on each test instance. From these empirical results, it is clear that our proposed RM-MEDA/ML performs better than the original RM-MEDA. It achieves the better metric values in all comparisons. Wilcoxon's rank sum tests indicate that all these better results achieved by RM-MEDA/ML are with statistical significance.

To be specific, F1 to F3 are with linear variable linkages. The performances of both algorithms are similar on F1 and F2, while the performance of RM-MEDA/ML is more consistent than RM-MEDA in all 20 runs. F3 has a non-uniform mapping from the decision space to the objective space. In view of all 20 runs, the performance of RM-MEDA is more consistent than RM-MEDA/ML this time. But from Fig. 11, we find that the best non-dominated front obtained by RM-MEDA/ML is closer to the PF. F5 and F6 are with nonlinear variable linkages. RM-MEDA and RM-MEDA/ML have shown similar performances on F5. But from Fig. 12, it is clear that RM-MEDA/ML performs more consistently than RM-MEDA in view of all 20 runs. As for F6, the superiority of RM-MEDA/ML is

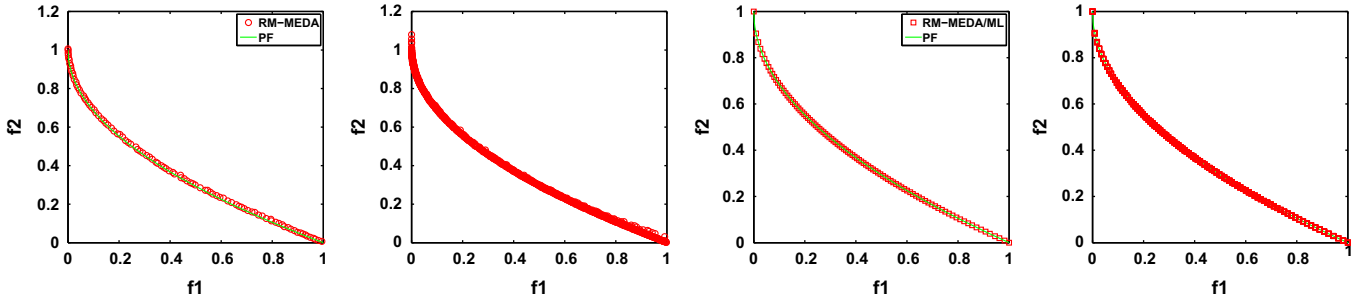


Fig. 9. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on F1.

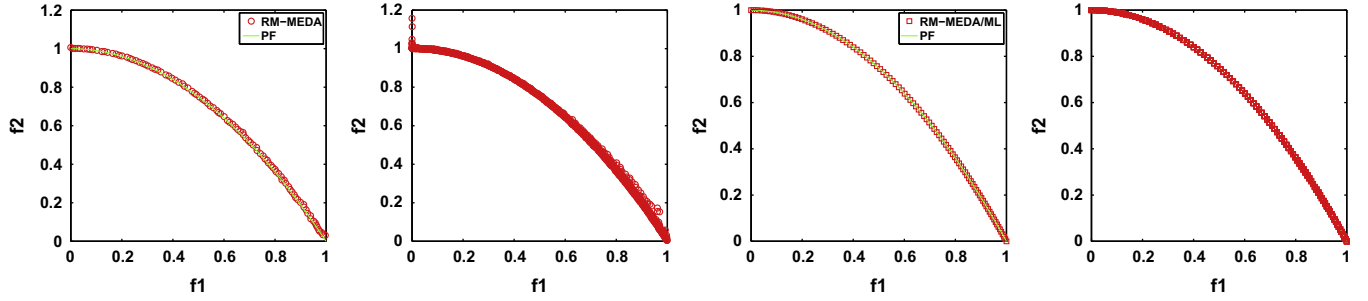


Fig. 10. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on F2.

