



An inverse model-based multiobjective estimation of distribution algorithm using Random-Forest variable importance methods

Pezhman Gholamnezhad | Ali Broumandnia | Vahid Seydi

Department of Computer, South Tehran Branch, Islamic Azad University, Tehran, Iran

Correspondence

Ali Broumandnia, Department of Computer, South Tehran Branch, Islamic Azad University, Ahang St, Blv South-Nabard, Tehran, Iran
Email: broumandnia@gmail.com

Abstract

Most existing methods of multiobjective estimation of distributed algorithms apply the estimation of distribution of the Pareto-solution on the decision space during the search and little work has proposed on making a regression-model for representing the final solution set. Some inverse-model-based approaches were reported, such as inversed-model of multiobjective evolutionary algorithm (IM-MOEA), where an inverse functional mapping from Pareto-Front to Pareto-solution is constructed on nondominated solutions based on Gaussian process and random grouping technique. But some of the effective inverse models, during this process, may be removed. This paper proposes an inversed-model based on random forest framework. The main idea is to apply the process of random forest variable importance that determines some of the best assignment of decision variables (x_n) to objective functions (f_m) for constructing Gaussian process in inversed-models that map all nondominated solutions from the objective space to the decision space. In this work, three approaches have been used: classical permutation, Naïve testing approach, and novel permutation variable importance. The proposed algorithm has been tested on the benchmark test suite for evolutionary algorithms [modified Deb K, Thiele L, Laumanns M, Zitzler E (DTLZ)

and Walking Fish Group (WFG)] and indicates that the proposed method is a competitive and promising approach.

KEY WORDS

estimation of distribution algorithm, Gaussian process, inverse modeling, multiobjective optimization, random forest variable importance

1 | INTRODUCTION

1.1 | Background

Random forests are a general tool for classification with high dimensional data and apply for ranking candidate predictors based on variable importance measure. They provide two simple methods for feature selection: mean decrease impurity (MDI) and mean decrease accuracy. Also there are two importance measure for ranking predictor variables: the Gini importance and the permutation importance. In Gini importance, with removing any association between the response and a predictor variable, the prediction accuracy is changed and large change illustrates that the predictor variable is important. But there is no natural cut off that can be discriminated important and nonimportant variables. For this purpose, some methods were proposed that are permutation-based and need the repeated computation of forests. That's it for high-dimensional setting typically containing thousands of candidate predictor, which have a very high computing time. Also, the predictor variables contains variables that do not have their "own" effect on the response, but are related with the response due to their relationship with truly influential predictor variables. Regarding the values variable importance, the negative values or values of zero, illustrate that variable does not improve the trees' predictive strength, because the error rates are similar or even larger when applying unchanged version of the variable. In contrast, a positive value for the variable importance, express that the variable at least some improves the trees' predictive strength, because the error rates are smaller when applying the original version of the variable for deriving tree predictions. However it cannot be deduced that a positive value for the variable importance demonstrates a relative value. A new computationally fast heuristic procedure of a variable importance test is applied that is suitable for high dimensional data where many variables do not carry any information. This approach is based on a modified version of permutation variable importance measure which is taken from cross-validation procedure.¹

The performances of multiobjective evolutionary algorithms (MOEAs) depend partly on the genetic operator or selection strategy.^{2,3} For this purpose, estimation of distribution algorithms (EDAs) was developed to solve optimization problems.⁴⁻⁶ The basic idea of EDAs is making a stochastic model from the parental distribution in the decision-maker and to produce offspring individuals by sampling from the model. Instead of genetic operators, these algorithms produce new candidate solutions from a probabilistic model, which is learnt from a set of promising solutions.



Typically the EDA-based MOEA, based on their estimation models, are divided in to two groups: The first group is the Bayesian network-based EDAs and the second group is often known as the mixture probability model-based EDAs. The main difference between EDAs and most arbitrary evolutionary algorithms is that in evolutionary algorithms new candidate solutions are produced using an implicit distribution, defined by one or more variation operators, whereas EDAs use explicit probability distribution encoded by a Bayesian network, a multivariate normal distribution, or another model class. Most multiobjective EDAs, concentrate on the estimation distribution of the PS in the decision space in search and represent the final Pareto optimal solutions, in template of the set, such as an archive. But they still had some challenge such as time consuming, and have strict requirement of the training data, and with increase of dimensionality the decision variable, the performance sharply is decreased. Hence In recent years, the surrogate-assisted evolutionary algorithms (SAEAs) have been applied in order to solve the multiobjective EDAs.⁷ The surrogate model is used when evaluating of fitness is computationally and cannot be easily measured and it is necessary simulation to evaluate. But they still had some challenges such as the choice of surrogate model, Problem of increasing dimension in decision variables and it is impossible to determine what should be predicted by the surrogate. But little work has been reported on making a regression model for showing the final solution set. Some primary work has reported where piecewise linear models were made to represent the solution set attained by an MOEA, leading the improve quality of the solutions.⁸ Most recently, an inverse functional mapping from Parento-Front (PF) to Pareto-solution (PS) was made based on the approximate PS generated by MOEAs at the end of evolutionary optimization.^{9,10} In this method, the inverse model is not used during the optimization. Also, the inverse modeling based on MOEA (IM-MOEA), was proposed that mapped nondominated solutions from the object space (PF) to the decision space (PS).¹¹ In this method, the m -input n -output multivariate inverse model is parsed in to $m \times n$ multivariate models, which considerably facilitates the model making and removes the requirement for dimensionality reduction. Then each univariate inverse model, discovered by a Gaussian process and a random grouping method.^{12,13}

1.2 | Related work

Many multiobjective EDAs are suggested from single objective EDAs, but from the first group (eg, the Bayesian network-based EDAs), for example, multiobjective Bayesian optimization algorithm (BOA) was proposed to create a Bayesian network as a model in order to generate offspring.¹⁴ A related work was proposed to predict the model by augmentation of Pareto ranking approach and BOA.¹⁵ Also, a Bayesian multiobjective was proposed whose model is made over the solutions selected by ϵ -Pareto ranking method.¹⁶ In addition, in order to select a subset of solutions serving for a regression decision tree, an improved nondominated sorting approach was applied by decision tree-based multiobjective EDA to learn the model.¹⁷ Recently, in order to addressing many objective optimization problems, the multidimensional Bayesian network was suggested.¹⁸ In second group (eg, the mixture probability model-based EDAs), the multiobjective mixture-based iterated density estimation evolutionary algorithm using the mixed probability distributions in order to sampling well-distributed solutions.¹⁹ The multiobjective Parzen-based EDA learns from the Gaussian and Cauchy kernels to create its model.²⁰ The multiobjective hierarchical BOA for discrete multiobjective optimization problems (MOPs) was proposed.²¹ Also, the multiobjective extended compact genetic algorithm applied a marginal product model as the mixture of the

probability model.²² Furthermore, the regularity model-based multiobjective estimation of distribution algorithm (RM-MEDA), was suggested which the decision vectors are mapped from the n -dimensional decision space to the ($M-1$)-D hidden space using a local principal component analysis.² In recent years, the SAEAs have been applied in order to solve the multiobjective EDAs.⁷ Recently, Pan et al suggested a classification-based surrogate-assisted evolutionary algorithm (CSEA) that applies an artificial neural network instead of approximating the objective values in order to learn the dominance relationship between a candidate solution and a set of reference solutions.²³ Also, the IM-MOEA, was proposed that mapped nondominated solutions from the object space (PF) to the decision space (PS).¹¹ This algorithm is completely different from the SAEAs.^{24,25} The SAEAs build a map from the decision space to objective space and estimate the fitness of candidate solutions and helpful when no explicit fitness function exist or fitness functions are expensive computationally. While, the inverse models, have been suggested to approximate the distribution of the Pareto optimal solutions. Also a random grouping method used in this process (IM-MOEA) for dealing with large-scale optimization in order to reduce the number of required inverse models.^{26,27} But some of the effective inverse models, during the random grouping, may be removed. Also, random grouping method, removes the chance of allocated decision variables, to other objective spaces randomly. In addition, this method cannot determine the best assignment of decision variables (x_n) to objective functions (f_m). The idea of IM-MOEA has extended in order to solve MOPs with irregular PFs in a dynamic environment.²⁸ In this model instead of using Gaussian process, a simple linear model has been applied in order to simplify the inverse modeling process. Also some approaches has concentrated on the PF modeling only, such as the Pareto-adaptive ϵ -dominance-based algorithm.²⁹ The rest of this paper is formed as follows: The next section is motivation and contributions. The problem statement is presented in Section 3 and briefly is reviewed some background necessary in the paper. The proposed algorithm is described in Section 4. Section 5 illustrates experimental analysis of applying algorithm, results, and computational complexity. Finally, conclusions and future work are given in Section 6.

