Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

A Multiobjective Evolutionary Algorithm using Gaussian Process based Inverse Modeling

Ran Cheng, Yaochu Jin, Kaname Narukawa

2015

Preprint:

This is an accepted article published in IEEE Transactions on Evolutionary Computation. The final authenticated version is available online at: https://doi.org/[DOI not available]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

A Multiobjective Evolutionary Algorithm using Gaussian Process based Inverse Modeling

Ran Cheng, Yaochu Jin, *Senior Member, IEEE*, Kaname Narukawa and Bernhard Sendhoff, *Senior Member, IEEE*

***Abstract*—To approximate the Pareto front, most existing multiobjective evolutionary algorithms store the non-dominated solutions found so far in the population or in an external archive during the search. Such algorithms often require a high degree of diversity of the stored solutions and only a limited number of solutions can be achieved. By contrast, model-based algorithms can alleviate the requirement on solution diversity and in principle, as many solutions as needed can be generated. This paper proposes a new model-based method for representing and searching non-dominated solutions. The main idea is to construct Gaussian process based inverse models that map all found non-dominated solutions from the objective space to the decision space. These inverse models are then used to create offspring by sampling the objective space. To facilitate inverse modeling, the multivariate inverse function is decomposed into a group of univariate functions, where the number of inverse models is reduced using a random grouping technique. Extensive empirical simulations demonstrate that the proposed algorithm exhibits robust search performance on a variety of medium to high dimensional multiobjective optimization test problems. Additional non-dominated solutions are generated *a posteriori* using the constructed models to increase the density of solutions in the preferred regions at a low computational cost.**

***Index Terms*—Multiobjective optimization, estimation of distribution algorithms, inverse modeling, random grouping, Gaussian processes**

I. Introduction

MOST real-world optimization problems involve multiple optimization criteria (objectives), which often conflict with each other. Such optimization problems are known as multiobjective optimization problems (MOPs), which can be formulated as follows:

$$\begin{aligned} \min \quad & \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), ..., f_m(\vec{x})) \\ \text{s.t.} \quad & \vec{x} \in X, \quad \vec{f} \in Y \end{aligned} \tag{1}$$

where $X \subset \mathbb{R}^n$ is the decision space and $\vec{x} = (x_1, x_2, ..., x_n) \in X$ is the decision vector, $Y \subset \mathbb{R}^m$ is the objective space and $\vec{f} \in Y$ is the objective vector, which is composed of m objective functions $f_1(\vec{x})$, $f_2(\vec{x})$,...,$f_m(\vec{x})$ that map $\vec{x}$ from X to Y.

Due to the conflicts between the objectives, i.e., improvement of one objective results in deterioration of another, no single solution is able to optimize all objectives at the same time. As a consequence, there exist a set of optimal solutions that trade off between different objectives. In multiobjective optimization (MOO), a solution $\vec{x}_1$ is said to dominate another solution $\vec{x}_2$ (denoted as $\vec{x}_1 \succ \vec{x}_2$) iff

$$\begin{cases} \forall i \in 1, 2, ..., m : f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \\ \exists j \in 1, 2, ..., m : f_j(\vec{x}_1) < f_j(\vec{x}_2). \end{cases} \tag{2}$$

If a solution $\vec{x}^*$ cannot be dominated by any other feasible solutions, $\vec{x}^*$ is said to be *Pareto optimal* and the union of all $\vec{x}^*$ is called Pareto set (PS), while the union of $\vec{f}(\vec{x}^*)$ is termed Pareto front (PF). For most continuous MOPs, the Pareto optimal solution set often consists of an infinite number of solutions. Therefore, in practice, only a representative subset of the Pareto optimal solutions can be achieved. Note, however that in many cases, the non-dominated solution set obtained by a multiobjective optimization algorithm may not necessarily be Pareto optimal.

Evolutionary algorithms (EAs) are well suited for solving MOPs as they can obtain a solution set in one single run [1]. Since Schaffer put forward the vector evaluated genetic algorithm [2] in 1984, a large number of multiobjective evolutionary algorithms (MOEAs) have been proposed. They can be in general categorized into three groups, including weighted aggregation based methods, dominance based methods and performance indicator based approaches. Weighted aggregation based methods are now also known as decomposition approaches, where an MOP is decomposed into a number of single objective optimization problems using a number of weight combinations that are either randomly generated [3], [4], dynamically and continuously changed [4], [5], or predefined and evenly distributed [6]. One most recent variant of weighted aggregation algorithms, termed multiobjective evolutionary algorithm based on decomposition (MOEA/D) [7] has received increasing attention [8]–[10] due to its computational simplicity and attractive search performance.

Dominance based MOEAs have been prevalent over the past two decades. Early dominance based non-elitism MOEAs include the non-dominated sorting genetic algorithm [11], the niched-Pareto genetic algorithm [12], and the multiobjective genetic algorithm [13]. It was found out later on that use of elitism strategies can substantially improve the convergence of MOEAs. As a result, several popular elitism MOEAs have been developed, including the elitist non-dominated sorting genetic algorithm (NSGA-II) [14], the strength Pareto evolutionary algorithm [15], [16] and the Pareto envelope-based selection algorithm [17], [18], among many others. Although

R. Cheng and Y. Jin are with the Department of Computing, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom (e-mail: {r.cheng;yaochu.jin}@surrey.ac.uk). Y. Jin is also with the College of Information Sciences and Technology, Donghua University, Shanghai 201620, P. R. China.

Kaname Narukawa and Bernhard Sendhoff are with the Honda Research Institute Europe, 63073 Offenbach, Germany (e-mail: {kaname.narukawa;bernhard.sendhoff}@honda-ri.de).

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

dominance based MOEAs are very powerful for solving bi-objective or three-objective MOPs, their convergence performance dramatically degrades when the number of objectives is larger than three, mainly due to the loss selection pressure.

One different yet interesting idea is to assign a fitness to individuals of an MOEA based on a performance indicator [19]. Among various performance indicator based MOEAs, hypervolume [20], which is known to be able to account for both accuracy and diversity of non-dominated solution sets, has most widely been adopted [21]–[24]. The main challenge of this class of MOEAs is the high computational complexity for calculating the performance indicator, especially when the number of objectives is large [25]–[27].

Most MOEAs mentioned above focus on the development of an effective fitness calculation or selection strategy when adapting single-objective EAs to solving MOEAs. Not much attention has been paid to designing effective reproduction strategies that explicitly exploit the connectedness and regularity in the distribution of Pareto optimal solutions [28]. However, many local search strategies, such as the dynamic weighted aggregation method [5], restricted mating strategy [29] and many others [3], [30], [31] have implicitly taken advantage of the connectedness property to enhance the search efficiency. Another class of MOEAs that are believed to be able to learn problem structures is the estimation of distribution algorithms (EDAs) [32]–[34]. Instead of using traditional reproduction operators such as crossover and mutation, EDAs build probabilistic models to estimate the distribution of promising candidate solutions, and then new candidate solutions are generated by sampling the models.

Whereas many multiobjective EDAs are directly adapted from single objective EDAs [35]–[37], several algorithms have also been proposed to take the distinctive properties of MOPs into considerations. For example, a Voronoi-based estimation of distribution algorithm for multiobjective optimization was suggested to make use of all the candidate solutions in different fronts [38], while in [39], [40], a mixture distribution based multiobjective EDA was proposed to preserve population diversity. Recently, a multiobjective EDA based on joint modeling of objectives and variables [41] was reported, which is able to capture the dependencies between decision variables, as most EDAs do, between objectives, and decision variables and objectives.

It is well aware that EDAs using joint probability distribution models require a large population for high-dimensional optimization problems. In addition to regularization techniques [42] for single objective optimization, many multiobjective EDAs have been designed by taking advantage of the regularity in distributions of Pareto optimal solutions in both the decision and objective space [28], [43], [44], which is unique to MOPs. One specific observation related to regularity is the Karush-Kuhn-Tucker (KKT) condition, indicating that both PF and PS are $(m-1)$-D piecewise continuous manifolds [45] for m-objective optimization problem under certain mild conditions. To explicitly exploit this regularity property, a multiobjective EDA, termed the regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA), was proposed in [46]. In RM-MEDA, the decision vectors are mapped from the n-dimensional decision space to the $(m-1)$-D latent space using a local principal component analysis presented in [47].

In the literature, most existing multiobjective EDAs focus on the estimation of distribution of the PS in the decision space during the search and still represent the final Pareto optimal solutions in form of a set, such as an archive. Little work has been reported on building a regression model for representing the final solution set. Some preliminary work was reported in [28], where piece-wise linear models were built to represent the solution set achieved by an MOEA, leading to improved quality of the solutions.

An attractive feature of the regularity property is that for any m-objective optimization problems, both PF and PS are (m -1)-D manifolds, regardless of the dimension of the decision space. This regularity property has been essential for some model-based approaches such as RM-MEDA [46]. Most recently, interesting ideas along this line have been reported in [48], [49], where an inverse functional mapping from PF to PS was built based on the approximated PS obtained by MOEAs at the end of evolutionary optimization. This model was then used to generate additional non-dominated solutions, thereby enhancing the density of the solutions. It should be noted that in [48], [49] the inverse model is not used during the optimization.

This paper presents an inverse modeling based multiobjective evolutionary algorithm, IM-MOEA for short. Different from most existing multiobjective EDAs, the proposed IM-MOEA construct models that map non-dominated solutions from the objective space (PF) to the decision space (PS). The original m-input n-output multivariate inverse model is decomposed into $m \times n$ univariate models, which significantly simplifies the model building and removes the need for dimensionality reduction. Each univariate inverse model is then realized by a Gaussian process [50], which has the advantage of modeling both the global regularity and the local randomness in the distribution of the non-dominated solutions during the search. A *random grouping* method originally used for dealing with large scale optimization [51], [52] is introduced to reduce the number of needed inverse models, which considerably enhances the scalability of the algorithm.

It should be noted that inverse modeling itself does not explicitly require the regularity property. However, if the KKT condition holds, the (m-1)-D PF in the objective space will be mapped onto the (m-1)-D PS in the decision space. The fact that both the PF and PS are (m-1)-D will considerably increase the likeliness, although it is not able to guarantee, that the functional mapping from the PF to PS will be a one-to-one mapping. As will be further discussed Section II, if the functional mapping from the PF to PS is a one-to-one mapping, the performance of the proposed algorithm will be much better, although it still works even if this one-to-one mapping condition is not satisfied.

Note also that the motivation of this work is completely different from those in surrogate-assisted evolutionary algorithms (SAEAs) [53], although some work on SAEAs also adopted the Gaussian process as surrogates [54]. Briefly speaking, surrogate models, which construct a function mapping from

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

the decision space to objective space, are used to estimate the fitness of candidate solutions. Surrogate models are helpful when no explicit fitness function exists or when the original fitness function is computationally very expensive, among others [55]. By contrast, the inverse models proposed in this work are used to approximate the distribution of the Pareto optimal solutions, which are then used to generate offspring. Similar discussions on the difference between surrogate methods and inverse modeling for multiobjective optimization can also be found in [48], [49]. In short, the Gaussian models used in our work are for generating new candidate solutions rather than for estimating fitness values. Therefore, the proposed method belongs to a class of multiobjective EDAs that use probabilistic models instead of genetic variations to produce offspring.

The rest of this paper is organized as follows. In Section II, the basic idea of IM-MOEA is first elaborated, followed by the main components of the proposed algorithm. Experimental results comparing the performance of the proposed algorithm and four representative existing MOEAs on 19 test instances are presented in Section III. The effectiveness of *a posteriori* generation of additional non-dominated solutions is also demonstrated. Section IV presents further empirical analysis of the properties of IM-MOEA, including examination of sensitivity of its performance to parameter settings, analysis of its computational complexity, and investigation of the influence of the mutation operator on its performance. Limitations of IM-MOEA and possible remedies have also been discussed. Conclusions and future work are given in Section V.

II. The Proposed Algorithm

A. Basic ideas and the overall framework

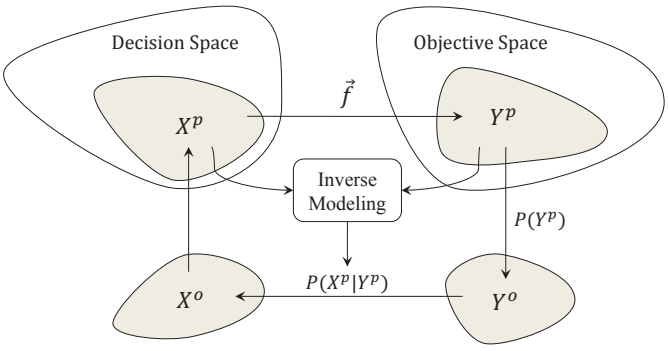


Fig. 1. A schematic illustration of the basic idea of the proposed IM-MOEA. In the figure, X^p and Y^p are a set of the decision and objective vectors, respectively, denoting the current parent population. $\vec{f}$ is the vector of objective functions. To generate offspring, a set of new objective vectors Y^o are sampled from the probability distribution of the parents in the objective space ($P(Y^p)$). Y^o is then mapped back to the decision space using the conditional probability distribution $P(X^p|Y^p)$ approximated by the estimated inverse model, resulting in a set of decision variables X^o. X^o and Y^o together form the offspring population.