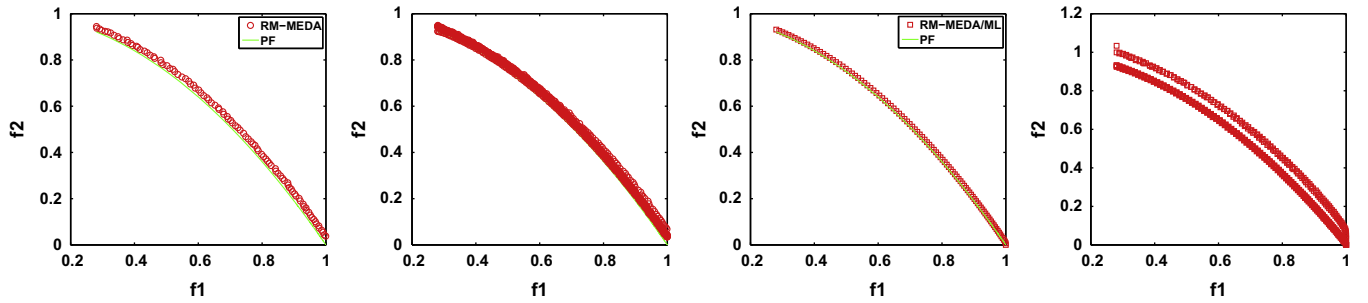


Fig. 11. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on F3.

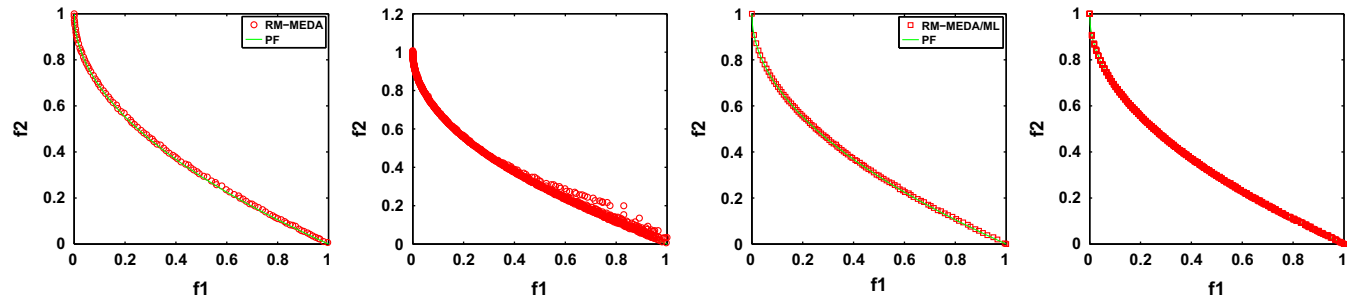


Fig. 12. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on F5.

very clear, as RM-MEDA cannot approximate the entire PF in all 20 runs. Although these F-series test instances have variable linkages, their PSs are usually linear manifolds in the decision space. In this case, the modeling technique of RM-MEDA still can use several linear models to approximate the entire PS. This explains the similar performance of RM-MEDA and RM-MEDA/ML on F-series test instances. But for UF-series test instances, they not only have nonlinear variable linkages, but also have complicated PSs in the decision space. As discussed in Section 3, it is very difficult to determine how many local linear models can approximate those

complicated PSs. Therefore, RM-MEDA can easily build a wrong model which greatly deviates from the PS. Obviously, new off-spring solutions sampled from such wrong model are usually inferior or invalid candidates, which may greatly mislead the search process. In contrast, our proposed RM-MEDA/ML applies the advanced manifold learning techniques to progressively learn the nonlinear and complicated regularity property of the MOP in question. This facilitates the model building for nonlinear PS manifold. From the empirical results, it is obvious that our proposed RM-MEDA/ML performs much better than RM-MEDA

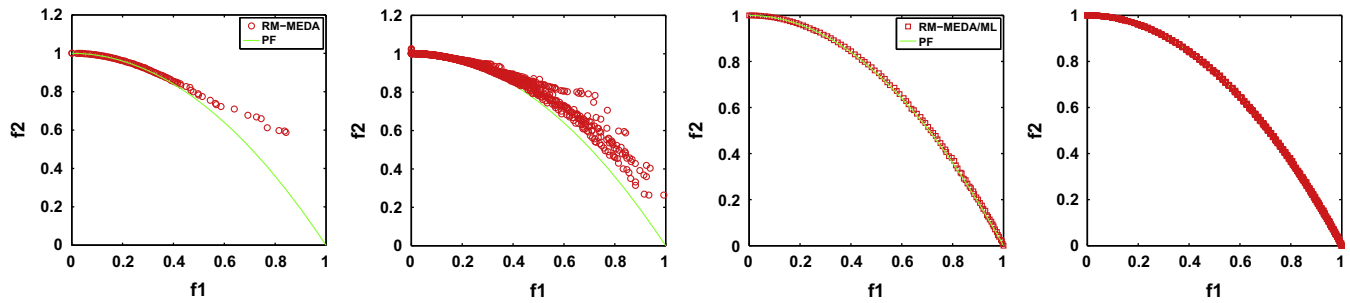


Fig. 13. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on F6.