2 | MOTIVATION AND CONTRIBUTIONS

It can be seen that the basic idea in these algorithms is the replacement of heuristic operators with machine learning models such as regression models (eg, the Gaussian process), clustering models or classification models. So, recently, the model-based evolutionary algorithms (MBEAs) have been suggested in three groups:⁶ The EDAs, surrogate-assisted evolutionary algorithms and the inverses-models to map from the objective space to the decision space. In this paper, we concentrate on the inverses-model from the MBEAs and a new approach, based on an inverses model of Gaussian process and random forest framework, is proposed: random forest variable importance based on the inverses model of estimation of distribution algorithm (RFVI-IMEDA) in which a Gaussian process and random forest variable importance algorithm^{30,31} is used for mapping nondominated solutions from the objective space (PF) to the decision space (PS). Briefly, the Gaussian process is applied for estimating fitness values and random forest feature importance method is used to determine the best assignment of decision variables (x_n) to objective functions (f_m) in inverse model process. In fact, the importance of a variable x_n in decision space for predicting f_m is determined by random forest feature importance and variables are selected that are most importance. Therefore, the suggested method depends on a group of multiobjective EDAs that use probabilistic models instead of genetic conversions to generate offspring.



3 | PROBLEM STATEMENT

The offspring in common EDAs are generated by creating and sampling a probabilistic model of promising solutions in the decision space. Then these samples mapped from decision space to objective space via the objective functions. In IM-MOEA, given the distributions of current parents in the decision space, namely $P(X^P)$, and in the objective space, $P(Y^P)$, the conditional probability distribution of $P(X^P|Y^P)$, is estimated and can be approximated by a probabilistic inverse model, that mapped nondominated solutions from the object space (PF) to the decision space (PS). To generate offspring, this method, produces samples in the objective space based on the information of the objective values of the current parent population, marked by Y^o . Then Y^o , using $P(X^P|Y^P)$ are mapped back to the decision space using the Bayes' theorem that has been described in Section 3.2:

$$P(X^o) = \frac{P(X^P | Y^P)P(Y^o)}{P(Y^P | X^P)}, \quad (1)$$

where $P(X^P|Y^P)$, is a priori knowledge. These offspring are incorporated with the current parent population, from which parents for the next generation will be selected.¹¹ It is hard to estimate the m -input and n -output inverse model $P(X^P|Y^P)$, where m and n are, respectively, the number of objectives and decision variables. As regards all decision variables are independent of each other. This method uses random grouping strategy, where multiple decision variables are randomly grouped together to be derived from the same objective using inverse models. It divides all of objective space into a number of subspaces, which can be approximated as follows:

$$P(X|Y) \approx \prod_{i=1}^N (P(x_i | f_j) + \epsilon_{j,i}), \quad (2)$$

where $j = 1, 2, \dots, m$, $m > 2$, and it is assumed that $\epsilon(j, i) \sim N(0, (\delta_n)^2)$, is a Gaussian noise. As a result, The Gaussian process can be applied to the inverse models that has been described in Section 3.1. In this way:

$$P(x_i | f_j) = N(0, C + (\sigma_n)^2 I) \quad (3)$$

In order to raise the population diversity, polynomial mutation operator is used to all the sampled individuals.³² Figure 1 shows the framework of IM-MOEA.

In IM-MOEA, some of the effective inverse models, during this process, may be removed. For example, a 10-dimensional three objective optimization problem, three groups of models will be created. If $L = 2$, due to the random grouping method, x_2, x_4 , belonged to group 1, x_1, x_6 , to group 2, and x_5, x_8 , to the third group. Thus the first group includes the following two inverse models: $p(x_2|f_1)$, $p(x_4|f_1)$, the second group includes $p(x_1|f_2)$, $p(x_6|f_2)$, and the third group includes $p(x_5|f_3)$, $p(x_8|f_3)$. As a result, the values of the allocated decision variables (x_1, x_2, x_4, x_5, x_6 and x_8) will be change with new values created by the inverse models, and the values of the other four decision variables (x_3, x_7, x_9 and x_{10}) will stay unchanged. In this example, when x_2, x_4 allocated to f_1 , the chances of assigning x_2 and x_4 to f_2 and f_3 , disappear. Also, when x_1 and x_6 allocated to f_2 , the chances of assigning x_5 and x_8 , to f_1 and f_2 , disappear. Thus random grouping method, removes the chance of allocated decision variables, to other objective spaces randomly. In addition, this method cannot determine the best assignment of x_n to f_m and its function is based on random

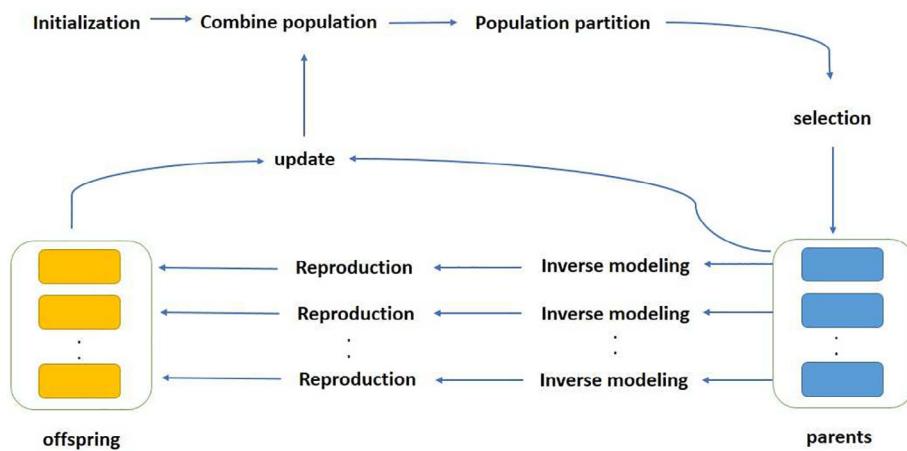


FIGURE 1 The framework of inversed-model of multiobjective evolutionary algorithm [Color figure can be viewed at wileyonlinelibrary.com]

assignment. The random forests methods can be used to identify predictor variables are the most important and termed as random forest feature importance.³³ In this section, some background concepts are presented.

3.1 | Gaussian process

A Gaussian process is a set of random variables indexed by time or space, such that every finite set of those random variables has a multivariate normal distribution, that is, every finite linear mixture of them is normally distributed.³⁴ The distribution of a Gaussian process is the joint distribution of all those (infinitely many) random variables and also it is a distribution over functions with a continuous domain, for example, time or space, that is mean GP defines a distribution over functions of the form $f : x \rightarrow \mathbb{R}$, which is completely specified by mean function $\mu(x)$ and covariance function $k(x, x')$ using the following equation:

$$f(x) \sim \text{GP}(\mu(x), k(x, x')). \quad (4)$$

In probability theory and statistics, covariance is a measure of how much two variables change together, and the covariance function or kernel, describes the spatial or temporal covariance of a random variable process or field. For a random field or stochastic process $Z(x)$ on a domain D , a covariance function $C(X, Y)$ gives the covariance of the values of the random field at the two locations X and Y using the following equation:

$$C(x, y) = \text{cov}(Z(x), Z(y)) \quad (5)$$

The same $C(x, y)$ is called auto covariance function in two instances: in time series (to indicate exactly the same concept except that x and y refer to locations in time rather than in space), and in multivariate random fields (to refer to the covariance of a variable with itself, as opposed to the cross covariance between two different variables at different locations, $\text{cov}(Z(x_1), Y(x_2))$).³⁵