Traditional EDAs create offspring by building and sampling a probabilistic model of promising solutions in the decision space. The samples in the decision space are then mapped to the objective space using the objective functions. By contrast, IM-MOEA estimates the conditional probability distribution $P(X^p|Y^p)$, given the distributions of the current parents in the decisions space, denoted by $P(X^p)$, and in the objective space denoted by $P(Y^p)$. Here, the conditional probability $P(X^p|Y^p)$ can be approximated by a probabilistic inverse model mapping Y^p back to X^p, such as a Gaussian process. To create offspring, IM-MOEA starts with generating samples in the objective space, denoted by Y^o, based on the information of the objective values of the current parent population. Sampling in the objective space offers two benefits. First, it is straightforward to generate evenly distributed solutions in the objective space. Second, it is very convenient to incorporate user preferences, if available. Once the offspring set Y^o are created, they are mapped back to the decision space using $P(X^p|Y^p)$ using the Bayes' theorem, thereby completing the generation of the offspring population:

$$P(X^o) = \frac{P(X^p|Y^p)P(Y^o)}{P(Y^p|X^p)}, \tag{3}$$

where $P(Y^p|X^p)$ is the *a priori* knowledge, i.e., the objective functions $\vec{f}$ that map X^p to Y^p. These offspring are combined with the current parent population, from which parents for the next generation will be selected. The offspring will then be combined with the parents, based on which parents for the next generation will be selected.

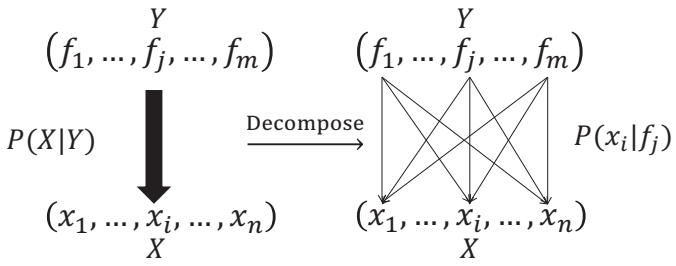


Fig. 2. Decomposition of the m-input-and-n-output multivariate conditional probability distribution (probabilistic inverse model) into $m \times n$ univariate conditional probability distribution.

In practice, it is difficult to directly estimate the m-input-and-n-output inverse model $P(X^p|Y^p)$, where m and n are the number of objectives and the number of decision variables, respectively. To ease the task, we decompose the m-input-and-n-output probabilistic inverse model into $m \times n$ univariate models, $P(x_i|f_j)$, where, $1 \leq i \leq n$ and $1 \leq j \leq m$, as illustrated in Fig. 2. Firstly, by assuming that $x_i, i = 1, 2, ..., n$ are independent, we have

$$P(X|Y) = \prod_{i=1}^{n} P(x_i|Y). \tag{4}$$

Theoretically, this decomposition strictly holds if and only if all decision variables are independent of each other. Nevertheless, the limitation resulting from the independence assumption on the decision variables is alleviated to a great extent due to the random grouping strategy introduced in inverse modeling, where multiple decision variables are randomly grouped together to be derived from the same objective using inverse models. This has been evidenced by the very encouraging results the proposed algorithm has achieved on the test problems

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

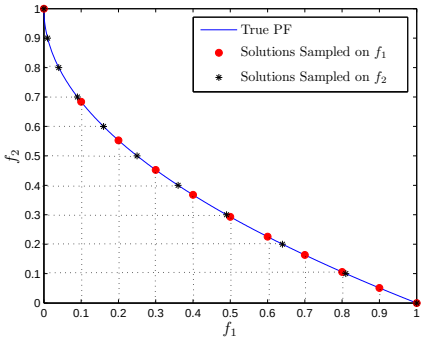


Fig. 3. An example to show the independent sampling on each objective of a bi-objective MOP.

whose decision variables are strongly correlated and the KKT condition is not satisfied.

At this step, we can build inverse models that map solutions from the entire objective space $(f_1, f_2, f_3, ..., f_m)$ to any decision variable $x_i, i = 1, 2, ..., n$. Unfortunately, we are not able to draw samples freely in the whole objective to generate new solutions that are non-dominated or Pareto optimal. Instead, we must construct an additional model in the objective space that can describe the distribution of the estimated PF to constraint the samples. By contrast, if we build multiple univariate inverse models that map single objectives to single decision variables, we can then draw new samples freely within a given interval of each objective. Typically, it is much easier to determine the sampling intervals for each objective than to figure out distribution of the PF. Due to the above reasons, we first rewrite (4) into the following form:

$$P(X|Y) = \prod_{i=1}^{n} P(x_i|f_1, f_2, ..., f_{j-1}, f_j, f_{j+1}, ..., f_m), \quad (5)$$

where $j = 1, 2, ..., m$. If the KKT condition holds, the probability distribution in (5) will depend only on $(m-1)$ objectives. Therefore, we can rewrite the above equation by arbitrarily removing one of the m-objectives:

$$P(X|Y) = \prod_{i=1}^{n} P(x_i|f_1, f_2, ..., f_{j-1}, f_{j+1}, ..., f_m). \quad (6)$$

Let us now consider two different situations. First, for bi-objective optimization problems, if the KKT condition holds, the dimension of the PF and PS equals 1, i.e., (6) can be rewritten into

$$P(X|Y) = \prod_{i=1}^{n} P(x_i|f_j), \quad (7)$$

where $j = 1$ or $j = 2$. In other words, for bi-objective optimization problems, n univariate inverse models can be built to create new candidate solutions by sampling f_1 or f_2. An illustrative example is given in Fig. 3, where we can see that the whole PF can be generated by sampling either f_1 or f_2. It is however advisable to sampling both objectives independently, although one set of the samples is theoretically redundant. In practice, having one redundant set of samples can enhance the population diversity of IM-MOEA.

For $m \geq 2$, (6) can no longer be rewritten into (7). If the KKT condition holds, $x_i, i = 1, 2, ..., n$ may depend on $(m-1)$ objectives. If we divide the entire objective space into a number of subspaces, (6) can be approximated in the following form:

$$P(X|Y) \approx \prod_{i=1}^{n} (P(x_i|f_j) + \epsilon_{j,i}), \quad (8)$$

where $j = 1, 2, ..., m$, $m > 2$, and $\epsilon_{j,i}$ is an error term introduced by converting (6) into (8). The amount of the approximation error depends on the size and location of the subspace the inverse model accounts for. For convenience, we assume that $\epsilon_{j,i} \sim \mathcal{N}(0, (\sigma_n)^2)$ can be captured by additive Gaussian noise. Consequently, Gaussian processes can be used to construct the inverse models.

Similar to the bi-objective case, for an m-objective optimization problem, where $m > 2$, it is sufficient in principle to sample any one of the $m-1$ objectives. However, we still recommend to undertake samples on m objectives separately to enhance the population diversity.

From the above discussions, we can see that sampling from each objective separately and independently is feasible and computationally efficient, although it is not fully equivalent to sampling from the entire objective space.

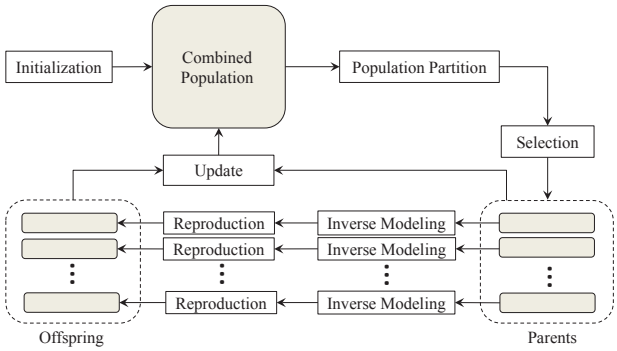


Fig. 4. The framework of the proposed IM-MOEA.

The overall algorithm framework and pseudo code, based on the main ideas presented Fig. 1, are presented in Fig. 4 and Algorithm 1, respectively. In each generation of IM-MOEA, the combined population (parent plus offspring) are first divided into a number of subpopulations based on predefined reference points in the objective space (Line 5 in Algorithm 1). Selection will then be performed within each subpopulation (Line 6 in Algorithm 1). IM-MOEA adopts the elitist non-dominated sorting proposed in NSGA-II [14], which sorts the individuals in each subpopulation into a number of non-dominated fronts and a crowding distance is calculated for individuals on the same front. Selection is then performed within each subpopulation to get the parents of the subpopulation. Then, an inverse model is built for each sub-population using the selected parents in the subpopulation for reproduction. Offspring individuals reproduced by all subpopulations are put together with the parents in the present generation to form the combined population for the next generation. It can be seen that the main components of IM-MOEA, including partition of the combined population, inverse modeling and

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

offspring generation will be described in detail in the following subsections.

Algorithm 1 The pseudo code of the proposed IM-MOEA.

```
/*initialization*/
Initialization: set t = 0, randomly initialize population P(0), define the K reference vectors for subpopulation creation;
/*main loop*/
while termination condition is not satisfied do
    Partition of the combined population: partition the combined population P(t) by associating the solutions with the K predefined reference vectors;
    Non-dominated Sorting and Selection: create sub parent populations P^1(t), ..., P^K(t);
    for k = 1 to K do
        Inverse Modeling: for each subpopulation P^k(t), apply the random grouping technique to determine which inverse models are to be built; training a Gaussian process for each inverse model;
        Reproduction: reproduce new candidate solutions O^k(t) for each subpopulation by sampling the objective space and mapping them back to the decision space using the inverse models; perform mutation on the sampled candidate solutions;
    end for
    Update the combined population: P(t+1) = ⋃_{k=1}^{K} (P^k(t) ∪ O^k(t));
    t = t + 1;
end while
```

B. Partition of the combined population

To ease inverse modeling, the combined population is divided into subpopulations based on a number of predefined reference vectors that evenly partition the objective space, as illustrated in Fig. 6 where a three-objective space is evenly divided by 10 reference vectors.

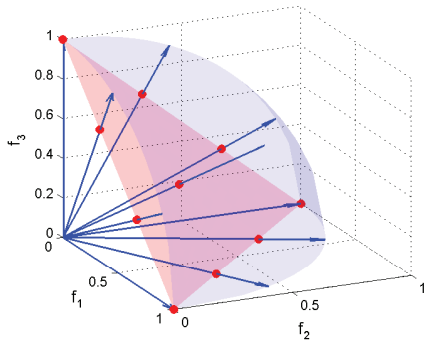


Fig. 5. An illustration of how to generate the evenly distributed reference vectors in a three-objective space. In this case, 10 reference vectors are generated.

The reference vectors are generated using the method suggested in [56]. For an m-objective space, a group of K uniformly distributed points on a unit hyperplane is generated using the canonical simplex-lattice design method [57]:

$$\begin{cases} w_k^i \in \{\frac{0}{H}, \frac{1}{H}, ..., \frac{H}{H}\}, \sum_{i=1}^{m} w_k^i = 1, \\ \vec{w}_k = (w_k^1, w_k^2, ..., w_k^m), k = 1, ..., K, \end{cases} \tag{9}$$

where H is a positive integer. Then, K uniformly distributed reference vectors $\vec{v}_k = [v_k^1, v_k^2, ..., v_k^m], k = 1, 2, ..., K$ can be approximated by mapping the uniformly distributed points from the unit hyperplane to a unit hypersphere as follows:

$$\vec{v}_k = \frac{\vec{w}_k}{||\vec{w}_k||}, k = 1, ..., K. \tag{10}$$

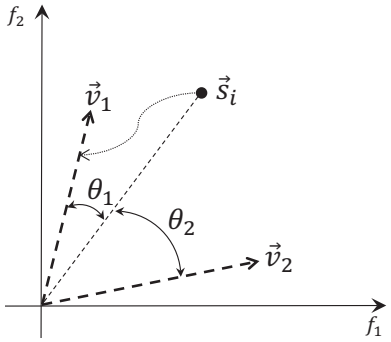


Fig. 6. An illustration of how to associate a candidate solution with a reference vector. In this case, solution $\vec{s}_i$ is associated with reference vector $\vec{v}_1$ since $\theta_1 < \theta_2$.

Given the predefined reference vectors, the combined population is partitioned into K subpopulations by associating each solution with one reference vector according its positions in the objective space. For an arbitrary solution s_i, it is associated with a reference vector $\vec{v}_k$ iff the acute angle between its position in objective space and $\vec{v}_k$ is the minimum among all reference vectors:

$$k_i = \underset{k=1,...,K}{\operatorname{argmin}} (\frac{\vec{s}_i}{||\vec{s}_i||} \times \vec{v}_k), \tag{11}$$

where operator $\times$ calculates the sine function value of the acute angle between $\frac{\vec{s}_i}{||\vec{s}_i||}$ and $\vec{v}_k$. If the combined population size is N, the size of each subpopulation is set to $N_k = \lfloor \frac{N}{K} \rfloor$.