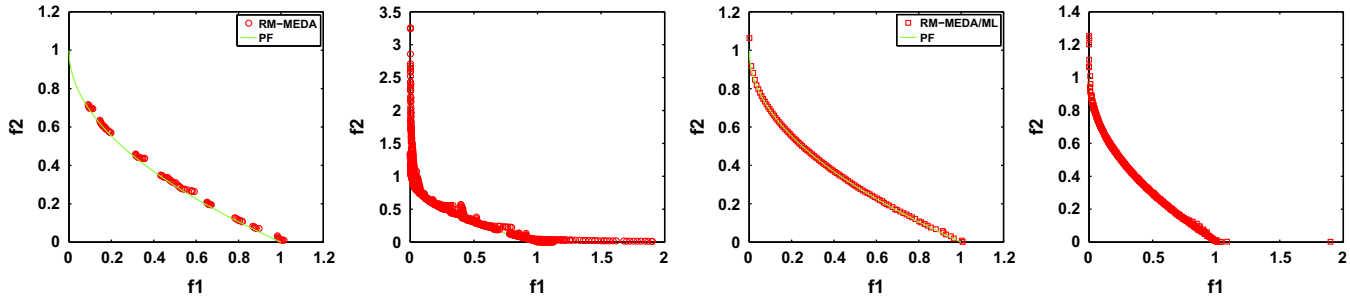


Fig. 14. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on UF1.

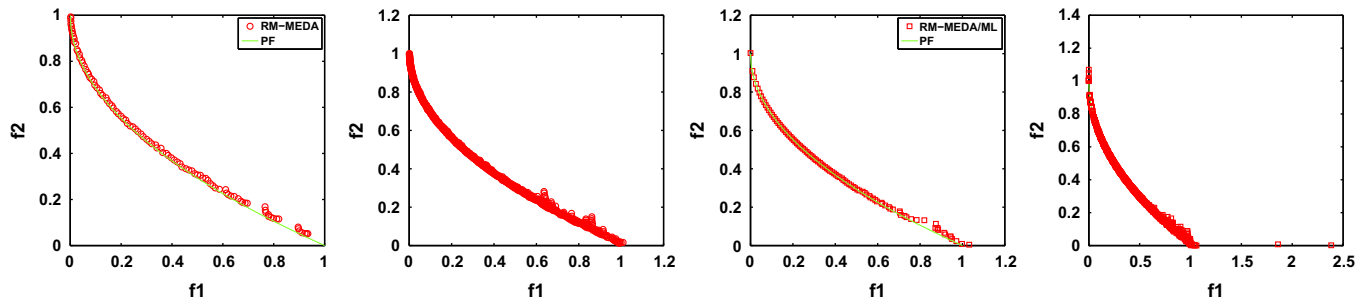


Fig. 15. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on UF2.

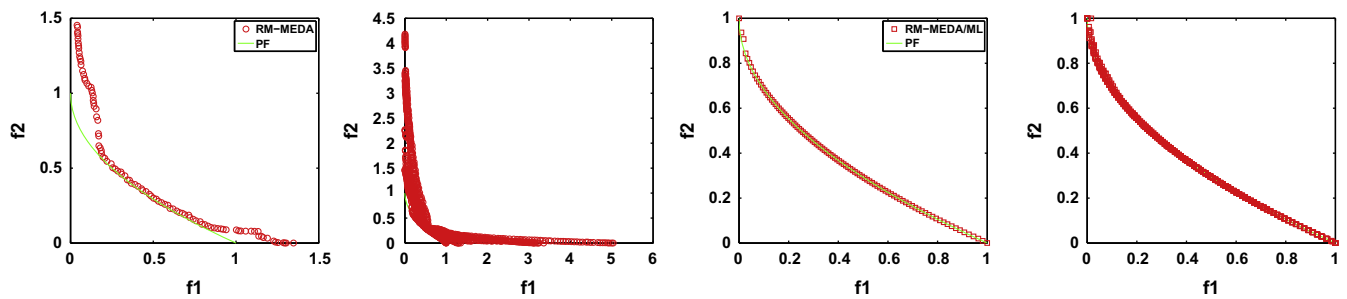


Fig. 16. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on UF3.