A function is a valid covariance function if and only if this variance is nonnegative for all possible choices of N and weights w_1, \dots, w_n . A function with this property is called positive definite.³⁶ Ultimately Gaussian processes translate as taking priors on functions and the smoothness of these priors can be induced by the covariance function. If we expect that for “near-by” input points x and x' their corresponding output points y and y' to be “near-by” also, then the assumption of continuity is present. If we wish to allow for significant displacement then we might choose a rougher covariance function.³⁷ There are a common covariance functions, such as constant, linear, Gaussian noise, squared exponential, Orenstein-Uhlenbeck, Matern, periodic and rational quadratic.³⁸

3.2 | Bayesian linear regression

In statistics, linear regression is a linear scheme for modeling the relationship between a scalar dependent variable y and one or more expository variables (or independent variables) defined x . The case of expository variable is termed simple linear regression. For more than one expository variable, the process is termed multiple linear regression.³⁹ In linear regression, the communications are patterned using linear predictor functions whose unknown model parameters are approximated from the data. Here, let $S = \{x^{(i)}, y^{(i)}\}_{i=1}^n$ be a training set of i.i.d examples from some unknown distribution. The regression model is described by using the following equation:

$$y = f(X) + \epsilon. \quad (6)$$

The goal is to estimate the regression function $f(\cdot)$, to make prediction of $y_* = f(X_*)$, with parametric (Maximum Likelihood, Bayesian inference [parametric model $f_w(x)$]) or nonparametric methods (Kernel regression [Nadaraya-Watson estimator], GP regression), which in linear regression using the following equation:

$$y = XW + \epsilon, \quad (7)$$

where

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}, \quad X = \begin{pmatrix} X_1^T \\ X_2^T \\ \vdots \\ X_n^T \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \dots & x_{1p} \\ 1 & x_{21} & \dots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \dots & x_{np} \end{pmatrix}, \quad W = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{pmatrix}, \quad \epsilon = \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{pmatrix}. \quad (8)$$

The Bayesian linear regression is a procedure to linear regression in which the statistical analysis is assumed within the context of Bayesian inference. When the regression pattern has error that have a normal distribution, and if a particular form of prior distribution is assumed, explicit results are available for the posterior probability distributions of the model's parameters.⁴⁰ In Equation (7), ϵ , is a normal with mean zero and δ^2 variance, $N(0, \delta^2 I)$, and W is a multivariate Gaussian, and let $Z_x = XW$, which Z_x , is a GP. These two variables are independent of each other and therefore $y = Z + \epsilon$, is multivariable Gaussian process. As a result: $y \sim N(\mu + 0, K + \delta^2 I)$,

where, $K + \delta^2 I$, is covariance. We can divide y in two parts $y_a = \begin{pmatrix} y_1 \\ \vdots \\ y_l \end{pmatrix}$ and $y_b = \begin{pmatrix} y_{l+1} \\ \vdots \\ y_n \end{pmatrix}$, Similarly we can consider the mean and variance as follows: $\mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}$ and $\Sigma^{-1} = K = \begin{pmatrix} k_{aa} & k_{ab} \\ k_{ba} & k_{bb} \end{pmatrix}$. As we know, in regression, we have some observed data at some X_i and y_i that's here y_b and y_a , is unobserved data. So, we like to get the posterior predictive distribution on the y 's corresponding to X_1, \dots, X_l , that given these y' s, y_{l+1}, \dots, y_n . A number of useful properties of multivariate Gaussian as follows:

- Normalization.** The density function normalizes, that is, $\int_x P(x, \mu, \Sigma) dx = 1$.
- marginalizaton.** The marginal densities,

$$P(x_A) = \int_{x_B} p(x_A, x_B; \mu, \Sigma) dx_B = \int_{x_A} p(x_A, x_B; \mu, \Sigma) dx_A, \quad (9)$$

are Gaussian:

$$x_A \sim N(\mu_A, \Sigma_{AA}), x_B \sim N(\mu_B, \Sigma_{BB}). \quad (10)$$

- Conditioning.**

The conditional densities:

$$p(x_A | x_B) = \frac{p(x_A, x_B; \mu, \Sigma)}{P(x_B)} = \frac{p(x_A, x_B; \mu, \Sigma)}{\int_{x_A} p(x_A, x_B; \mu, \Sigma) dx_A}. \quad (11)$$

Due to the Equation (4) and multivariate Gaussian process:

$$\begin{aligned} P(y_A | y_B) &= \frac{p(y_A, y_B; \mu, \Sigma)}{\int_{y_A} p(y_A, y_B; \mu, \Sigma) dy_A} \\ &= \frac{1}{\int_{y_A} p(y_A, y_B; \mu, \Sigma) dy_A} \cdot \left[\frac{1}{(2\pi)^m} \exp \left(-\frac{1}{2} (y - \mu)^T \Sigma^{-1} (y - \mu) \right) \right] \\ &= \frac{1}{Z_1} \exp \left\{ -\frac{1}{2} \left(\begin{bmatrix} y_A \\ y_B \end{bmatrix} - \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \right)^T \begin{bmatrix} V_{aa} & V_{ab} \\ V_{ba} & V_{bb} \end{bmatrix} \left(\begin{bmatrix} y_A \\ y_B \end{bmatrix} - \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \right) \right\}, \end{aligned} \quad (12)$$

where Z_1 is a proportionality constant which does not depend on y_a . To simplify this phrase:

$$\begin{aligned} \left(\begin{bmatrix} y_A \\ y_B \end{bmatrix} - \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \right)^T \begin{bmatrix} V_{aa} & V_{ab} \\ V_{ba} & V_{bb} \end{bmatrix} \left(\begin{bmatrix} y_A \\ y_B \end{bmatrix} - \begin{bmatrix} \mu_A \\ \mu_B \end{bmatrix} \right) = (y_A - \mu_A)^T V_{aa} (y_A - \mu_A) + (y_A - \mu_A)^T V_{ab} (y_B - \mu_B) \\ + (y_B - \mu_B)^T V_{ba} (y_A - \mu_A) + (y_B - \mu_B)^T V_{bb} (y_B - \mu_B). \end{aligned} \quad (13)$$



With preserving only sections dependent on y_a and using the rule that $V_{ab} = V_{ba}^T$, we have:

$$P(Y_a|Y_b) = \frac{1}{Z_2} \exp \left(-\frac{1}{2} [y_a^T V_{aa} y_a - 2y_a^T V_{aa} \mu_a + 2y_a^T V_{ab} (y_b - \mu_b)] \right), \quad (14)$$

where Z_2 is a new proportionality constant not depending on y_a . Finally, using the completion of squares argument,⁴¹ we have:

$$P(Y_a|Y_b) = \frac{1}{Z_3} \exp \left(-\frac{1}{2} (y_a - \mu')^T V_{aa} (y_a - \mu') \right), \quad (15)$$

where Z_3 is again a new proportionality constant which does not depend on y_a , and $\mu' = \mu_a + V_{ab} V_{bb}^{(-1)} (y_b - \mu_b)$. If $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \rightarrow A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix} = \frac{1}{(ad-bc)} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$. Thus:

$$\begin{bmatrix} V_{aa} & V_{ab} \\ V_{ba} & V_{bb} \end{bmatrix}^{-1} = \frac{1}{V_{aa} V_{bb} - V_{ab} V_{ba}} \begin{bmatrix} V_{bb} & -V_{ab} \\ -V_{ba} & V_{bb} \end{bmatrix}. \quad (16)$$

Given that, $C = K + \delta^2 I$, we have:

$$P(y_a|y_b) = N(y_a|\mu, D). \quad (17)$$

$$\mu = \mu_a + V_{ab} V_{bb}^{-1} (y_b - \mu_b) = \mu_a + k_{ab} (k_{bb} + \delta^2 I)^{-1} (y_b - \mu). \quad (18)$$

$$D = V_{aa} - V_{ab} V_{bb}^{-1} V_{ba} = (k_{aa} + \delta^2 I) - k_{ab} (k_{bb} + \delta^2 I)^{-1} k_{ba}. \quad (19)$$

4 | THE PROPOSED ALGORITHM

4.1 | Framework of the proposed algorithm

The IM-MOEA uses the random grouping method in order to parse high-dimensional optimization problems into a number of low dimensional subproblems, without a prior knowledge of the nonseparability of a problem. In this algorithm, we replace three methods of random forest variable importance with random grouping in order to prevent the elimination of some effective inverse models and losing the chance of allocated decision variables, to other objective spaces randomly:

1. The classical permutation variable importance random forest method that has been described in Section 4.2.
2. The random forest naïve variable importance method that has been described in Section 4.3.
3. The random forest cross-validation variable importance method that has been described in Section 4.4.