Note that IM-MOEA's strategy for partitioning the combined population into subpopulations using reference vectors shares some similarity to the method used in MOEA/D for dividing the population into subpopulations. The main difference is that while MOEA/D divides the population in the weight space, the proposed IM-MOEA does it in the objective space. Note also that the number of reference vectors used in IM-MOEA is usually much smaller than the number of weight combinations used in MOEA/D. Our empirical results show that 10 reference vectors is typically sufficient for bi-objective and three-objective MOPs, refer to Section IV-A.

C. Inverse modeling

In IM-MOEA, a Gaussian process (GP) has been adopted to estimate decomposed univariate probability distribution, $P_k(x_i|f_j), 1 \leq i \leq n, 1 \leq j \leq m, 1 \leq k \leq K$, for each subpopulation. It is clearly seen that, theoretically, m different inverse models (probability distribution models) can be built

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

for each decision variable, which will make the algorithm computationally very intensive. To address this issue, a random grouping method has been employed to reduce the number of GP models to be built. In the following, we first introduce the random grouping strategy followed by a brief description of the GP model.

1) Random grouping: The random grouping technique here is inspired by the divide-and-conquer methodology widely adopted in large scale optimization [52], where random grouping techniques are used to decompose a high-dimensional optimization problems into a number of low-dimensional subproblems. Without *a prior* knowledge of the nonseparability of a problem, it has been shown that random grouping is able to increase the probability of allocating correlated decision variable into one subproblem [51]. This will be very helpful in inverse modeling in IM-MOEA. On the one hand, it can significantly reduce the number of inverse models. On the other hand, for decision variables assigned in the same group, there correlations can be implicitly taken into account, thereby alleviating the inaccuracy caused by the independence assumption on the decision variables required in decomposing the m-input-and-n-output multivariate probability distribution into univariate ones.

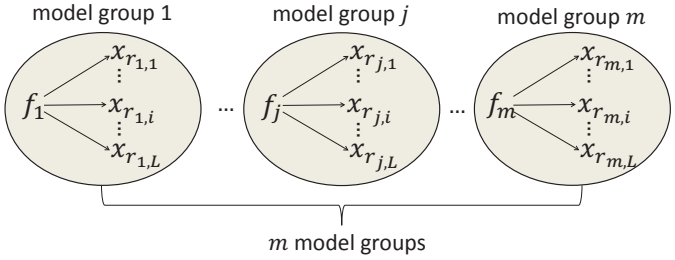


Fig. 7. An illustration of the random grouping method. In each model group j, a number of L decision variables are randomly grouped with the j-th objective to build L GP models. $x_{r_{j,i}}$ denotes the i-th decision variable in the j-th model group. In this way, a number of m model groups are generated with L GP models inside each of them.

Given m objectives, m groups of inverse models will be built for each subpopulation, where all models in group j, $1 \leq j \leq m$ use f_j as the variable. In each group, L decision variables will be randomly assigned to it to build inverse models using f_j as the variable, where $L \ll n$ is a parameter to be specified, refer to Fig. 7. Note that the random assignment is re-performed in each generation. For example, for a 10-dimensional three-objective optimization problem, three groups of models will be generated. If $L = 2$, then as a result of random assignment in a certain generation, x_2, x_4 are assigned to group 1, x_1, x_6 to group 2, and x_5 and x_8 to the third group. Thus, the first group contains the following two inverse models: $P(x_2|f_1), P(x_4|f_1)$, the second group contains $P(x_1|f_2), P(x_6|f_2)$, and the third group has $P(x_5|f_3), P(x_8|f_3)$. Consequently, the current values of the assigned decision variables (x_1, x_2, x_4, x_5, x_6 and x_8) will be replaced with new values generated by the inverse models, while the values of the other four decision variables (x_3, x_7, x_9 and x_{10}) will remain unchanged.

As we can see from the above example, for n-dimensional m-objective problems, the random group strategy can reduce the number of inverse models from $m \times n$ to $m \times L$, where L is usually much smaller than n.

2) Inverse modeling using Gaussian processes: Assume that the population size of the k-th subpopulation is N_k, meaning that there are N_k individuals (data pairs) in total for training the m groups of inverse models in this subpopulation. In order to increase the population diversity, each group of inverse models are trained using different training data pairs. Therefore, the individuals are approximately evenly and randomly divided into m groups, each having $N_t = \lfloor \frac{N_k}{m} \rfloor$ data for training L GP models. Note that for the L GP models in each group, the training and sampling procedure will be carried out only if there are at least two training data points, i.e, $N_t \geq 2$. Otherwise, no training and sampling of the GP model will be performed. Denote the training data set for training the inverse model ($P(x_i|f_j)$) in the j-th group as $T_{j,i}$, then

$$T_{j,i} = [\mathbf{f_j}, \mathbf{x_i}], \tag{12}$$

where $\mathbf{f_j} = (f_j^1, ..., f_j^{N_t})^\top$ and $\mathbf{x_i} = (x_i^1, ..., x_i^{N_t})^\top$, f_j^l, x_i^l are the j-th objective values and i-th decision value of the l-th individual, $1 \leq l \leq N_t$ that are randomly allocated to the j-th group of models. In the following, we briefly show how to estimate the probability distribution $P(x_i|f_j)$ using a GP model based on the given training data $T_{j,i}$.

GPs can be seen as a generalization of the Gaussian probability distribution in the function space [50], [58], [59], which is based on the assumption that the latent function is a sample of a Gaussian stochastic process. In this way, the inverse mapping $P(x_i|f_j)$ can be seen as a latent function $g(\cdot)$ represented by a number of n arbitrary function variables $\mathbf{g} = \{g_1, g_2, ..., g_{N_t}\}$ that follow a joint Gaussian distribution [60]:

$$P(\mathbf{g}|\mathbf{f_j}) = \mathcal{N}(\bar{\mathbf{g}}, C) \tag{13}$$

where $\mathcal{N}(\bar{\mathbf{g}}, C)$ denotes the multivariate Gaussian distribution with mean vector $\bar{\mathbf{g}}$ and covariance matrix C. $P(\mathbf{g}|\mathbf{f_j})$ is the conditional probability (regression prediction) with respect to the training data $\mathbf{f_j}$.

Training the GP model means to estimate the conditional probability of $P(\mathbf{g}|\mathbf{f_j})$, which is fully specified by the mean function $\mu(\mathbf{f_j})$ that calculates the mean value of all the training data in $\mathbf{f_j}$, and the covariance function $c(f_j^p, f_j^q)$ that calculates the covariance between any two training data points f_j^p and f_j^q, with $p, q \in 1, ..., N_t$. Without loss of generality, in practice, the mean function is always set to zero by subtracting an offset that can be obtained from the training data $T_{j,i}$:

$$P(\mathbf{g}|\mathbf{f_j}) = \mathcal{N}(0, C), \tag{14}$$

which reduces the task of training a GP model to the estimation of the covariance matrix C by choosing suitable covariance functions $c(f_j^p, f_j^q)$. In this work, for computational efficiency, the following simple linear covariance function without parameters is adopted [50]:

$$c(f_j^p, f_j^q) = f_j^{p\top} f_j^q. \tag{15}$$

By adopting this covariance function, the computational cost of the hyperparameter optimization can be saved.

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

Based on the assumption that the observations $\mathbf{x_i} = g(\mathbf{f_j}) + \epsilon$ are affected by white noise $\epsilon \sim \mathcal{N}(0, (\sigma_n)^2 I)$, a suitable noise model can be represented by

$$P(\mathbf{x_i}|\mathbf{g}) = \mathcal{N}(\mathbf{g}, (\sigma_n)^2 I), \tag{16}$$

where I is an identity matrix. In this way, the marginal likelihood can be obtained as follows:

$$P(\mathbf{x_i}|\mathbf{f_j}) = \int P(\mathbf{x_i}|\mathbf{g}) P(\mathbf{g}|\mathbf{f_j}) d\mathbf{g} = \mathcal{N}(0, C + (\sigma_n)^2 I). \tag{17}$$

With (17), given a test input $\mathbf{f_{j,*}}$, the predicted output $\mathbf{x_{i,*}}$ of the latent function $g(f_{j,*})$ can be obtained as a Gaussian distribution by applying Bayesian inference, with mean and variance being calculated as follows:

$$\begin{aligned} \mu_{j,i} &= C_*^\top (C + (\sigma_n)^2 I)^{-1} \mathbf{x_i}, \\ (\sigma_{j,i})^2 &= C_{**} - C_*^\top (C + (\sigma_n)^2 I)^{-1} C_*, \end{aligned} \tag{18}$$

where $C_{**} = [c(f_j^1, f_j^1), ..., c(f_j^{N_t}, f_j^{N_t})]$ is a matrix of covariance parameters between each element in the training data $\mathbf{f_j}$, and $C_* = [c(f_{j,*}^1, f_j^1), ..., c(f_{j,*}^{N_t}, f_j^{N_t})]$ is a matrix of covariance parameters between each element in the test input $\mathbf{f_{j,*}}$ and each element in the training data $\mathbf{f_j}$. In this way, the inverse model $P(x_i|f_j)$ is represented as a set of normal distributions described in (18).

In our case, the test input points $\mathbf{f_{j,*}}$ are directly sampled in the objective space based on the estimated range of f_j. Theoretically, a decision maker would be able to integrate any preference in generating samples for the test input points. In this work, without loss of generality, the test input points are uniformly generated within an interval extended from $[f_j^{min}, f_j^{max}]$ as follows:

$$\mathbf{f_{j,*}} = \{\frac{[f_j^{min} - 0.5\xi_j, f_j^{max} + 0.5\xi_j]}{N_s}\}, \tag{19}$$

where N_s is the sample size, f_j^{min} and f_j^{max} are the minimum and maximum values among the elements in the training data $\mathbf{f_j}$, respectively, and $\xi_j = f_j^{max} - f_j^{min}$, as shown in Fig. 8.

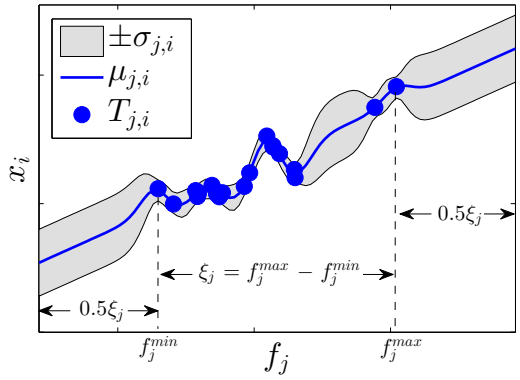


Fig. 8. An example of a Gaussian process. The training target is to regress the inverse mapping from objective f_j to decision variable x_i given the training data $T_{j,i}$. The curve in the middle of the figure consists of the mean values $\mu_{j,i}$ of the predictions on the test input points uniformly distributed within $[f_j^{min} - 0.5\xi, f_j^{max} + 0.5\xi]$, refer to (19), and the dashed area covers the length of $\pm\sigma_{j,i}$ around the mean value curve $\mu_{j,i}$, which reflects the uncertainty of the prediction on each test input point.

D. Reproduction

Once the inverse models learned by the GPs are available, the test input $\mathbf{f_{j,*}}$, which are uniformly generated in the objective space in training the GP, can be mapped to the decision space, $\mathbf{x_{i,*}}$. This can be obtained by adding the mean values $\mu_{\mathbf{j,i}} = (\mu_{j,i}^1, ..., \mu_{j,i}^{N_s})$ with Gaussian white noise $\mathbf{z} \sim (\mathcal{N}(0, (\sigma_{j,i}^1)^2), ..., \mathcal{N}(0, (\sigma_{j,i}^{N_s})^2))$:

$$\mathbf{x_{i,*}} = \mu_{\mathbf{j,i}}(\mathbf{f_{j,*}}) + \mathbf{z_{j,i}}, \tag{20}$$

where $\mathbf{f_{j,*}}$ is the test input as in (19), $\mu_{j,i}^1, ..., \mu_{j,i}^{N_s}$ and $(\sigma_{j,i}^1)^2, ..., (\sigma_{j,i}^{N_s})^2$ are the standard output of a trained GP model as in (18), and N_s is the sample dataset size. For simplicity, N_s is set equal to the training dataset size N_t, such that the newly generated candidate solutions $\mathbf{x_{i,*}} = (x_{i,*}^1, ...x_{i,*}^{N_s})$ can exactly replace the parent individuals $\mathbf{x_i} = (x_i^1, ..., x_i^{N_t})$ in the current subpopulation. In order to enhance the population diversity, mutation is applied to all the sampled individuals, which is a common practice in EDAs [61]. In this work, the widely used polynomial mutation operator is adopted [62], although it is not limited to any particular mutation operation. However, it should be noted that the proposed IM-MOEA can still achieve equally good performance on most MOPs without the mutation operator. Further discussions about the influence of the mutation operator on the performance of IM-MOEA will be provided in Section IV-C.

III. Comparative Studies

In this section, the performance of IM-MOEA is compared with four state-of-the-art MOEAs, including the RM-MEDA [46], an improved version of RM-MEDA [63], denoted as IRM-MEDA hereafter, a variant of MOAE/D with differential evolution [64], denoted as MOEA/D-DE hereafter, and NSGA-II [14].