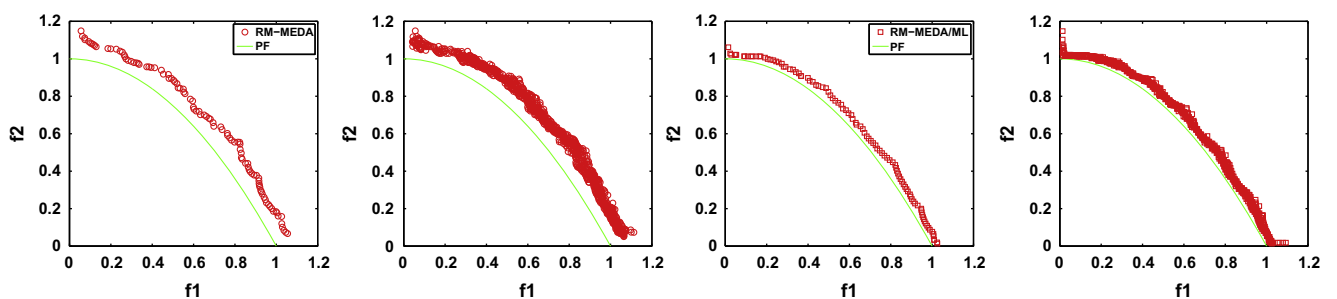


Fig. 17. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on UF4.

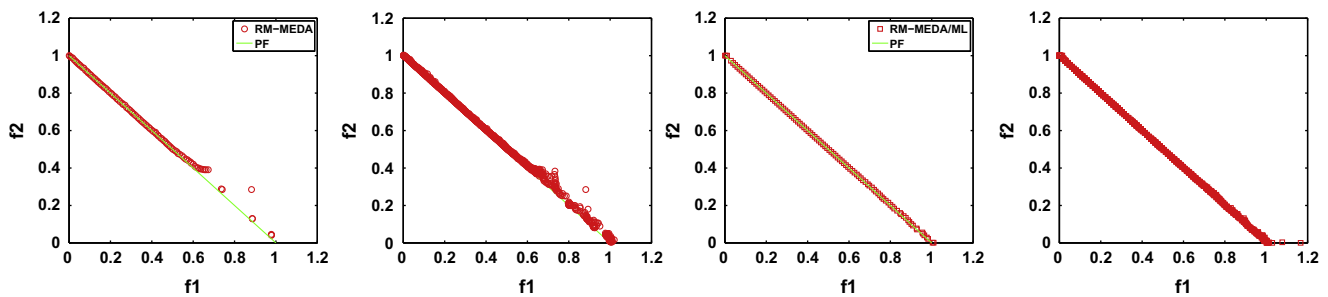


Fig. 18. Plots of solutions obtained by RM-MEDA and RM-MEDA/ML on UF7.

in terms of both convergence and diversity, while RM-MEDA can approximate only some limited regions of the PF.

However, the encouraging performance achieved by RM-MEDA/ML is no free lunch. Fig. 8 presents the comparisons of average CPU-time costs of RM-MEDA/ML and RM-MEDA on all test instances. It is obvious that the manifold learning techniques, used in RM-MEDA/ML, cost more time on model building than the local PCA used in RM-MEDA.

6. Conclusion and future works

This paper presents a general framework for using advanced manifold learning techniques in EMO. In classic EMO algorithms, recombination operators are usually developed from the single objective optimization. In contrast, our proposed framework provides an avenue to progressively learn the regularity property of the MOP in question during the search process. The proposed recombination operator is tailored for continuous MOPs, especially those with complicated PSs in the decision space. From our empirical studies, it is clear that our proposed RM-MEDA/ML shows promising performance on all benchmark problems, especially the UF-series test instances, which are with complicated PSs.

As mentioned in the outset of this paper, this is a very preliminary study on applying advanced machine learning techniques (manifold learning in this paper) to the context of EMO. The major purpose of this work is to draw significant attentions, from the community, that the importance and usefulness of bridging the gap between machine learning and evolutionary computation. Many follow-up works can be done along this direction.