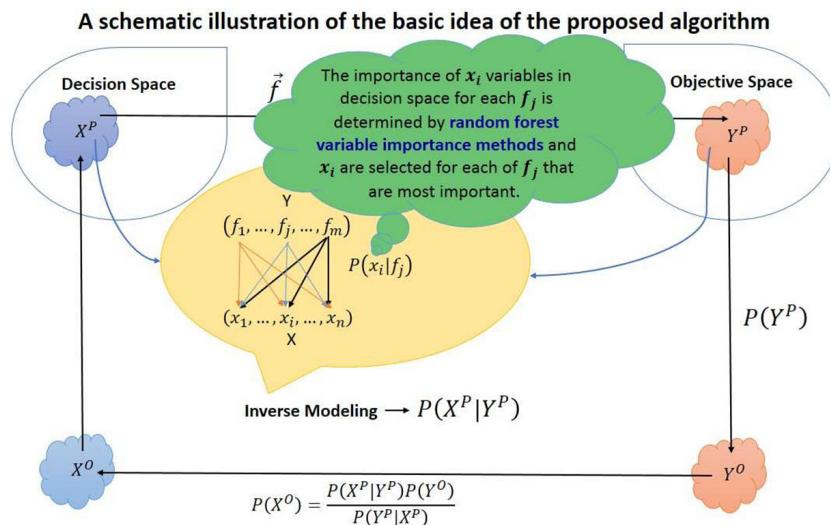


FIGURE 2 The schematic illustration of the basic idea in proposed algorithm [Color figure can be viewed at wileyonlinelibrary.com]

In addition, random grouping process cannot determine the best assignment of x_n to f_m and its performance is based on random assignment. In fact, the importance of a variable x_n in decision space for predicting f_m is determined by random forest feature importance and variables are selected that are most importance. This paper proposes a new model-based on random forest, in order to determine the best assignment of x_n to f_m . When the distribution of each gene (x_i) over any of fitness functions (f_j), are known, the best distributions can be select and it is not necessary to be assigned x_n to f_m randomly. On other hands, for decision variables assigned to the same group, their correlations can be implicitly increased by selecting the best distributions. Given m objectives, m groups of inverse models will be made for each subpopulation, where all models in group j , $1 \leq j \leq m$, apply f_j as the variable. In each group, L decision variables will be selected by suggested method to build inverse models using f_j as the variable, where $L \ll n$ is a parameter to be determined. The schematic illustration of the basic idea in the proposed algorithm is shown in Figure 2. In IM-MOEA, the Gaussian process (GP) and Bayesian linear regression is used 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 and it assumes that there are N_k individuals (data pairs) overall for training m groups of inverse models. In proposed algorithms, each groups of inverse models are trained through variable importance of random forest. The univariate probability distribution, $P_k(x_i|f_j)$, is calculated according to Equation (19), in which the mean and variance of the Gaussian distribution, are obtained according to Equations (15) and (16). The overall algorithm framework of the proposed method, based on random forest feature importance, is illustrated in Algorithm 1. In each generation, population is divided in to a number of subpopulations based on predefined reference points in the objective space. Then selection is done within each subpopulation and using NSGA-II, the individuals in each subpopulation sorts into a number of nondominated fronts and a crowding distance is applied. Then an inversed model based on random forest feature importance is made for each subpopulation for reproduction. Offspring individuals reproduced by all subpopulations are put together with the parents in the present generation to form the population for the next generation.



Algorithm 1. The overall algorithm framework of the proposed method, based on random forest

Initialization: randomly initialize population, define 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;

Nondominated Sorting and Selection: create sub parent populations $P^1(t), \dots, P^K(t)$;

for $k = 1$ to K **do**

Inverse Modeling: for each sub population $P^k(t)$, apply the **variable importance of Random Forest** method to

 determine the best assignment of x_n to f_m in each group, which inverse models are to be build; training a

 Gaussian process with Bayesian linear regression for each inverse model;

Reproduction: reproduce new condidate solutions, $O^k(t)$, for each subpopulation by sampling the objective space using

 the inverse models; perform mutation on the sampled condidate solutions;

Update the combined population: $P(t + 1) = \bigcup_{(k=1)}^K (P^k(t) \cup O^k(t))$;

end for

$t = t + 1$;

end while

Algorithm 2 uses the random forest naïve variable importance method and the individuals in each subpopulation sorts into a number of nondominated fronts and a crowding distance is applied for each subpopulation of nondominated solutions. Then the random forest naive variable importance is applied to determine the best assignment of x_n to f_m in each group and null distribution is approximated based on the observed nonpositive importance scores, while in classical random forest variable importance, some values of variable importance were negative or close to zero and do not improve the trees' predictive strength. In fact, with the random forest naive variable importance, the importance of all $x_n, n = 1, \dots, n$, is determined for each of the $f_m, m = 1, \dots, m$ in each subpopulation and then x_i are chosen for each f_m which are higher variable importance than other x_n . Then an inversed model is made based on the selected x_i for each f_m , for each subpopulation in order to reproduction.

Algorithm 3 uses the random forest cross-validation variable importance method and apply twofold cross-validation in order to assignment of the best x_n to f_m in each group of subpopulation. In this method, each subpopulation that are sorted into a number of nondominated fronts and a crowding distance, are divided into two sets with equal size randomly and each set generates a random forest. Then two forests are applied to compute the hold out variable importance. The null distribution for the hold-out variable importance is approximated based on the observed nonpositive importance scores. Therefore, the Naïve testing approach is applied in this method. In fact, in this method, the x_i are chosen for each f_m which are higher variable importance than other x_n . Then an inversed model is made based on the selected x_i for each f_m , for each subpopulation in order to reproduction. This method is appropriate for high dimensional data.

Algorithm 2. The overall algorithm framework of the proposed method, based on random forest naïve variable importance

Initialization: randomly initialize population, define K reference vectors for subpopulation creation.
 $/*\text{main loop}*/$
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;
 Nondominated Sorting and Selection: create sub parent populations $P^1(t), \dots, P^K(t)$;
 for $k = 1$ to K **do**
 Inverse Modeling: for each subpopulation $P^k(t)$, apply the **variable importance of Random Forest Naive** method
 to determine the best assignment of x_n to f_m in each group, which inverse models are to be build; training a Gaussian process with Bayesian linear regression for each inverse model;
 Reproduction: reproduce new candidate solutions $O^k(t)$ for each subpopulation by sampling the objective space using the inverse models; perform mutation on the sampled candidate solutions;
 Update the combined population: $P(t + 1) = \bigcup_{(k=1)}^K (P^k(t) \cup O^k(t))$;
 end for
 $t = t + 1$;
end while

4.2 | Classical permutation variable importance random forest

Decision trees are a public method for different machine learning tasks. But, in particular, when trees are grown very deep, tend to learn highly immethodical patterns. They over fit their training sets, that is, have low bias, but very high variance. Random forests are a way of averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. Therefore, Random Forest is an ensemble method that operate by making a collection of decision trees at training time and creating the class that is classification or mean prediction (regression) of the individual trees.⁴² This algorithm is a flexible, even without hyper parameter tuning, a great result most of the time and it can be used for both classification and regression tasks.⁴³ Leo Breiman suggested a method of making a forest of uncorrelated trees via a classification and regression tree (CART) like procedure, which combined with randomized node optimization and bagging. Also, he combined several components, which formed the basis of modern style of random forests, in particular using out-of bag error, as an estimate of generalization error and measuring variable importance via permutation. The training algorithm for random forests uses the general methods of bootstrap aggregating, also called bagging, in order to tree learners. Through bagging approach, resampling is used for the generation of each tree in the forest. A bagging subset of the training dataset is generated to train each tree in the forest. In fact, on average, each tree uses two-thirds of training dataset. The unused elements are called the out-of-bag (OOB) samples and this elements can be applied for validation. Given a training set $X = x_1, \dots, x_n$ with responses $Y = y_1, \dots, y_n$, through bagging approach,