In the following, we will first introduce the test instances and performance metrics used in this study. Then, a brief introduction to the four compared algorithms is provided together with the experimental settings. Then we present and discuss the optimization results. Finally, the benefit of the inverse-model based representation of the non-dominated solution is further exploited by generating solutions *a posteriori*.

A. Test instances

In our experimental studies, we use a total of 19 test instances for comparison. The first test suite consists of ten test instances (F1 to F10), which are modified from the test instances introduced in [46], and the second one is the WFG test suite, containing nine test problems (WFG1 to WFG9) [65].

The test instances used in [46] were modified from the ZDT [66] and DTLZ [67] test instances to introduce linear or non-linear correlations between the decision variables by linking the first decision variable to the rest, which is very specific. To be more realistic, we revise the linkages between the decision variables as follows:

$$\begin{aligned} x_i &\rightarrow (1 + \alpha\frac{i}{n})x_i, \\ x_i^2 &\rightarrow x_i^{\frac{1}{1+\beta\frac{i}{n}}}, \end{aligned} \tag{21}$$

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

TABLE I
MODIFIED ZDT AND DTLZ TEST INSTANCES WITH LINEAR AND NONLINEAR VARIABLE LINKAGES.

Instance	Variables	Objectives	Characteristics
$F1$	$[0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^{n}((1+\alpha\frac{i}{n})x_i - x_1)^2)/(n-1)$	convex PF linear variable linkage
$F2$	$[0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^{n}((1+\alpha\frac{i}{n})x_i - x_1)^2)/(n-1)$	concave PF nonuniformly distributed linear variable linkage
$F3$	$[0,1]^n$	$f_1(x) = -\exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^{n}((1+\alpha\frac{i}{n})x_i - x_1)^2)/(n-1)$	concave PF linear variable linkage
$F4$	$[0,1]^n$	$f_1(x) = \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)(1+g(x))$ $f_2(x) = \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2)(1+g(x))$ $f_3(x) = \sin(\frac{\pi}{2}x_1)(1+g(x))$ $g(x) = \sum_{i=3}^{n}((1+\alpha\frac{i}{n})x_i - x_1)^2$	concave PF linear variable linkage 3 objectives
$F5$	$[0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ $g(x) = 1 + 9(\sum_{i=2}^{n}(x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1)^2)/(n-1)$	convex PF nonlinear variable linkage
$F6$	$[0,1]^n$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^{n}(x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1)^2)/(n-1)$	concave PF nonuniformly distributed nonlinear variable linkage
$F7$	$[0,1]^n$	$f_1(x) = -\exp(-4x_1)\sin^6(6\pi x_1)$ $f_2(x) = g(x)[1 - (f_1(x)/g(x))^2]$ $g(x) = 1 + 9(\sum_{i=2}^{n}(x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1)^2)/(n-1)$	concave PF nonlinear variable linkage
$F8$	$[0,1]^n$	$f_1(x) = \cos(\frac{\pi}{2}x_1)\cos(\frac{\pi}{2}x_2)(1+g(x))$ $f_2(x) = \cos(\frac{\pi}{2}x_1)\sin(\frac{\pi}{2}x_2)(1+g(x))$ $f_3(x) = \sin(\frac{\pi}{2}x_1)(1+g(x))$ $g(x) = \sum_{i=3}^{n}(x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1)^2$	concave PF nonlinear variable linkage 3 objectives
$F9$	$[0,1]\times[0,10]^{n-1}$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ $g(x) = \sum_{i=2}^{n}\frac{(x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1)^2}{4000} - \prod_{i=2}^{n}\cos(\frac{x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1}{\sqrt{i-1}}) + 2$	convex PF nonlinear variable linkage multimodal with Griewank function
$F10$	$[0,1]\times[0,10]^{n-1}$	$f_1(x) = x_1$ $f_2(x) = g(x)[1 - \sqrt{f_1(x)/g(x)}]$ $g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}[(x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1)^2 - 10\cos(2\pi(x_i^{\frac{1}{1+\beta\frac{i}{n}}} - x_1))]$	convex PF nonlinear variable linkage multimodal with Rastrigin function

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

where i is the index of each decision variable, α and β are two control parameters. The resulting test instances are summarized in Table I, in which the linkage between the variables are variable specific, controlled by two parameters α and β. In this work, we set $\alpha = 5$ and $\beta = 3$, and the number of decision variables is set to $n = 30$. Note however, that the Pareto sets of F1 to F10 are still (m - 1)-D manifolds.

By contrast, the dimensionality of the Pareto set of the WFG test instances can be defined to be higher than (m - 1). In this study, we use three-objective WFG test instances whose Pareto set has a dimensionality of four. The number of decision variables of the WFG functions is set to 34 and 104, respectively.

B. Performance indicators

Inverted generational distance (IGD) [46], [68] and hypervolume (HV) [62] are used as two performance indicators in the performance comparisons.

1) Inverted generational distance (IGD): Let P^* be a set of uniformly distributed solutions along PF. Let P be an approximation to PF. The inverted generational distance between P^* and P can be defined as:

$$IGD(P^*, P) = \frac{\sum_{v \in P^*} d(v, P)}{|P^*|}, \tag{22}$$

where $d(v, P)$ is the minimum Euclidean distance from the point v to P. The IGD metric is able to measure both diversity and convergence of P if $|P^*|$ is large enough, and a smaller IGD value indicates a better performance. In this work, IGD is used to measure the performance on test instances F1 to F10 (most of which are bi-objective MOPs), and 500 uniformly distributed points are selected from the PF of each test instance to be P^*.

2) Hypervolume (HV): Let $y^* = (y_1^*, ..., y_m^*)$ be a reference point in the objective space which is dominated by all Pareto optimal solutions. Let P be the approximation to PF. The HV value of P (with regard to y^*) is the volume of the region which is dominated by P and dominates y^*. In this work, HV is used to measure the performance on the three-objective WFG test instances, and the reference point $y^* = (2.5, 4.5, 6.5)$ is used for all instances.

C. Four compared algorithms

In this study, we compare IM-MOEA with the following four algorithms.

1) RM-MEDA is one of the state-of-the-art multiobjective estimation of distribution algorithms. It is meant to design a reproduction operator by explicitly modeling the regularity in the distribution of Pareto optimal solutions. RM-MEDA has been shown to outperform many traditional MOEAs on some of the modified ZDT and DTLZ functions.
2) IRM-MEDA is an enhanced variant of RM-MEDA. In IRM-MEDA, a strategy for adapting the cluster number is suggested to improve the performance of RM-MEDA, where the number of clusters is predefined. It is shown that IRM-MEDA has significantly better performance in comparison with RM-MEDA in terms of both search performance and computational efficiency.
3) MOEA/D-DE is a popular MOEA based on weighted aggregation, which has been shown to perform well on various MOPs [69]–[71]. In this comparative study, we use MOEA/D-DE, a variant where the simulated binary crossover (SBX) operator is replaced by the crossover operator in differential evolution [72]. In addition, a two-level neighborhood control is used in MOEA/D-DE to control the number of old solutions that can be replaced by a new solution, such that better population diversity can be maintained. In addition, polynomial mutation is used.
4) NSGA-II is probably the most popular dominance based MOEAs [73]–[76]. Here, SBX and polynomial mutation are used for reproduction.

D. Experimental settings

TABLE II
PARAMETER SETTINGS OF THE FOUR ALGORITHMS IN COMPARISON.

Algorithm	Parameter settings
RM-MEDA	$K = 5$
IRM-MEDA	$\theta = \frac{3}{180}\pi$
MOEA/D-DE	$T = 20$, $\delta = 0.9$, $n_r = 2$ $p_m = 1/n$, $\eta_m = 20$
NSGA-II	$p_c = 1.0$, $\eta_c = 20$ $p_m = 1/n$, $\eta_m = 20$

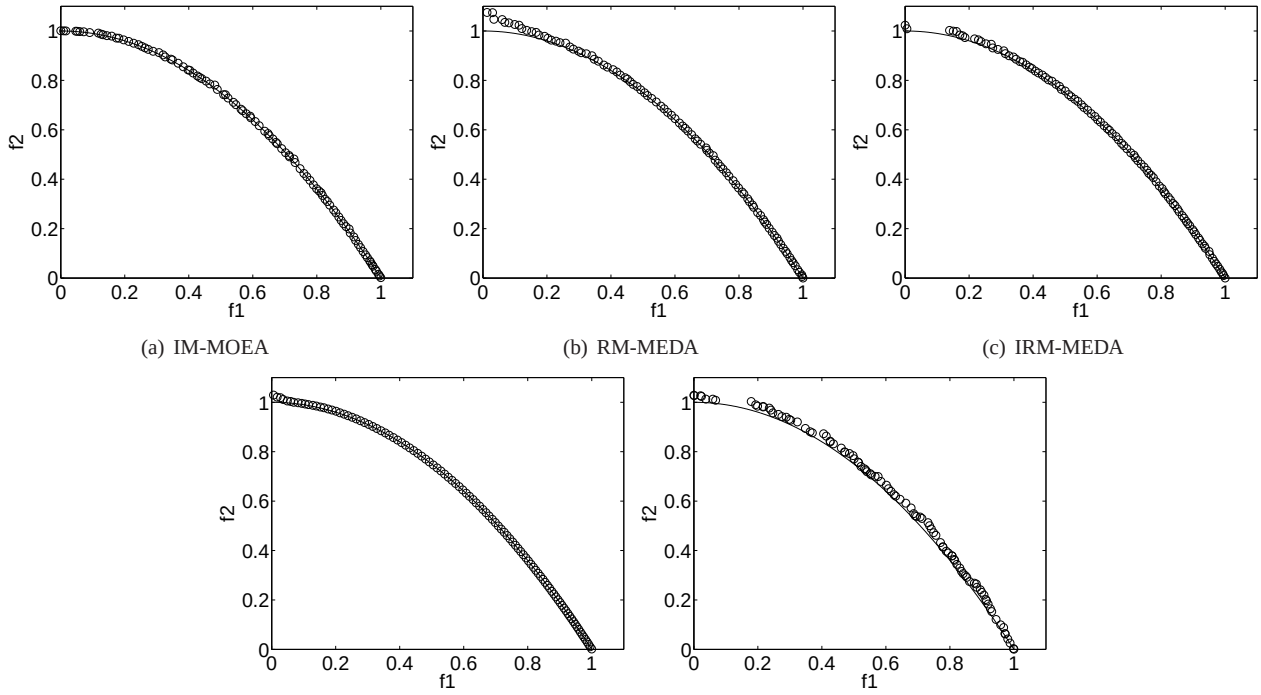
The parameter settings adopted in our studies for the compared algorithms were all recommended by the original papers, as summarized in Table II. For the proposed IM-MOEA, there are two parameters to be set, i.e., the number of reference vectors K, and the model group size L. In the studies here, K is set to 10 and L set to 3. Sensitivity analysis of the performance to the parameters will be examined in Section IV-A.

The population size is set to 100 for IM-MOEA, RM-MEDA, IRM-MEDA and NSGA-II. However, the population size for MOEA/D-DE needs to be specified based on the number of objectives. As a result, we use 100 and 105 as the population sizes for MOEA/D-DE on bi-objective and three-objective MOPs, respectively.

We performed 20 independent runs for each compared algorithm on each test instance. The termination condition for each algorithm is set to a maximum of 100,000 fitness evaluations for all the test instances. The Wilcoxon rank sum test is adopted to compare the results obtained by IM-MOEA and those by the other four algorithms at a significance level of 0.05. In the tables that summarize the statistical results, the first line presents the mean values and the second line the standard deviations. As a result of the Wilcoxon rank sum test, a $+$ labeled in front of a result indicates that the compared algorithm is outperformed by IM-MOEA; by contrast, a $-$ means that IM-MOEA is outperformed by the compared algorithm; while a $\approx$ mean that there is no statistically significant difference between the results obtained by IM-MOEA and the compared algorithm. The best statistical results are all highlighted.