1. As a preliminary study, this work only addresses the bi-objective continuous MOPs. As for problems with three or more objectives, the PS manifold should be a surface or hypersurface, which makes the interpolation of new solutions become more difficult. One may consider the Delaunay triangulation [8] to obtain a triangulation of the points projected by the PC algorithm. Then, new offspring solutions can be sampled within each simplex.
2. As discussed in [32], the regularity property is not only applicable for PS, but also for PF in the objective space. That is to say, the PF is also a $(m-1)$ -dimensional manifold embedded in the m -dimensional objective space. In this case, it is interesting to apply our proposed framework to model the PF manifold and to guide the selection of solutions for the next generation.
3. Other than the PC and LE algorithms used in this paper, there are many other manifold learning techniques in the machine learning field. It is interesting and important to compare and investigate merits and drawbacks of different manifold learning techniques for different MOPs with various PS manifolds.

4. In this paper, we only consider the continuous MOPs, in which the PSs are also continuous manifolds. It is also interesting and important to extend our idea for problems whose PS manifolds are several disconnected segments.

References

- [1] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.* 15 (6) (2003) 1373–1396.
- [2] P. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, *IEEE Trans. Evol. Comput.* 7 (April (2)) (2003) 174–188.
- [3] D. Büche, M. Milano, P. Koumoutsakos, Self-organizing maps for multi-objective optimization, in: GECCO'02: Proceedings of the 2002 Annual Conference on Genetic and Evolutionary Computation, 2002, pp. 152–155.
- [4] S. Chen, D. Amid, O. Shir, L. Limonad, D. Boaz, A. Anaby-Tavor, T. Schreck, Self-organizing maps for multi-objective Pareto frontiers, in: PacificVis'13: Proceedings of the 2013 IEEE Pacific Visualization Symposium, 2013, pp. 153–160.
- [5] C.A. Coello Coello, G.B. Lamont, D.A.V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Genetic and Evolutionary Computation, 2nd ed., Springer, New York, 2007.
- [6] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Inc., New York, NY, USA, 2001.
- [7] K. Deb, M. Goyal, A combined genetic adaptive search (geneas) for engineering design, *Comput. Sci. Inform.* 26 (4) (1996) 30–45.
- [8] L. Guibas, J. Stolfi, Primitives for the manipulation of general subdivisions and the computation of Voronoi, *ACM Trans. Graph.* 4 (April (2)) (1985) 74–123.
- [9] T. Hastie, W. Stuetzle, Principal curves, *J. Am. Stat. Assoc.* 84 (406) (1989) 502–516.
- [10] Y. Jin, B. Sendhoff, Connectedness, regularity and the success of local search in evolutionary multi-objective optimization, in: CEC'03: Proceedings of the 2003 IEEE Congress on Evolutionary Computation, 2003, pp. 1910–1917.
- [11] N. Kambhatla, T.K. Leen, Dimension reduction by local principal component analysis, *Neural Comput.* 9 (7) (1997) 1493–1516.
- [12] K. Li, Á. Fialho, S. Kwong, Multi-objective differential evolution with adaptive control of parameters and operators, in: C.A.C. Coello (Ed.), LION'11: Proceedings of the 5th International Conference Learning and Intelligent Optimization, Springer, Lecture Notes in Computer Science, January 2011, pp. 473–487.
- [13] K. Li, Á. Fialho, S. Kwong, Q. Zhang, Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.* 18 (1) (2014) 114–130.
- [14] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, R. Shen, Achieving balance between proximity and diversity in multi-objective evolutionary algorithm, *Inf. Sci.* 182 (1) (2012) 220–242.
- [15] K. Li, S. Kwong, R. Wang, J. Cao, I.J. Rudas, Multi-objective differential evolution with self-navigation, in: SMC'12: Proceedings of the 2012 IEEE International Conference on Systems, Man, and Cybernetics, IEEE, Seoul, Korea (South), October 2012, pp. 508–513.
- [16] K. Li, S. Kwong, R. Wang, K.-S. Tang, K.-F. Man, Learning paradigm based on jumping genes: a general framework for enhancing exploration in evolutionary multiobjective optimization, *Inf. Sci.* 226 (2013) 1–22.
- [17] K. Li, R. Wang, S. Kwong, J. Cao, Evolving extreme learning machine paradigm with adaptive operator selection and parameter control, *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.* 21 (2013) 143–154.
- [18] K. Li, Q. Zhang, S. Kwong, M. Li, R. Wang, Stable matching based selection in evolutionary multiobjective optimization, *IEEE Trans. Evol. Comput.* (2014), <http://dx.doi.org/10.1109/TEVC.2013.2293776>.
- [19] K. Miettinen, *Nonlinear Multiobjective Optimization*, vol. 12, Kluwer Academic Publisher, Boston, 1999.
- [20] A.A. Mohtano, C.A.C. Coello, E. Mezura-Montes, Multiobjective evolutionary algorithms in aeronautical and aerospace engineering, *IEEE Trans. Evol. Comput.* 16 (August (5)) (2012) 662–694.
- [21] U. Ozertem, D. Erdogmus, Locally defined principal curves and surfaces, *J. Mach. Learn. Res.* 12 (July) (2011) 1249–1286.