B times selects a random sample with replacement of the training set and fits trees to these samples:

For $b = 1 \dots B$:

1. Sample, with replacement, n training examples from X, Y ; call these X_b, Y_b .
2. Train a classification or regression tree f_b on X_b, Y_b .

end For

Algorithm 3. The overall algorithm framework of the proposed method, based on random forest cross-validation variable importance

Initialization: randomly initialize population, define 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 $P^1(t), \dots, P^K(t)$;

for $k = 1$ to K **do**

Inverse Modeling: for each sub population $P^k(t)$, apply the **variable importance of Random Forest**

Cross-validation method to determine the best assignment of x_n to f_m in each group, which inverse models are to be

 build; training a Gaussian process with Bayesian linear regression for each inverse model;

Reproduction: reproduce new candidate solutions $O^k(t)$ for each subpopulation by sampling the objective space using

 the inverse models; perform mutation on the sampled candidate solutions;

Update the combined population: $P(t+1) = \bigcup_{(k=1)}^K (P^k(t) \cup O^k(t))$;

end for

$t = t + 1$;

end while

After training, predictions for samples x' which have not been seen, can be generated by averaging the predictions from all the individual regression trees on x' using the following equation:

$$\hat{f} = \frac{1}{N} \sum_{b=1}^B f_b(x'), \quad (20)$$

Or by taking the majority vote in the case of classification trees. This bagging procedure results to better model performance, because it reduces the variance of the model, without growing the bias. Additionally, an estimate of the uncertainty of prediction can be built as the SD of the predictions from all the individual regression trees on x' using the following equation:

$$\delta = \sqrt{\sum_{b=1}^B (f_b(x') - \hat{f})^2 / B - 1} \quad (21)$$

The number of samples/trees, B , is a free parameter. An optimal number of trees B can be detected through cross-validation⁴⁴ or by observing OOB error.⁴⁵ Cross-validation or out-of-sample testing is any of different similar model validation methods for evaluating how a result of a statistical analysis will generalize to an independent dataset. OOB error also named OOB estimate, is method of measuring the prediction error of random forests, that using bootstrap aggregation (bagging) to subsample data samples used for training. OOB is the mean prediction error on each training sample x_i , using only the trees that did not have x_i in their bootstrap sample. Subsampling lets one to define an OOB estimate of the prediction efficiency improvement by assessing predictions on those observations which were not applied in the making of the next base learner. The OOB samples can be applied to estimate the importance of each variable and create a rank for them. For each tree and its respective OOB samples, the accuracy of this set, is computed and then randomly a variable among samples is permuted and the accuracy of new set is recomputed. It allows you to have a relevance comparison metric as variable importance measure.

Another way to generate an importance rank is, on each tree and each node split, to compute the split improvement by the Gini index measure and use these values to compare the importance of variables.

Breiman (2001, 2002) suggested to measure the importance of a variable X_m for predicting Y with Equation (20) through adding up the weighted impurity decreases $p(t)i\Delta(s_t, t)$ for all nodes t where X_m is applied, and then is averaged over all N_T trees in the forest:

$$Imp(X_m) = \frac{1}{N_T} \sum_T \sum_{t \in T: v(S_t) = X_m} p(t)i\Delta(s_t, t), \quad (22)$$

where $p(t)$ is the relative amount of $\frac{N_t}{N}$ samples who achieved t and, $v(s_t)$ is the variable used in splits $_t$. When we apply Gini index as impurity function, this measure is known as Gini importance or Mean Decrease Gini and this equation is called the MDI.

Suppose that we have a set of categorical input variables $V = \{X_1, \dots, X_p\}$ and a categorical output Y and the joint (Shannon) entropy of a set of V , is defined by the following equation:

$$H(X) = - \sum_{x \in X} P_X(x) \log_2 P_X(x), \quad (23)$$

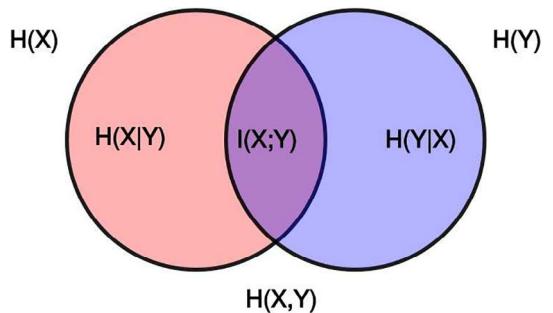
and the mean conditional entropy of a set of random variables V , given the values of variables Y , is defined by the following equation:

$$H(X|Y) = - \sum_{x \in X} \sum_{y \in Y} P_{X,Y}(x, y) \log_2 P_{X,Y}(x, y). \quad (24)$$

In probability theory and information theory, the mutual information of two random variables, is a measure to show the degree of mutual dependence between the two variables. More specifically, it quantifies the amount of information acquired about one random variable through observing the other random variable. This concept is related with the entropy of a random variable. As shown in Figure 3, additive and subtractive relationships different information measures associated with correlated variables X and Y . The space contained by both circles is the joint entropy $H(X, Y)$. The circle on the left (red and violet), is the individual entropy $H(X)$, with the red being the conditional entropy $H(X|Y)$. The circle on the right (blue and violet) is $H(Y)$, with the blue being $H(Y|X)$. The violet is mutual information $I(X; Y)$.



FIGURE 3 The relation between mutual information and entropy [Color figure can be viewed at wileyonlinelibrary.com]



The mean conditional mutual information between the set of random variables $X = \{X_1, \dots, X_p\}$ and the set of random variables $Y = \{Y_1, \dots, Y_q\}$, given the values of the third set of random variables $Z = \{Z_1, \dots, Z_r\}$, is defined by the following equation:

$$\begin{aligned} H(X; Y|Z) &= H(X|Z) - H(X|Y, Z) = H(Y|Z) - H(Y|X, Z) \\ &= - \sum_{x \in X} \sum_{y \in Y} \sum_{z \in Z} P_{X,Y,Z}(x, y, z) \log_2 \frac{P_{X|Z}(x|z)P_{Y|Z}(y|z)}{P_{X,Y|Z}(x, y|z)}. \end{aligned} \quad (25)$$

And also remind the chaining rule the following equation:

$$I(X, Z; Y|W) = I(X; Y|W) + I(Z; Y|W) + I(Z; Y|W, X). \quad (26)$$

And symmetry of the conditional mutual information between sets of random variables are as following equation:

$$I(X; Y|Z) = I(Y; X|Z). \quad (27)$$

Given a training sample Γ of N joint observations of X_1, \dots, X_p, Y independently drawn from the joint distribution $P(X_1, \dots, X_p, Y)$. A randomized and fully developed tree is a decision tree in which each node t is partitioned using a variable X_j chosen uniformly at random among those not yet applied at the parent nodes of t , where each node t is split in to $|X_j|$ subtrees, that is, one for each possible value of X_j , and where the recursive construction process stops only when all P variables have been applied along the current branch. Each node has a left and right child and impurity measure of each node is equal the following equation:

$$i(t) = P_L \cdot i(N_L) + (1 - P_L) \cdot i(N_R), \quad (28)$$

where N_L and N_R are impurity measure of left and right node and P_L is the proportion of samples which are assigned in to left node and $(1 - P_L)$ is the proportion of samples which are assigned in to right node. Using the entropy $H(Y|t) = - \sum_j P_{j|t} \log_2 P(j|t)$ as impurity measure $i(t)$, the importance of a variable X_j , can be rewritten in terms of mutual information using the following equation:

$$Imp(X_j) = \frac{1}{M} \sum_{m=1}^M \sum_{t \in \varphi_m} P(t) I(Y; X_j|t). \quad (29)$$