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation



(a) IM-MOEA (b) RM-MEDA (c) IRM-MEDA

(d) MOEA/D-DE (e) NSGA-II

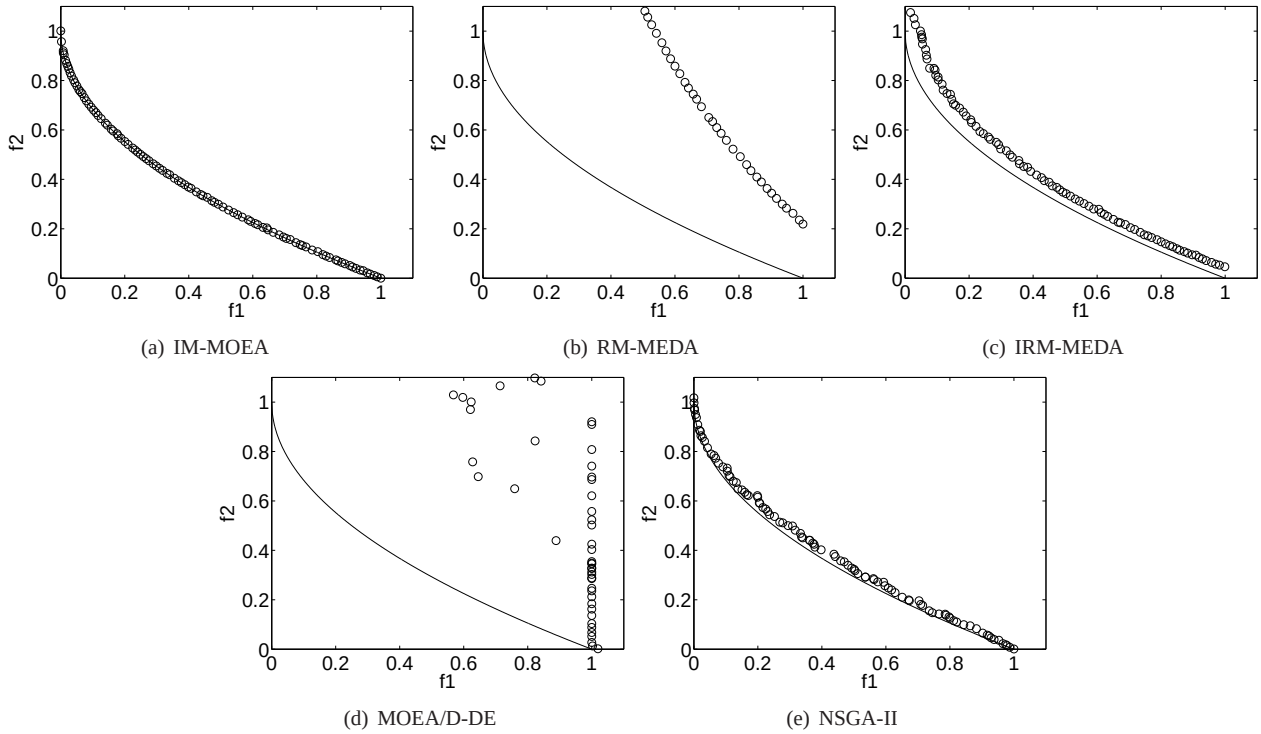
Fig. 9. The non-dominated solutions with the best IGD values obtained by each algorithm among 20 runs in the objective space on F6.

TABLE III
STATISTICAL RESULTS OF IGD VALUES OBTAINED BY IM-MOEA AND EACH ALGORITHM IN COMPARISON ON 30-D F1 TO F10.

Test instances	IM-MOEA		RM-MEDA		IRM-MEDA		MOEA/D-DE		NSGA-II
F1	4.044E−03	+	4.795E−03	+	4.903E−03	+	4.154E−03	+	1.558E−02
	4.573E−05		1.496E−04		1.875E−04		1.751E−05		6.555E−04
F2	4.261E−03	+	4.555E−03	+	4.629E−03	−	3.899E−03	+	1.845E−02
	6.013E−05		2.441E−04		2.348E−04		2.081E−05		2.284E−03
F3	2.199E−03	+	3.518E−03	+	2.603E−03	≈	2.144E−03	+	8.576E−03
	1.108E−04		1.386E−03		1.529E−04		1.569E−04		2.349E−03
F4	7.049E−02	+	8.806E−02	+	9.427E−02	+	2.924E−01	+	1.788E−01
	2.290E−03		6.837E−03		1.058E−02		2.812E−01		1.206E−02
F5	4.357E−03	+	7.136E−03	+	6.473E−03	−	4.228E−03	+	1.510E−02
	7.131E−05		5.365E−04		3.000E−04		1.472E−04		8.974E−04
F6	5.127E−03	+	1.175E−02	+	9.464E−03	≈	5.288E−03	+	1.943E−02
	1.709E−04		1.015E−03		1.402E−03		6.279E−04		1.527E−03
F7	2.703E−03	+	1.285E−02	+	6.474E−03	−	2.178E−03	+	7.107E−03
	3.672E−04		4.836E−03		2.231E−03		1.069E−04		5.803E−04
F8	8.147E−02	+	1.171E−01	+	1.102E−01	+	1.005E−01	+	5.512E−01
	4.396E−03		8.340E−03		4.793E−03		1.007E−02		1.235E−01
F9	5.372E−03	+	3.684E−01	+	1.954E−01	+	3.198E−01	+	2.160E−02
	1.053E−03		1.665E−01		2.005E−01		2.003E−02		2.759E−03
F10	1.022E+00	+	5.886E+01	+	3.953E+01	+	3.629E+00	−	2.180E−01
	4.395E−01		2.460E+01		7.846E+00		2.065E+00		2.765E−02
+ / ≈ / −			10 / 0 / 0		10 / 0 / 0		5 / 2 / 3		9 / 0 / 1

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation



(a) IM-MOEA (b) RM-MEDA (c) IRM-MEDA

(d) MOEA/D-DE (e) NSGA-II

Fig. 10. The non-dominated solutions with the best IGD values obtained by each algorithm among 20 runs in the objective space on F9.

E. Results

1) Performance on F1 to F10: Table III summarizes the statistical results in terms of IGD values obtained by the five algorithms. It can be seen that IM-MOEA shows the best overall performance. The best results on F1, F4, F6, F8 and F9 are obtained by IM-MOEA. From these results, we can see that IM-MOEA outperforms the four compared algorithms on F1, F4, F6, F8, and F9. We also note that the second best performed algorithm is MOEA/D-DE, which shows very competitive performance on F2, F3, F5 and F7. However, MOEA/D-DE is significantly outperformed by IM-MOEA on the two three-objective test instances, F4 and F8. We also find that IM-MOEA performs significantly better than RM-MEDA and IRM-MEDA on all the 10 test problems, and the performance difference is even more significant on the six test instances whose decision variables are nonlinearly correlated. Note that RM-MEDA and IRM-MEDA have also been shown very promising on these test instances. In particular, IRM-MEDA has shown very competitive results on F5 to F10, which have strong nonlinear correlations between the decision variables. On the other hand, it is interesting to note that NSGA-II outperforms all other compared algorithms on F10, which is a highly multimodal test instance, although it is outperformed by others on most of the test instances.

The non-dominated solutions achieved by each compared algorithm on F6 and F9 in the run resulting the best IGD among 20 runs are presented in Fig. 9 and Fig. 10, respectively. Note that MOEA/D-DE achieves the best IGD on F6, and naturally, the non-dominated solutions it achieved show excellent distribution and accuracy on most part of the PF. Nevertheless, the solutions achieved by IM-MOEA on the left end of the non-dominated front have better convergence than those achieved by MOEA/D-DE. On F9, IM-MOEA has achieved the best IGD and the performance difference between IM-MOEA and the compared algorithms becomes more apparent if we take a look at the non-dominated solutions shown in Fig. 10. By contrast, the performance of MOEA/D-DE on F9 is not as attractive as on the other test instances. This might be attributed to the fact that randomly generated candidate solutions of F9 are strongly non-uniformly distributed in the objective space, which may result in a quick loss of the diversity of the population, leading to a premature convergence. This issue could be addressed by introducing a partition mechanism in the objective space, e.g., as the one presented in MOEA/D-M2M [56].

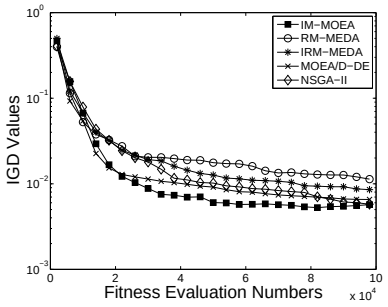


Fig. 11. Averaged convergence profiles of the IGD values for the five algorithms on F6.

To examine the convergence speed of the five algorithms, the convergence profiles of the IGD values on F6 and F9 are plotted in Fig. 11 and Fig. 12, respectively. It can be seen that although IM-MOEA converges slower than the

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

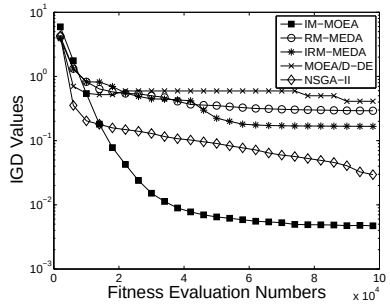


Fig. 12. Averaged convergence profiles of the IGD values for the five algorithms on F9.

compared algorithms when the number of fitness evaluations is smaller than some 20,000. After that, IM-MOEA exceeds the compared algorithms in convergence. This is reasonable as IM-MOEA needs sufficient training samples to build the inverse models correctly.

2) Performance on the WFG test instances: In this section, we compare IM-MOEA with the other four algorithms on three-objective WFG test instances. To evaluate the scalability of IM-MOEA, two sets of comparisons are conducted, one using 34-D test instances and the other 104-D.

Table IV summarizes the results with respect to HV values obtained by the five algorithms on 34-D instances. It can be seen that IM-MOEA performs statistically significantly better than the other four compared algorithms on six out of nine test instances and comparably well with MOEA/D-DE on WFG3. RM-MEDA outperforms others on WFG2, while MOEA/D-DE performs the best on WFG6. Table V shows the optimization results regarding HV values obtained by the five algorithms on 104-D instances. Similarly, IM-MOEA performs significantly better than others on seven test instances, while RM-MEDA performs the best on WFG1 and MOEA/D-DE on WFG6. The proposed IM-MOEA continues to scale up well on most instances, except WFG1 and WFG6. It can be seen that RM-MEDA and IRM-MEDA has good scalability on WFG1, and MOEA/D-DE scales up well on WFG6.

The non-dominated solutions resulting in the best HV among the 20 runs achieved by each algorithm on WFG5 and WFG6 are presented in Figs. 13 and 14, respectively. From Fig. 13, we can see that solutions obtained by IM-MOEA on WFG5 exhibit much better convergence and distribution compared to those of others, confirming the better performance indicated by the HV. On the other hand, as indicated by HV, MOEA/D-DE performs best on WFG6, in particular in terms convergence, refer to Fig. 14. The overall distribution of the solutions obtained IM-MOEA, however, is better than that of the solutions achieved by MOEA/D-DE.

It should be noted that overall, RM-MEDA and IRM-MEDA is not competitive in comparison with IM-MOEA. This might be due to the fact that the principal curves used in RM-MEDA and IRM-MEDA are linear, which are inadequate to describe the strongly non-linear distributions of the modified DTZ and DTLZ test problems as well as the WFG test instances. Among the other four compared algorithms, RM-MEDA and IRM-MEDA show significantly better overall performance than NSGA-II, although they are slightly outperformed by MOEA/D-DE on some test instances. Another observation we can make is that non-dominated solutions obtained by MOEA/D-DE are very sparse on the top and denser in the bottom for test instances from WFG4 to WFG9. This is probably due to the fact that these WFG test instances are non-uniform, which means that a uniform distribution in weights will lead to a non-uniform distribution in the solutions.

F. A posteriori *sampling*

Most existing MOEAs store the obtained non-dominated solutions in form of a set in the population or in an archive. In this case, once the optimization is complete, no new solutions can be obtained without performing additional optimization runs. Very recently, it has been suggested that an *inverse mapping* from the objective space to the decision space can be very useful for enriching the solutions after the optimization is complete [48], [49]. The benefit of model-based representation has also been discussed in [28], [38].

One advantage of the proposed IM-MOEA over most MOEAs is that inverse models from the objectives to the decision space have been learned in addition to the non-dominated solutions. Naturally, we are able to sample solutions with the estimated inverse models in preferred regions in the objective space. The only difference is that to sample additional solutions, a full inverse mapping needs to be trained, i.e., the number of models to be built for group L should be set to n so that all decision variables can be obtained given a sampled solution in the objective space.

Fig. 15(a) shows an example in which 100 non-dominated solutions have been achieved by IM-MOEA during optimization. However, solutions in the interested regions, defined by three circles of a radius centered at $(0.3, 0.3, 0.9)$, $(0.3, 0.9, 0.3)$, $(0.9, 0.3, 0.3)$, respectively, are very limited. To obtain additional solutions in these regions, the non-dominated solutions inside each circle shown in Fig. 15(a) are utilized for training the inverse models related to the interested region. Once the inverse models have been trained, 300 new solutions are sampled inside the interested regions as in Fig. 15(b). If the decision maker is still not satisfied with the 300 non-dominated solutions in Fig. 15(b), they can be used to train more inverse models, and consequently more new solutions can be generated. Fig. 15(c) shows 3000 new solutions generated using the inverse models in the three interested regions.

IV. Analysis and Discussions

This section empirically analyzes the sensitivity of the algorithm's performance to its parameters and examines its runtime in comparison with RM-MEDA. Then, the influence of the mutation operator on the performance of IM-MOEA is investigated. Finally, some discussions on the limitations of IM-MOEA as well as possible remedies are given.

A. Parameter sensitivity analysis

In IM-MOEA, there are two parameters that require to be specified by the user, i.e., the number of reference vectors

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

TABLE IV
STATISTICAL RESULTS OF HV VALUES OBTAINED BY IM-MOEA AND EACH ALGORITHM IN COMPARISON ON 34-D WFG FUNCTIONS.