- [22] H. Rajabali pour Cheshmehgaz, M. Desa, A. Wibowo, A flexible three-level logistic network design considering cost and time criteria with a multi-objective evolutionary algorithm, *J. Intell. Manuf.* (2011) 1–17.
- [23] D.K. Saxena, J.A. Duro, A. Tiwari, K. Deb, Q. Zhang, Objective reduction in many-objective optimization: linear and nonlinear algorithms, *IEEE Trans. Evol. Comput.* 17 (1) (2013) 77–99.
- [24] O. Schütze, S. Mostaghim, M. Dellnitz, J. Teich, Covering Pareto sets by multilevel evolutionary subdivision techniques, in: EMO'03: Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, 2003, pp. 118–132.
- [25] K. Sindhya, S. Ruuska, T. Haanpää, K. Miettinen, A new hybrid mutation operator for multiobjective optimization with differential evolution, *Soft Comput.* 15 (10) (2011) 2041–2055.
- [26] M.G.C. Tapia, C.A.C. Coello, Applications of multi-objective evolutionary algorithms in economics and finance: a survey, in: CEC'07: Proceedings 2007 IEEE Congress on Evolutionary Computation, Singapore, September 2007, pp. 532–539.
- [27] Y. Wang, J. Xiang, Z. Cai, A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator, *Appl. Soft Comput.* 12 (11) (2012) 3526–3538.
- [28] K.Q. Weinberger, L.K. Saul, Unsupervised learning of image manifolds by semidefinite programming, *Int. J. Comput. Vis.* 70 (2006) 77–90.
- [29] H. Yin, Learning nonlinear principal manifolds by self-organising maps, in: A. Gorban, B. Kégl, D. Wunsch, A. Zinovyev (Eds.), *Principal Manifolds for Data Visualization and Dimension Reduction, Lecture Notes in Computational Science and Engineering*, vol. 58, Springer, Berlin, Heidelberg, 2008, pp. 68–95.
- [30] Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm, *IEEE Trans. Evol. Comput.* 12 (1) (2008) 41–63.
- [31] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, Technical Report CES-487, University of Essex and Nanyang Technological University, 2008.
- [32] A. Zhou, Q. Zhang, Y. Jin, B. Sendhoff, Combination of EDA and DE for continuous biobjective optimization, in: CEC'08: Proceedings of the 2008 IEEE Congress on Evolutionary Computation, 2008, pp. 1447–1454.
- [33] E. Zitzler, L. Thiele, Multiobjective Evolutionary Algorithms: A comparative case study and the strength pareto approach, *IEEE Trans. Evol. Comput.* 3 (4) (1999) 257–271.
- [34] E. Zitzler, L. Thiele, M. Laumanns, Performance assessment of multiobjective optimizers: an analysis and review, *IEEE Trans. Evol. Comput.* 7 (April (2)) (2003) 117–132.



Ke Li was born in Hunan, China, in 1985. He received the B.Sc. and M.Sc. degrees in computer science and technology from Xiangtan University, Hunan, China, in 2007 and 2010, respectively, and the Ph.D. in computer science from City University of Hong Kong, Kowloon, Hong Kong, in 2014. His research interests include evolutionary multiobjective optimization, surrogate-assisted evolutionary algorithms, and statistical machine learning techniques.



Sam Kwong received the B.S. and M.S. degrees in electrical engineering from the State University of New York at Buffalo in 1983, the University of Waterloo, Waterloo, ON, Canada, in 1985, and the Ph.D. degree from the University of Hagen, Germany, in 1996. From 1985 to 1987, he was a Diagnostic Engineer with Control Data, Canada. He joined Bell Northern Research, Canada, as a Member of Scientific Staff. In 1990, he became a Lecturer in the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor and Head in the Department of Computer Science. His main research interests include evolutionary computation and video coding. Dr. Kwong is an Associate Editor of IEEE Transactions on Industrial Electronics, IEEE Transactions on Industrial Informatics and Information Sciences Journal.