As the size of training sample grows to infinity, $P(t)$ turns in to the probability that an object attains node t , that is, a probability $P(B(t) = b(t))$, where $B(t) = (X_{(i_1)}, \dots, X_{(i_k)})$ is a subset of k variables tested in the branch from the root node to the parent of t and $b(t)$ is the vector of values of these variables. As, the number of M randomize trees grows to infinity, the importance of a variable X_j can be written using the following equation:

$$Imp(X_j) = \sum_{B \subseteq V^{-j}} \sum_{B \in X_{i_1}, \dots, X_{i_k}} \alpha(B, b, X_j, p) P(B = b) I(Y; X_j | B = b), \quad (30)$$

where b is a set of values for the variables in B , and $\alpha(B, b, X_j, p)$ is the probability that a node t (at depth k) in a randomize tree tests the variables X_j , which is equal to a branch with a path defined, that leading to t , through all k variables $X_{(i_1)}, \dots, X_{(i_k)} \in B$ and their corresponding values in b and the probability of this branch is equal to the probability of choosing (uniformly at random) $X_{(i_1)}$ at the root node as remaining $X_{(i_2)}, \dots, X_{(i_k)}$ variables in the subtree corresponding to the value $X_{(i_1)}$ of $X_{(i_1)}$ defined in b . It is certain that the root node is split in to $|X_{(i_1)}|$ subtrees. Then the probability of testing X_j at the end of this branch is equal to the probability of choosing X_j among the remaining $p - k$ variables. By recursion, we have the following equation:

$$\frac{1}{p} \frac{1}{p-1} \cdots \frac{1}{p-k+1} \frac{1}{p-k} = \frac{(p-k)!}{p!} \frac{1}{p-k}. \quad (31)$$

Since the order along which the variables appear in the branch is of no importance, $\alpha(B, b, X_j, p)$ in Equation (29), actually includes all $k!$ ways of making a branch composed of the variables and values in B and b :

$$\alpha(B, b, X_j, p) P(B = b) I(Y; X_j | B = b) = k! \frac{(p-k)!}{p!} \frac{1}{p-k} = \frac{1}{C_p^k} \frac{1}{p-k}. \quad (32)$$

From this expression, $\alpha(B, b, X_j, p)$ depends only on the size k of B and the number p of variables. So, by grouping in the previous equation of $Imp(X_j)$ in Equation (30) conditioning variable subsets B according to their sizes and using the definition of conditional mutual information, α can be factored out:

$$Imp(X_j) = \sum_{k=0}^{p-1} \frac{1}{C_p^k} \frac{1}{p-k} \sum_{B \in P_k(V^{-j})} I(X_j; Y | B), \quad (33)$$

where $V^{(-j)}$ denotes the subset $V \setminus \{X_j\}$, and $P_k(V^{(-j)})$ is the set of subsets of $V^{(-j)}$ cardinality k and $I(X_j; Y | B)$ is the conditional mutual information of X_j and Y given the variables in B , which can be calculated by Equation (22).

A variable (eg, $Imp(X_j)$) is called related if after the random permutation, the errors of tree prediction or accuracy of prediction increase significantly. Regarding the values variable importance, the negative values or values of zero, illustrate that variable does not improve the trees' predictive strength, because the error rates are similar or even larger when applying unchanged version of the variable. In contrast, a positive value for the variable importance expresses that the variable at least some improves the trees' predictive strength, because the error rates are smaller when applying the original version of the variable for deriving tree predictions. However it cannot be deduced that a positive value for the variable importance demonstrates a relative value. For



this purpose a test procedure is required. The testing approach of Altmann et al was proposed as a heuristic in order to modify biased importance measures, such as the Gini importance measure. Also, the P -values are computed from importance scores. With this feature, the user can select related variables. In the first step of this process, the variable importance scores are computed for all variables. In the second step, for variables that are not associated with the response, the importance scores are obtained by randomly permuting the response variable in order to split any continuity between the response variable and all predictor variables. Then a new random forest is created and the importance scores for the predictor variables is computed. This process is repeated s times. The importance scores can be considered as the unknown null distribution. The S importance scores are applied for computing the P -value for variables. For this purpose, the fraction of S importance scores that are greater than the original importance score, is computed. Also, it can be assumed a parametric distribution such as the Gaussian, Log-normal, or Gamma distribution for the importance scores of irrelevant predictor variables. The parameters of each distribution are computed by the maximum likelihood estimate. This approach is called parametric approach.

4.3 | Random forest naïve variable importance

In this method, a new heuristic approach is proposed in which the null distribution is approximated based on the observed non-positive importance scores. For this purpose, the variable importance null distribution reflects the observed negative and zero importance scores on the y-axis. Let $M_1 = \{VI_j | VI_j < 0; j = 1 \dots p\}$ defines the observed negative variable importance scores and $M_2 = \{VI_j | VI_j = 0; j = 1 \dots p\}$ defines the observed variable importance scores that are equal zeros and P defines the number of candidate predictors. In this method, a new importance score is defined which multiplying the negative importance scores by $-1 : M_3 = \{-VI_j | VI_j < 0; j = 1 \dots p\} = -M_1$. Therefore, the null distribution \hat{F}_0 is obtained as the empirical cumulative distribution function of $M = M_1 \cup M_2 \cup M_3$. This method is not appropriate for all type of data and data should include a large number of variables without effect.

4.4 | Random forest cross-validation variable importance

This method does not apply the OOB observation and uses the cross-validation procedure. The data is split into k sets that have equal size. Then k forests is constructed where the l th forest is created based on observations that are not part of the l th set. The simplest type of cross-validation results for $k = 2$, so that each of the two sets is once applied for generating the forest and once for deriving importance scores. This method is known as twofold cross-validation or hold out method. Therefore, the data is divided into two sets with equal size randomly and each set generates a random forest. Then two forests are applied to compute the hold out variable importance. The null distribution for the hold-out variable importance is approximated based on the observed non-positive importance scores. Also, the Naïve testing approach is applied in this method. For this purpose the sets is defined: $M_1 = \{VI_j^{HO} | VI_j^{HO} < 0; j = 1 \dots p\}$ for all negative importance scores, $M_2 = \{VI_j^{HO} | VI_j^{HO} = 0; j = 1 \dots p\}$ for all importance scores of zeros and $M_3 = \{-VI_j^{HO} | VI_j^{HO} < 0; j = 1 \dots p\} = -M_1$ for all negative importance scores multiplied by -1 and consider the empirical cumulative distribution function \hat{F}_0 of $M = M_1 \cup M_2 \cup M_3$. This method uses the difference in error rates before and after randomly permuting the values of the considered variable and called random forest cross-validation.

5 | EXPERIMENTAL ANALYSIS

In this section, some tests have been conducted in order to show the performance of the three proposed methods at a wide range of parameters and proposed methods are tested and compared with IM-MOEA, CSEA, and RM-MEDA on the two groups of benchmark tests: the modified test instances of ZDT and DTLZ, and The WFG test suite.

5.1 | Test sets

The three proposed methods is tested on two sets of MOPs, the modified test instances of ZDT and DTLZ, which introduced in Reference 2 (F_1 to F_{10}) which is used for results of IM-MOEA¹¹ and the linkage between the decision variables as the following equations:

$$x_i \rightarrow \left(1 + \alpha \frac{i}{n}\right) x_i, \quad (34)$$

$$x_i^2 \rightarrow \left(x_i^{\frac{1}{1+\beta \frac{i}{n}}}\right), \quad (35)$$

where i is the index of each decision variable, and α and β are two control parameters and are equal to $\alpha = 5$ and $\beta = 3$, and the number of decision variables is put to $n = 30$. In this test problem F1 has a linear PF and test problem F3 has many local PFs and one global PF in order to check an MOEA's ability to converge to the global PF. The test problem F4 checks an MOEA's ability in order to maintain a good distribution of solutions. Both test problems F5 and F6 have a curve PF. The test problem F7 has a disconnected set of Pareto-optimal regions in order to maintain subpopulation in different Pareto-optimal regions.

The WFG test suite, containing nine test problems (WFG1 to WFG9).⁴⁶ We use three-objective WFG test instances whose pareto set has a dimentionality of four. The number of decision variables of WFG functions is put to 104. Both test problems of WFG1 and WFG7 are separable and unimodal. The nonseparable reduction of WFG6 and WFG9 is more difficult than that of WFG2 and WFG3, the multimodality of WFG4 has larger “hill sizes” and is thus more difficult than that of WFG9.