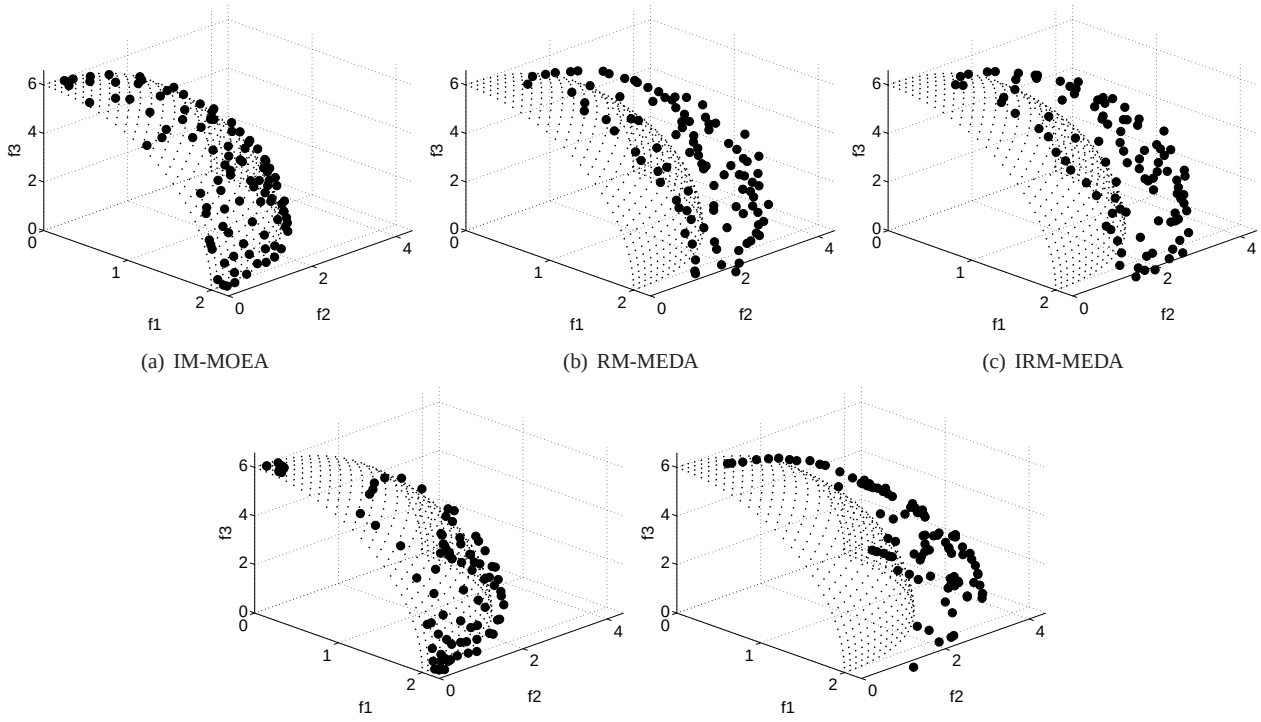
Test instances	IM-MOEA		RM-MEDA		IRM-MEDA		MOEA/D-DE		NSGA-II
WFG1	2.585E+01 8.538E−01	+	2.581E+01 1.287E−01	≈	2.584E+01 1.126E−01	+	2.445E+01 7.427E−01	≈	2.546E+01 5.750E−01
WFG2	6.520E+01 2.086E+00	−	6.587E+01 4.709E−01	+	6.464E+01 3.519E−01	+	6.423E+01 4.804E−01	+	5.055E+01 2.570E+00
WFG3	4.504E+01 2.581E−01	≈	4.473E+01 1.902E−01	+	4.294E+01 3.506E−01	≈	4.515E+01 3.994E−01	+	4.046E+01 6.457E−01
WFG4	4.164E+01 2.685E−01	+	3.660E+01 2.080E−01	+	3.514E+01 2.968E−01	+	3.692E+01 4.460E−01	+	3.014E+01 1.082E+00
WFG5	4.080E+01 1.169E−01	+	2.629E+01 1.676E+00	+	2.414E+01 8.942E−01	+	3.685E+01 4.466E−01	+	2.453E+01 3.342E+00
WFG6	3.905E+01 7.446E−01	+	3.478E+01 5.419E−01	+	3.100E+01 8.612E−01	−	4.019E+01 4.635E−01	+	3.013E+01 1.210E+00
WFG7	4.148E+01 1.665E−01	+	3.525E+01 1.138E+00	+	3.242E+01 5.902E−01	+	3.950E+01 3.596E−01	+	2.855E+01 8.657E−01
WFG8	3.830E+01 1.558E−01	+	2.961E+01 6.815E−01	+	2.788E+01 6.825E−01	+	3.279E+01 5.944E−01	+	2.186E+01 1.202E+00
WFG9	3.901E+01 7.841E−01	+	3.750E+01 1.693E−01	+	3.672E+01 2.582E−01	+	3.776E+01 4.233E−01	+	3.275E+01 2.265E+00
+ / ≈ / −			7 / 1 / 1		8 / 1 / 0		7 / 1 / 1		8 / 1 / 0

TABLE V
STATISTICAL RESULTS OF HV VALUES OBTAINED BY IM-MOEA AND EACH ALGORITHM IN COMPARISON ON 104-D WFG TEST INSTANCES.

Test instances	IM-MOEA		RM-MEDA		IRM-MEDA		MOEA/D-DE		NSGA-II
WFG1	2.370E+01 1.207E+00	−	2.624E+01 1.055E−01	−	2.607E+01 7.768E−02	+	2.335E+01 6.886E−01	−	2.510E+01 2.863E−01
WFG2	6.194E+01 2.869E+00	+	6.178E+01 3.962E−01	+	6.055E+01 5.180E−01	+	5.944E+01 2.907E+00	+	4.577E+01 2.770E+00
WFG3	4.359E+01 1.532E−01	+	4.192E+01 2.122E−01	+	4.062E+01 2.302E−01	+	4.284E+01 3.554E−01	+	3.514E+01 5.647E−01
WFG4	4.078E+01 2.631E−01	+	3.490E+01 2.045E−01	+	3.324E+01 2.859E−01	+	3.518E+01 6.578E−01	+	2.627E+01 4.823E−01
WFG5	4.048E+01 1.426E−01	+	2.211E+01 1.501E+00	+	2.195E+01 9.046E−01	+	3.637E+01 4.154E−01	+	1.841E+01 1.093E+00
WFG6	3.719E+01 7.065E−01	+	3.083E+01 6.910E−01	+	2.452E+01 4.958E−01	−	4.060E+01 5.192E−01	+	2.206E+01 1.722E+00
WFG7	3.991E+01 3.415E−01	+	3.249E+01 7.197E−01	+	3.032E+01 4.177E−01	+	3.658E+01 9.823E−01	+	2.299E+01 7.120E−01
WFG8	3.925E+01 2.148E−01	+	2.921E+01 2.820E−01	+	2.734E+01 3.700E−01	+	3.151E+01 6.890E−01	+	1.979E+01 1.096E+00
WFG9	3.816E+01 1.130E+00	+	3.218E+01 5.081E−01	+	3.027E+01 2.799E+00	+	3.745E+01 5.518E−01	+	2.184E+01 3.215E+00
+ / ≈ / −			8 / 0 / 1		8 / 0 / 1		8 / 0 / 1		8 / 0 / 1

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

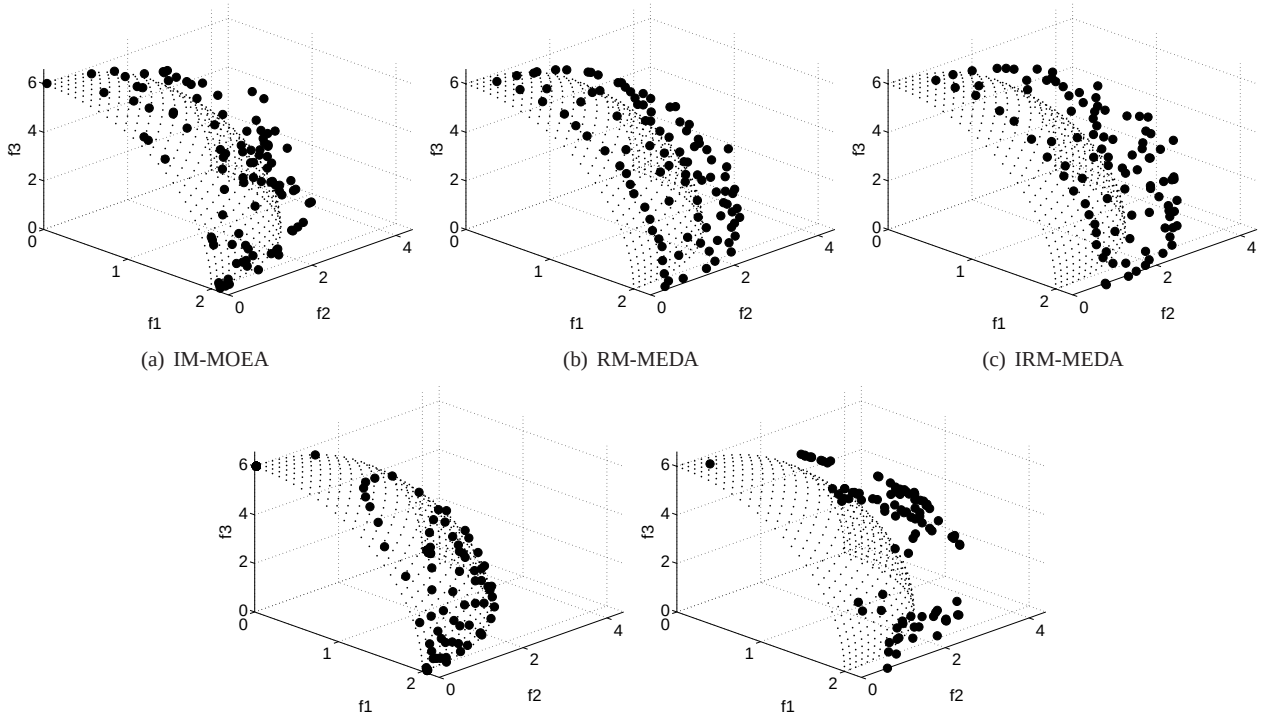
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation



(a) IM-MOEA (b) RM-MEDA (c) IRM-MEDA

(d) MOEA/D-DE (e) NSGA-II

Fig. 13. The non-dominated solutions with the best HV values obtained by each algorithm among 20 runs in the objective space on 104-D WFG5.



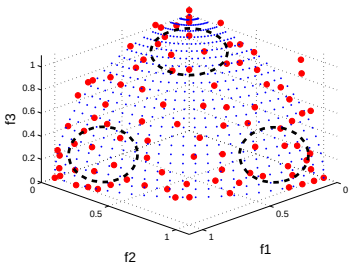
(a) IM-MOEA (b) RM-MEDA (c) IRM-MEDA

(d) MOEA/D-DE (e) NSGA-II

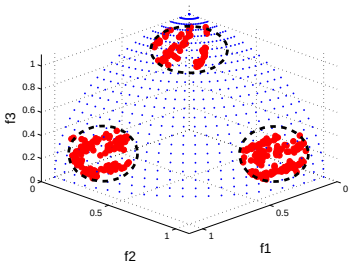
Fig. 14. The non-dominated solutions with the best HV values obtained by each algorithm among 20 runs in the objective space on 104-D WFG6.

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

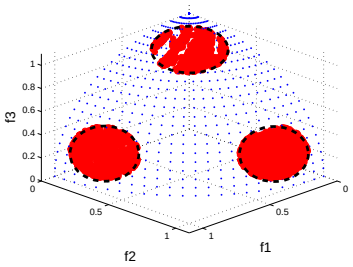
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation



(a) The 100 solutions obtained at the end run of IM-MOEA



(b) 300 solutions sampled using the inverse models trained on the 100 solutions in (a)



(c) 3000 solutions sampled using the inverse models trained on the 300 solutions in (b)

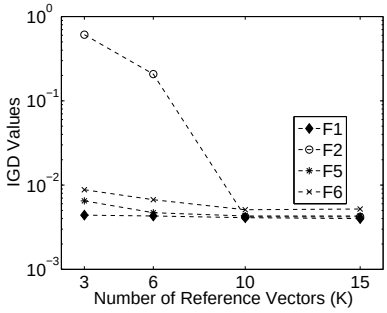
Fig. 15. An example of *a posteriori* sampling on F4. The dashed circles mark the interested regions on the PF.

K and the number of inverse models in each group L. In this subsection, we will analyze the sensitivity of IM-MOEA to these two parameters by using different settings on several typical test instances. Since the settings of K and L are independent, we will analyze their influence on the performance of IM-MOEA one by one.

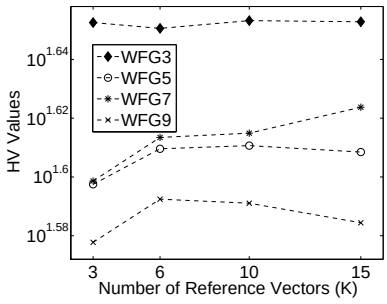
1) Sensitivity to the number reference vectors (K): As discussed in Section II-B, the number of reference vectors K directly determines the number of subpopulations, and the subpopulation size N_k can be calculated as $N_k = \lfloor \frac{N}{K} \rfloor$, where N is the size of the whole population. Inside each subpopulation, a number of $m \times L$ GP models are built based on the random grouping method introduced in Section II-C, where m is number of the objectives and L is the number of models in each group. Since there are m model groups in total and the GP models in each group are built using the related training data, in which the number of data points $N_t = \lfloor \frac{N_k}{m} \rfloor = \lfloor \frac{N}{m \times K} \rfloor$. For building a univariate GP, the minimum number of needed training data is two, i.e., $N_t \geq 2$ and

$$K \leq \lfloor \frac{N}{2m} \rfloor. \tag{23}$$

Give the population size $N = 100$, the possible values of K for bi-objective and three-objective problems are $K \in \{1, ..., 25\}$ and $K \in \{1, ..., 16\}$, respectively. Practically, for three-objective problems, $K \in 3, 6, 10, 15$ will be considered. In the following, we examine the performance of IM-MOEA for these four K values.



(a)



(b)

Fig. 16. The IGD and HV values averaged over 20 runs obtained by IM-MOEA with different numbers of reference vectors, K. (a) IGD values on F1, F2, F5 and F6; (b) HV values on 34-D WFG3, WFG5, WFG7, WFG9.

As shown by Fig. 16, IM-MOEA is not very sensitive to K in most cases, although $K = 3$ seems to be too small. According to the above findings, we recommend that $K = 10$.

2) Sensitivity to the number of models in each group (L): The number of models in each group, L, is usually much smaller than the maximum value, n. However, if L is too small, the random grouping method may lose effectiveness as the information of variable linkages (correlations) can be lost. In the following, we investigate the performance of IM-MOEA on four typical 104-D WFG instances, when $L \in \{1, 3, 5, 7\}$, respectively.

It can be seen from Fig. 17 that IM-MOEA is not very sensitive to the settings of L as long as $L \geq 3$. Accordingly, we recommend $L = 3$.

B. Runtime

Model-based methods tend to be more time consuming than traditional MOEAs as the model training procedures typically take more time than traditional evolutionary operators like crossover and mutation. In this subsection, we compare the runtime of the proposed IM-MOEA with that of RM-MEDA, which is also a model-based method. Here, we check the

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

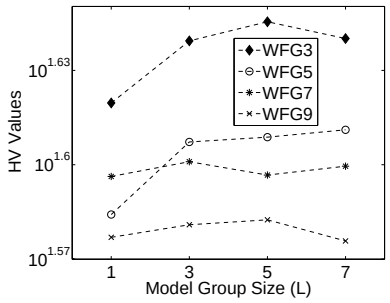


Fig. 17. The HV values averaged over 20 runs obtained by IM-MOEA with different settings of L on 104-D WFG3, WFG5, WFG7 and WFG9.

elapsed runtime on a bi-objective test instance (F2) and a three-objective test instance (F4) with different numbers of decision variables. All simulations are implemented using Matlab R2012a with Microsoft Windows 7 Enterprize SP1 64-bit as the operating system on a PC with an Intel Core i5-2500 3.3GHz CPU.

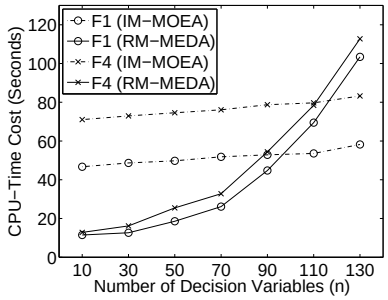


Fig. 18. Runtimes of IM-MOEA and RM-MEDA averaged over 20 runs on F1 and F4 with different numbers of decision variables. The maximum fitness evaluation number is 100,000 for each single run.

As shown in Fig. 18, the runtime of IM-MOEA is higher than that of RM-MEDA when the number of decision variables is smaller than 110. However, we can also see that the runtime IM-MOEA increases only very slowly as the number of decision variables increases, while the runtime of RM-MEDA increases very rapidly.

C. Influence of the mutation operator

In order to investigate the influence of the mutation operator on the performance of the proposed IM-MOEA, we have compared two variants of IM-MOEA, one using the mutation operator (same as in the above experiments), and the other not (denoted as IM-MOEA* hereafter). The comparisons are conducted on 30-D F1 to F10 and 104-D WFG test instances.

As evident from Table VI, among the ten test problems, IM-MOEA* outperforms IM-MOEA on four of them, performs comparably well with IM-MOEA on five of them, and is outperformed by IM-MOEA only on F10. Two observations can be made from these results. First, the excellent performance of IM-MOEA can mainly be attributed to the inverse models rather than the mutation operators. Second, a mutation operator is definitely helpful for optimization problems that are highly multimodal and non-uniform.

TABLE VI
STATISTICAL RESULTS OF IGD VALUES OBTAINED BY IM-MOEA AND IM-MOEA* (THE VERSION WITHOUT A MUTATION OPERATOR) ON 30-D F1 TO F10.

30-D	IM-MOEA		IM-MOEA*
F1	4.044E−03 4.573E−05	≈	4.006E−03 5.120E−05
F2	4.261E−03 6.013E−05	−	4.089E−03 3.039E−05
F3	2.199E−03 1.108E−04	≈	2.137E−03 7.241E−05
F4	7.049E−02 2.290E−03	−	6.129E−02 2.763E−03
F5	4.357E−03 7.131E−05	≈	4.405E−03 8.046E−05
F6	5.127E−03 1.709E−04	−	4.826E−03 8.649E−05
F7	2.703E−03 3.672E−04	≈	2.986E−03 2.398E−04
F8	8.147E−02 4.396E−03	−	7.018E−02 2.687E−03
F9	5.372E−03 1.053E−03	≈	5.317E−03 9.250E−04
F10	1.022E+00 4.395E−01	+	2.176E+00 1.052E+00
+ / ≈ / −			1 / 5 / 4

TABLE VII
STATISTICAL RESULTS OF HV VALUES OBTAINED BY IM-MOEA AND IM-MOEA* (THE VERSION WITHOUT A MUTATION OPERATOR) ON 104-D WFG TEST INSTANCES.

104-D	IM-MOEA		IM-MOEA*
WFG1	2.370E+01 1.207E+00	+	6.812E+00 2.038E+00
WFG2	6.194E+01 2.869E+00	≈	5.972E+01 3.939E+00
WFG3	4.359E+01 1.532E−01	≈	4.375E+01 3.321E−01
WFG4	4.078E+01 2.631E−01	≈	4.091E+01 3.237E−01
WFG5	4.048E+01 1.426E−01	≈	4.056E+01 1.486E−01
WFG6	3.719E+01 7.065E−01	≈	3.691E+01 8.927E−01
WFG7	3.991E+01 3.415E−01	≈	3.920E+01 4.927E−01
WFG8	3.925E+01 2.148E−01	≈	3.934E+01 2.268E−01
WFG9	3.816E+01 1.130E+00	≈	3.864E+01 6.464E−01
+ / ≈ / −			1 / 8 / 0

Results in Table VII further confirm our observations made above. We can see that on the nine three-objective WFG problems, IM-MOEA and IM-MOEA* perform comparably well on eight of them. IM-MOEA* is outperformed by IM-MOEA only on WFG1, which is strongly non-uniform. For such problems, IM-MOEA is very likely to get stuck in local optimums in the absence of a mutation operator.

In practice, since it is impossible to know in advance how the fitness landscape is, it is advisable to integrate a mutation operator in the proposed IM-MOEA for more robust search performance.

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

D. Limitations and potential remedies

Although IM-MOEA has been shown to be very promising on a wide range of test problems, it has its limitations, like most other algorithms. One limitation comes from the reference vectors. In this work, the reference vectors are uniformly generated on a hypersphere. As shown by our experimental results, the uniform reference vectors work very well on most MOPs. However, uniformly distributed reference vectors may encounter big challenges if the PF has a very irregular distribution. For example, while the PF of most three-objective MOPs is a 2-D surface, the PF of three-objective WFG3 is a degenerate curve, resulting in a very narrow distribution in the objective space. Actually, such MOPs may cause general difficulties for all MOEAs relying on predefined reference vectors or reference points in the objective space [77]. To address this difficulty, methods for adapting the reference vectors to the distribution of the Pareto optimal solutions during the search can be helpful [78].

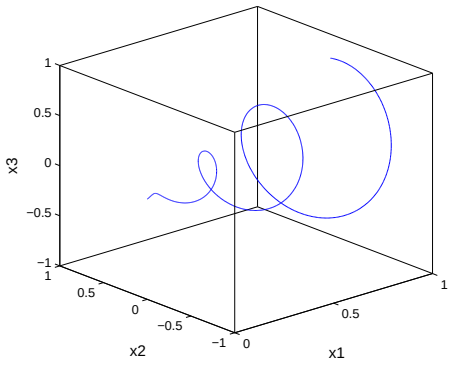
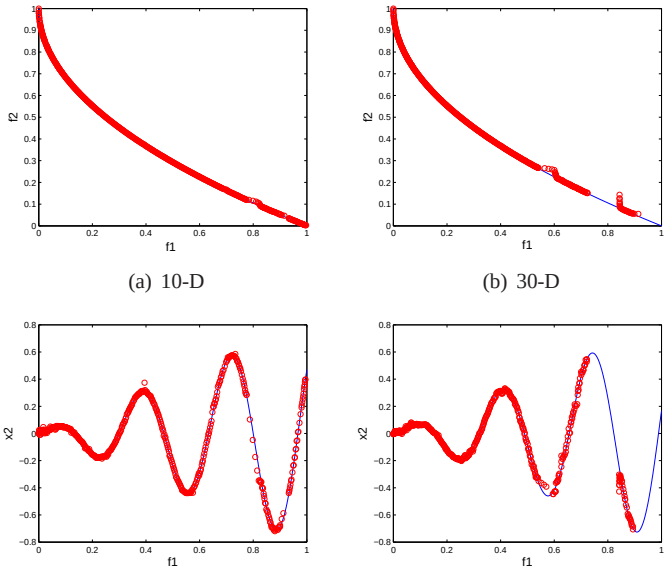


Fig. 19. An example of the Pareto set (x_1, x_2 and x_3) of test instance F3 introduced in [64].

In addition, IM-MOEA may perform poorly when the shape of a PS is extremely complex. For most MOPs as studied in this work, the shapes of the PSs are relatively smooth even if the decision variables are linearly or nonlinearly correlated. However, some MOPs, such as F3 in [64], may have a very complex PS. We have tested the performance of IM-MOEA on F3 and the results are presented in Fig. 20. We can see that when the decision space is of 10 dimensions, the PF approximated by IM-MOEA is acceptable, as shown in Fig. 20(a). However, when the dimension of the decision space increases to 30, there is a significant loss of solutions on the right end of the PF, as shown in Fig. 20 (b). To understand the reason for the loss of solutions, we have plotted the mapping from f_1 to x_2 in Fig. 20 (c) and 20 (d) for 10-D and 30-D, respectively. It can be seen that even if randomly uniformly distributed samples are collected in the objective space, very few samples fall in some particular regions in the decision space, e.g., probably because of the non-uniformity of the PF and the extreme non-smoothness in some intervals of the inverse model. One possible way to improve the performance of IM-MOEA for this type of MOPs is to increase the population size and the number of models. However, the performance improvement will be definitely at the cost of extra computational resources.



(a) 10-D (b) 30-D

(c) 10-D (d) 30-D

Fig. 20. The best PS and PF obtained using IM-MOEA among 20 single runs on 10-D and 30-D F3 introduced in [64], respectively. The population size is set to 500, and the number of reference vectors is set to 50. For each single run, a number of 500,000 fitness evaluations are implemented

V. Conclusion

In this paper, an inverse model based multiobjective evolutionary algorithm (IM-MOEA) has been suggested. The idea is to build a number of inverse models that map sampled points in the objective space back to the decision space. The main benefit of having such inverse models compared to forward models is that samples can directly generated in the objective space in creating offspring, which is very convenient for generating solutions in preferred regions. The multivariate inverse model is decomposed into multiple univariate inverse models to make the model building easier, and the needed number of inverse models has been drastically reduced using a random grouping strategy. Each univariate inverse model is implemented using a Gaussian process, which is well suited for modeling probabilistic functional maps. The proposed IM-MOEA is demonstrated to perform robustly competitive on a variety of test instances compared to four representative MOEAs, as long as the shapes of the Pareto sets are not too complex. An additional benefit of IM-MOEA is that an arbitrary number of solutions can be generated using relatively little computational effort *a posteriori* by sampling using the built inverse models.

Future work includes how to exploit the inverse model based method for integrating user preferences during evolutionary multiobjective optimization. The computational efficiency of the present IM-MOEA is not ideal and it is worth further investigation to reduce the runtime in solving small to medium optimization problems. The scalability of IM-MOEA for solving large scale optimization problems remains to be examined. In addition, its potential capability to solve many-objective

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

optimization problems [79] and real-world problems [80] is to be demonstrated.