5.2 | Performance metrics

A simple way to evaluate the quality of solution sets is Quality indicators (QIs). In general, QIs can be classified in to six categories.⁴⁷ (i) QIs for convergence, (ii) QIs for spread, (iii) QIs for uniformity, (iv) QIs for cardinality, (v) QIs for both spread and uniformity, and (vi) QIs for incorporated quality of the four quality aspects. There are two classes of convergence in QIs: one is to evaluate the Pareto dominance relation between solutions or sets and two is to evaluate the distance of a solution set to the PF. The GD indicator measures the quadratic mean of the Euclidean distances of solutions set to the nearest point on the PF.⁴⁸ Spread quality is related to the area of a solution set covering. The pure diversity (PD) summarizes the dissimilarity of each solution to the remaining solutions of a solution set.⁴⁹ Quality indicators for uniformity evaluate how uniformly a set's solutions are distributed and measuring the variation of the distance between the



solutions, such as spacing (SP).⁵⁰ QIs for cardinality add a different nondominated solution to the set under regard should improve the evaluation result. The QIs for spread and uniformity are closely related and can be used together to show the diversity of solution sets and classified into two groups: distance-based indicators and region division-based indicators. The region division-based QIs partitions a special space into many equal size cells and then computes the number of cells having solutions of the set. Some of them considers grid-like cells which partition the space into many hyperboxes, such as diversity metric (DM).⁵¹ QIs for all quality aspects cover convergence, spread, uniformity, and cardinality and they can be classified into two groups: distance-based QIs that measures the distance of PF to the solution set under consideration, such as inverted generational distance (IGD)^{2,52} and volume-based QIs that measures the size of volume and is specified by the consideration solution set in conjunction with some specifications, such as hypervolume (HV).⁵³

5.3 | Parameter studies

In this work, QIs of GD, PD, DM, HV, and IGD are applied to measure the performance on test instances F1 to F10 (most of whih are biobjective MOPs) and 500 uniformly distributed points are selected from the PF of each test instance to be P^* . Also, IGD and HV is applied to measure the performance on the three objective WFG test instances, and the reference point $y^* = (2.5, 4.5, 6.5)$ is used to for all instances.

5.4 | Parameter setting

For all the competing algorithms, the same parameter setting of problems is applied. Parameter adjustments for these compared algorithm as shown in Table 1. The population size is adjusted to 100 for proposed algorithm and IM-MEDA, RM-MEDA, and CSEA. We carried out 20 independent runs for each compared algorithm on each test instance. The final condition for each algorithm is adjusted to a maximum of 100 000 fitness evaluations for all the test instances.

5.5 | Evaluation and comparison of algorithms

In this section, the performance of three proposed methods are compared with the IM-MOEA,¹¹ that is to construct Gaussian process-based inverse model using the random grouping technique that map all found nondominated solutions from the objective space to the decision space, RM-MEDA, plan a reproduction operator by explicitly modeling the regularity in the distribution of Pareto optimal solutions and CSEA, use a artificial neural network in order to predict the dominance relationship between candidate solutions and reference solutions instead of approximating the objective values separately.

TABLE 1 Parameter settings of the four algorithms in comparison

Algorithm	IM-MOEA-RF-CV	IM-MOEA-RF-Naive	IM-MOEA-RF	IM-MOEA	RM-MEDA	CSEA
Parameter settings	$K = 10$	$K = 10$	$K = 10$	$K = 10$	$k = 5$	$k = 5$

The Wilcoxon rank sum test is adapted to compare the results obtained by our proposed algorithm and the other four algorithms at an importance level of 0.05. In the tables that summarize the statistical results, the first and second lines illustrate the mean values and the SDs, respectively. As a result of Wilcoxon rank sum test, a + labeled means that the compared algorithm is better than the proposed algorithm; by contrast, a - labeled means that the proposed algorithm is better than the compared algorithm; while a ≈ labeled means that there is no statistically significant difference between the result obtained by the proposed algorithm and the compared algorithm.

5.6 | Results

5.6.1 | The GD values

The GD indicator in Table 2 compares the statistical results of QIs for investigating of convergence on WFG1 to WFG9 that acquired by the six algorithms.

It can be illustrated that the convergence of the proposed algorithm (eg, IMMOEA-RF-CV, IMMOEA-RF-Naive, and IMMOEA-RF) on functions of WFG is better than CSEA, IM-MOEA, and RMMEDA. The IMMOEA-RF-CV is better than IMMOEA and RMMEDA in four cases and six cases, respectively. Also, IMMOEA-RF is better than both IM-MOEA and RM-MEDA in eight cases. It is observed that the proposed algorithm is better than the CSEA. IMMOEA-RF-Naive has the best performance compared to all of the algorithms compared, as well as among the proposed algorithms. Therefore, the convergence rate of proposed is better than the other algorithms compared and the quadratic mean of the Euclidean distances of solutions set in proposed algorithms to the nearest point on the PF is better result than IM-MOEA in inversed modeling and CSEA in surrogate assisted evolutionary algorithms (SAEAs).

5.6.2 | The PD values

The PD indicator in Table 3 compares the statistical results of QIs for investigating of spread on function of IM-MOEA, that acquired by the six algorithms.

It can be illustrated that the spread of the proposed algorithm (eg, IMMOEA-RF-CV, IMMOEA-RF-Naive, and IMMOEA-RF) on IM-MOEA functions is better than IM-MOEA and RMMEDA, and CSEA. The IM-MOEA-RF has only a worse performance and in four cases than the IM-MOEA and the RMMEDA, respectively. The IMMOEA-RF-CV is better than the IMMOEA-RF-Naive in two cases and are equal in five cases and it is better than the IMMOEA-RF in only four cases and in fact the IMMOEA-RF is overall better than the IMMOEA-RF-CV. The comparison of two algorithms IMMOEA-RF-Naive and IMMOEA-RF shows that the IMMOEA-RF-Naive is better than the IMMOEA-RF in three cases and are equal in four cases. Also CSEA has weakest results among the algorithms compared. In general, the comparison of three algorithms (eg, IMMOEA-RF-CV, IMMOEA-RF-Naive, and IMMOEA-RF) shows that the IMMOEA-RF-Naive performs better than three other proposed methods in spread and has the least amount of the dissimilarity of each solution to the remaining solutions of a solution set.

TABLE 2 Statistical results of GD values obtained by each algorithm in comparison on 104-D WFG1 to WFG9

Problem	M	D	FES	IMMOEA-RF-CV	IMMOEA-RF-Naive	IMMOEA-RF	IMMOEA	RMMEDA	CSEA
WFG1	3	104	10 000	2.7300e-1(1.63e-2) –	2.7341e-1(5.02e-1) –	2.6599e-1(6.55e-3) –	2.6407e-1(2.20e-2) –	1.6456e-1(6.47e-3) +	2.2509e-1(1.95e-1)
WFG2	3	104	10 000	5.4833e-2(4.17e-3) +	3.3481e-2(4.58e-3) +	3.4900e-2(4.26e-3) +	5.6924e-2(4.48e-3) +	8.0176e-2(4.02e-3) +	1.8828e-1(1.64e-1)
WFG3	3	104	10 000	1.1115e-1(8.13e-3) –	9.3605e-2(5.61e-3) +	9.5773e-2(3.71e-3) +	1.0782e-1(5.07e-3) +	1.1860e-1(2.01e-3) –	3.6296e-1(3.59e-2)
WFG4	3	104	10 000	3.9297e-2(4.77e-3) +	2.9336e-2(4.57e-3) +	3.2365e-2(5.18e-3) +	2.9572e-2(1.50e-3) +	5.3660e-2(1.19e-3) +	2.7579e-1(2.27e-1)
WFG5	3	104	10 000	7.2528e-2(3.04e-3) –	2.2682e-2(1.44e-2) +	2.1048e-2(8.60e-3) +	5.9938e-2(1.35e-3) +	6.1782e-2(2.56e-3) +	3.2274e-1(3.03e-1)
WFG6	3	104	10 000	4.7563e-2(9.28e-3) +	3.3294e-2(7.94e-3) +	3.1868e-2(9.23e-3) +	7.0041e-2(3.72e-3) +	1.0280e-1(3.55e-3) –	3.4513e-1(1.52e-1)
WFG7	3	104	10 000	4.2229e-2(3.33e-3) +	3.1815e-2(3.43e-3) +	3.2477e-2(3.31e-3) +	5.4099e-2(2.54e-3) +	7.3161e-2(1.10e-3) +	5.3793e-2(4.68e-2)
WFG8	3	104	10 000	5.2962e-2(5.81e-3) +	3.8531e-2(2.47e-3) +	4.0641e-2(3.10e-3) +	4.6490e-2(1.36e-3) +	7.6794e-2(4.51e-3) +	5.4923e-1(8.29e-2)
WFG9	3	104	10 000	1.8221e-2(3.98e-3) +	1.4617e-2(4.58e-3) +	1.2824e-2(2.23e-3) +	5.8736e-2(5.48e-3) +	1.3183e-1(7.34e-3) +	9.8600e-1(6.07e-3)
+/-=				7/2/0	8/1/0	8/1/0	8/1/0	3/0/7	