ACKNOWLEDGEMENTS

This work was supported in part by the Honda Research Institute Europe and the Joint Research Fund for Overseas Chinese, Hong Kong and Macao Scholars of the National Natural Science Foundation of China (Grant No. 61428302).

REFERENCES

[1] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

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

[3] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, 1998.

[4] Y. Jin, T. Okabe, and B. Sendhoff, "Adapting weighted aggregation for multiobjective evolution strategies," in *Proceedings of the Evolutionary multi-criterion optimization*. Springer, 2001, pp. 96–110.

[5] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?" in *Genetic and Evolutionary Computation Conference*, 2001, pp. 1042–1049.

[6] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proceedings of the Evolutionary Multi-Criterion Optimization*, 2001, pp. 82–95.

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

[8] Y. Mei, K. Tang, and X. Yao, "Decomposition-based memetic algorithm for multiobjective capacitated arc routing problem," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 151–165, 2011.

[9] Y. S. Liau, K. C. Tan, J. Hu, X. Qiu, and S. B. Gee, "Machine learning enhanced multi-objective evolutionary algorithm based on decomposition," in *Proceedings of the Intelligent Data Engineering and Automated Learning*. Springer, 2013, pp. 553–560.

[10] S. Gee, K. Tan, V. Shim, and N. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," *IEEE Transactions on Evolutionary Computation*, accepted in 2014.

[11] N. Srinivas and K. Deb, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1994.

[12] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proceedings of the IEEE Conference on Evolutionary Computation*. IEEE, 1994, pp. 82–87.

[13] C. M. Fonseca, P. J. Fleming *et al.*, "Genetic algorithms for multiobjective optimization: Formulationdiscussion and generalization," in *Proceedings of the International Conference on Genetic Algorithms*, vol. 93, 1993, pp. 416–423.

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

[15] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[16] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Evolutionary Methods for Design, Optimisation and Control*, 2002, pp. 95–100.

[17] D. W. Corne, J. D. Knowles, and M. J. Oates, "The Pareto envelope-based selection algorithm for multiobjective optimization," in *Parallel Problem Solving from Nature PPSN VI*. Springer, 2000, pp. 839–848.

[18] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates *et al.*, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2001.

[19] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.

[20] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithmsa comparative case study," in *Parallel Problem Solving from Nature (PPSN V)*. Springer, 1998, pp. 292–301.

[21] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[22] D. Brockhoff and E. Zitzler, "Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 2086–2093.

[23] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima, "Indicator-based evolutionary algorithm with hypervolume approximation by achievement scalarizing functions," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2010, pp. 527–534.

[24] E. Zitzler, D. Brockhoff, and L. Thiele, "The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration," in *Proceedings of the Evolutionary Multi-Criterion Optimization*. Springer, 2007, pp. 862–876.

[25] N. Beume, C. M. Fonseca, M. López-Ibáñez, L. Paquete, and J. Vahrenhold, "On the complexity of computing the hypervolume indicator," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1075–1082, 2009.

[26] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2012.

[27] J. Bader, K. Deb, and E. Zitzler, "Faster hypervolume-based search using monte carlo sampling," in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*. Springer, 2010, pp. 313–326.

[28] Y. Jin and B. Sendhoff, "Connectedness, regularity and the success of local search in evolutionary multi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 2003, pp. 1910–1917.

[29] H. Ishibuchi and Y. Shibata, "An empirical study on the effect of mating restriction on the search ability of emo algorithms," in *Proceedings of the Evolutionary Multi-criterion Optimization*, 2003, pp. 433–447.

[30] A. Jaszkiewicz, "Genetic local search for multiple objective combinatorial optimization," *European Journal of Operational Research*, vol. 137, no. 1, pp. 50–71, 2002.

[31] O. Schütze, S. Mostaghim, M. Dellnitz, and J. Teich, "Covering Pareto sets by multi-level evolutionary subdivision techniques," in *Proceedings of the Evolutionary Multi-criterion Optimization*, 2003, pp. 118–132.

[32] P. Larrañaga and J. A. Lozano, *Estimation of distribution algorithms: A new tool for evolutionary computation*. Springer, 2002.

[33] M. Pelikan, K. Sastry, and D. E. Goldberg, "Multiobjective estimation of distribution algorithms," in *Scalable optimization via probabilistic modeling*. Springer, 2006, pp. 223–248.

[34] L. Martí, J. Garcia, A. Berlanga, C. Coello Coello, and J. M. Molina, "On current model-building methods for multi-objective estimation of distribution algorithms: Shortcommings and directions for improvement," *Department of Informatics, Universidad Carlos III de Madrid, Madrid, Spain, Technical Report GIAA2010E001*, 2010.

[35] M. Costa and E. Minisci, "Moped: a multi-objective parzen-based estimation of distribution algorithm for continuous problems," in *Proceedings of the Evolutionary Multi-Criterion Optimization*. Springer, 2003, pp. 282–294.

[36] L. Martí, J. García, A. Berlanga, and J. M. Molina, "Solving complex high-dimensional problems with the multi-objective neural estimation of distribution algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2009, pp. 619–626.

[37] V. A. Shim, K. C. Tan, and C. Y. Cheong, "An energy-based sampling technique for multi-objective restricted boltzmann machine," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, pp. 767–785, 2013.

[38] T. Okabe, Y. Jin, B. Sendoff, and M. Olhofer, "Voronoi-based estimation of distribution algorithm for multi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 2. IEEE, 2004, pp. 1594–1601.

[39] D. Thierens and P. A. Bosman, "Multi-objective mixture-based iterated density estimation evolutionary algorithms," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2001, pp. 663–670.

[40] P. A. Bosman and D. Thierens, "The naive MIDEA: A baseline multi–objective EA," in *Proceedings of the Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 428–442.

[41] H. Karshenas, R. Santana, C. Bielza, and P. Larranaga, "Multi-objective estimation of distribution algorithm based on joint modeling of objec-

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation

tives and variables," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 519–542, 2014.
[42] ——, "Regularized continuous estimation of distribution algorithms," *Applied Soft Computing*, vol. 13, pp. 2412–2432, 2013.
[43] T. Okabe, Y. Jin, and B. Sendhoff, "On the dynamics of evolutionary multi-objective optimisation," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2002, pp. 247–255.
[44] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.
[45] M. Dellnitz, O. Schütze, and T. Hestermeyer, "Covering Pareto sets by multilevel subdivision techniques," *Journal of Optimization Theory and Applications*, vol. 124, no. 1, pp. 113–136, 2005.
[46] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.
[47] N. Kambhatla and T. K. Leen, "Dimension reduction by local principal component analysis," *Neural Computation*, vol. 9, no. 7, pp. 1493–1516, 1997.
[48] I. Giagkiozis and P. J. Fleming, "Increasing the density of available pareto optimal solutions," *Research Report No. 1028 at University of Sheffield*, 2012.
[49] ——, "Pareto front estimation for decision making," *Evolutionary Computation*, accepted in 2014.
[50] C. E. Rasmussen, *Gaussian processes for machine learning*. MIT Press, 2006.
[51] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.
[52] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences*, vol. 178, no. 15, pp. 2985–2999, 2008.
[53] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011.
[54] B. Liu, Q. Zhang, and G. Gielen, "A gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 180–192, 2014.
[55] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft computing*, vol. 9, no. 1, pp. 3–12, 2005.
[56] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 2450–455, 2014.
[57] J. A. Cornell, *Experiments with mixtures: designs, models, and the analysis of mixture data*. John Wiley & Sons, 2011.
[58] E. L. Snelson, "Flexible and efficient Gaussian process models for machine learning," Ph.D. dissertation, Citeseer, 2007.
[59] Y. Jin and J. Branke, "Evolutionary optimization in uncertain environments-a survey," *IEEE Transactions on Evolutionary Computation*, vol. 9, no. 3, pp. 303–317, 2005.
[60] P. Campigotto, A. Passerini, and R. Battiti, "Active learning of Pareto fronts," *IEEE Transactions on Neuralnetworks and Learning Systems*, vol. 25, no. 3, pp. 506–519, 2014.
[61] A. Zhou, Q. Zhang, and G. Zhang, "A multiobjective evolutionary algorithm based on decomposition and probability model," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–8.
[62] K. Deb *et al.*, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons Chichester, 2001, vol. 2012.
[63] Y. Wang, J. Xiang, and Z. Cai, "A regularity model-based multiobjective estimation of distribution algorithm with reducing redundant cluster operator," *Applied Soft Computing*, vol. 12, no. 11, pp. 3526–3538, 2012.
[64] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.
[65] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.
[66] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
[67] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proceedings of the Congress on Evolutionary Computation*. IEEE, 2002, pp. 825–830.
[68] A. Zhou, Q. Zhang, Y. Jin, E. Tsang, and T. Okabe, "A model-based evolutionary algorithm for bi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 2005, pp. 2568–2575.
[69] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Information Sciences*, vol. 182, no. 1, pp. 220–242, 2012.
[70] Z.-H. Zhan, J. Li, J. Cao, J. Zhang, H.-H. Chung, and Y.-H. Shi, "Multiple populations for multiple objectives: a coevolutionary technique for solving multiobjective optimization problems," *IEEE Transactions on Cybernetics*, vol. 43, no. 2, pp. 445–463, 2013.
[71] Y.-Y. Tan, Y.-C. Jiao, H. Li, and X.-K. Wang, "A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets," *Information Sciences*, vol. 213, pp. 14–38, 2012.
[72] R. Storn and K. Price, "Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
[73] M. Fazzolari, R. Alcala, Y. Nojima, H. Ishibuchi, and F. Herrera, "A review of the application of multiobjective evolutionary fuzzy systems: Current status and further directions," *IEEE Transactions on Fuzzy Systems*, vol. 21, no. 1, pp. 45–65, 2013.
[74] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.
[75] E. K. Burke, M. Gendreau, M. Hyde, G. Kendall, G. Ochoa, E. Özcan, and R. Qu, "Hyper-heuristics: A survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.
[76] K. Deb, "Multi-objective optimization," in *Search Methodologies*. Springer, 2014, pp. 403–449.
[77] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
[78] ——, "An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization," in *Proceedings of the Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 307–321.
[79] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review." in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2008, pp. 2419–2426.
[80] S. Nguyen, M. Zhang, M. Johnston, and K. C. Tan, "A computational study of representations in genetic programming to evolve dispatching rules for the job shop scheduling problem," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 621–639, 2013.



Ran Cheng received the B. Eng. degree from Northeastern University, Shenyang, China, in 2010. He is currently working towards the Ph.D. degree at the Department of Computing, University of Surrey, United Kingdom.

His current research interests include multi-/many-objective optimization, estimation of distribution algorithms, and swarm intelligence.

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2015.2395073, IEEE Transactions on Evolutionary Computation



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

He is a Professor of Computational Intelligence, Department of Computing, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He is also a Finland Distinguished Professor funded by the Finnish Agency for Innovation (Tekes) and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, China. His science-driven research interests lie in the interdisciplinary areas that bridge the gap between computational intelligence, computational neuroscience, and computational systems biology. He is also particularly interested in nature-inspired, real-world driven problem-solving. Dr Jin has (co)edited five books and three conference proceedings, authored a monograph, and (co)authored over 200 peer-reviewed journal and conference papers. He has been granted eight US, EU and Japan patents. His current research is funded by EC FP7, UK EPSRC and industry. He has delivered 20 invited keynote speeches at international conferences. He is an Associate Editor or Editorial Board Member of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NANOBIOSCIENCE, IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, Evolutionary Computation, BioSystems, the International Journal of Fuzzy Systems, Soft Computing, and Natural Computing.

Dr Jin is an IEEE Distinguished Lecturer and Vice President for Technical Activities of the IEEE Computational Intelligence Society. He was the recipient of the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology and the 2014 IEEE Computational Intelligence Magazine Outstanding Paper Award. He is a Fellow of British Computer Society and Senior Member of IEEE.



Kaname Narukawa received B.Eng. and M.Eng. degrees in Industrial Engineering from Osaka Prefecture University, Japan, in 2004 and 2005, respectively. He was a member of Honda Research Institute Europe, Germany. His current research interests include evolutionary computation, multi- and many-objective optimization.



Bernhard Sendhoff obtained a PhD in Applied Physics in May 1998, from the Ruhr-Universität Bochum, Germany. From 1999 to 2002 he worked for Honda R&D Europe GmbH, and since 2003, he has been with the Honda Research Institute Europe GmbH. Since 2007 he is Honorary Professor of the School of Computer Science of the University of Birmingham, UK. Since 2008, he is Honorary Professor at the Technical University of Darmstadt, Germany. Since 2011 he is President of the Honda Research Institute Europe GmbH. Bernhard Sendhoff is a senior member of the IEEE and of the ACM. His research focuses on methods from computational intelligence and their applications in development, production and services. He has authored and co-authored over 150 scientific papers and over 30 patents.

1089-778X (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.