TABLE 3 Statistical results of pure diversity values obtained by each algorithm in comparison on 30-D F1 to F10

Problem	M	D	FEs	IMMOEA-RF-CV	IMMOEA-RF-Naive	IMMOEA-RF	IMMOEA	RMMEDA	CSEA
F1	2	30	10000	1.9974e+3 (4.37e+2)=	2.4158e+3 (1.64e+3)=	1.9537 e+3 (9.13e+2)=	1.3857e+3 (2.42e+2)=	1.6943e+3 (2.27e+2)+	7.1470e+2 (1.25e+2)
F2	2	30	10000	2.0330e+3 (6.89e+2)=	2.2866e+3 (7.74e+2)+	1.6429 e+3 (3.48e+2)+	6.2366e+2 (3.05e+2)-	1.5594e+3 (2.09e+2)+	1.7887e+2 (1.04e+2)
F3	2	30	10000	1.4538e+3 (2.02e+2)-	1.5207e+3 (5.61e+2)-	1.6168e+3 (2.79e+2)-	1.2664 e+3 (2.32e+2)-	3.0409e+3 (1.06e+3)+	1.1716e+3 (2.08e+2)
F4	3	30	10000	4.6547e+5 (1.75e+5)+	6.1658e+5 (2.22e+5)+	5.8706e+5 (2.17e+5)+	6.2830e+5 (1.00e+5)+	1.9303e+5 (4.76e+4)+	1.2773e+5 (4.63e+4)
F5	2	30	10000	1.9495e+3 (3.05e+2)=	1.8068e+3 (2.78e+2)=	1.8263e+3 (3.09e+2)=	1.6102e+3 (1.54e+2)=	1.8233e+3 (3.42e+2)+	1.3986e+3 (3.09e+2)
F6	2	30	10000	2.0506e+3 (4.02e+2)=	1.88334e+3 (4.46e+2)=	1.9350e+3 (2.38e+2)=	1.6515e+3 (2.36e+2)	2.0551e+3 (2.85e+2)+	1.1496e+3 (2.57e+2)
F7	2	30	10000	1.3649e+3 (2.85e+2)-	1.3513e+3 (1.58e+2)-	1.5024e+3 (3.51e+2)=	1.6352e+3 (3.34e+2)=	1.9149e+3 (4.84e+2)+	1.3022e+3 (1.78e+2)
F8	3	30	10000	3.5527e+5 (5.37e+4)+	4.9253e+5 (7.73e+5) +	4.2338e+5 (6.32e+4)+	3.6660e+5 (3.81e+4)+	2.0920e+5 (1.90e+5)-	2.4261e+5 (5.25e+4)
F9	2	30	10000	1.5853e+3 (1.66e+2)=	1.7751e+3 (2.42e+2)=	1.7457e+3 (2.71e+2)=	1.9060e+3 (2.82e+2)+	1.8095e+3 (3.00e+2)+	1.6640e+3 (3.51e+2)
F10	2	30	10000	1.5379e+3 (1.76e+3)-	3.2641e+3 (3.87e+3)-	9.5037e+3 (4.61e+3)=	2.5249e+3 (1.87e+3)-	7.2951e+3 (1.41e+3)-	8.0445e+3 (4.99e+3)
+/-=				2/3/5	3/3/4	2/1/7	2/5/3	8/2/0	



5.6.3 | The DM values

The DM indicator in Table 4 compares the statistical results of QIs for investigating of spread and uniformity on WFG1 to WFG9, that acquired by the six algorithms.

It can be illustrated that the spread and uniformity of the proposed algorithm (eg, IMMOEA-RF) on functions of WFG is better than the IMMOEA and RMMEDA in three and in four cases are equal. Also, the IMMOEA-RF is better than the IMMOEA-RF-CV and the IMMOEA-RF-Naive. Given that the spread rate of proposed methods is better, it can deduce that uniformity of IMMOEA-RF-CV and IMMOEA-RF-Naive are weaker than IMMOEA-RF, IMMOEA, RMMEDA, and CSEA, but uniformity of IMMOEA-RF is best in comparison of all compared algorithms.

5.6.4 | The HV values

The HV indicator in Table 5 compares the statistical results of QIs for investigating of all quality aspects (eg, convergence, spread, uniformity, and cardinality) on functions of IMMOEA, that acquired by the six algorithms. Also, Figure 4 shows the resulting HV curves of F1 to F9, in which the horizontal and vertical axes are test instances and HV values, respectively.

It can be illustrated that all quality aspects (eg, convergence, spread, uniformity, and cardinality) of the proposed algorithm (eg, IMMOEA-RF-CV, IMMOEA-RF-Naive, and IMMOEA-RF) on functions of IM-MOEA is better than the IM-MOEA, REMMEDA, and CSEA. So altogether, the proposed algorithm is better than IM-MOEA and in general. Also, the comparison of these three proposed methods illustrates that they have almost the same performance. In F6 and F8, which the RMMEDA has better performance, both function are nonlinear variable linkage and also F6 is nonuniformly distributed. The CSEA has weakest results among the algorithms compared.

Table 6 shows the statistical results of WFG1 to WFG9 that acquired by the six algorithms in terms of HV values on 104-D. Also, Figure 5 shows the resulting HV curves of WFG1 to WFG9, in which the horizontal and vertical axes are test instances and HV values, respectively.

It can be illustrated that the three proposed methods have the better overall performance and the best results are acquired by IMMOEA-RF-Naive on WFG2, WFG3, and WFG7. The IMMOEA-RF-CV and the IMMOEA have the same performance, but the IMMOEA-RF-CV has a better performance than the RMMEDA. Also the performance of CSEA is better than IM-MOEA and REMMEDA. The nondominated solutions resulting in the best HV among the 20 runs acquired by each algorithm on WFG7 are displayed in Figure 6.

On Figure 6, it is observed that solutions acquired by three proposed algorithms on WFG7 have better convergence and distribution than other compare algorithms, confirming the better performance illustrated by HV.

5.6.5 | The IGD values

Table 7 shows the statistical results of F1 to F10 that acquired by the six algorithms in terms of IGD values. Also, Figure 7 shows the resulting IGD curves of F1 to F9, in which the horizontal and vertical axes are test instances and IGD values, respectively.

TABLE 4 Statistical results of diversity metric values obtained by each algorithm in comparison on 104-D WFG1 to WFG9

Problem	M	D	FES	IMMOEA-RF-CV	IMMOEA-RF-Naive	IMMOEA-RF	IMMOEA	RMMEDA	CSEA
WFG1	3	104	10 000	8.5519e-2 (1.30e-2) +	9.7665e-2 (2.27e-2) +	1.0639e-1 (2.85e-2) +	9.0680e-2 (2.20e-2) -	4.4450e-2 (4.71e-3) -	6.7125e-2 (5.24e-2)
WFG2	3	104	10 000	5.7863e-1 (3.95e-2) =	5.5818e-1 (4.09e-2) =	5.6997e-1 (2.65e-2) =	6.2182e-1 (2.35e-2) +	5.7946e-1 (2.70e-2) -	6.5104e-1 (6.00e-2)
WFG3	3	104	10 000	4.8777e-1 (1.68e-2) +	5.1681e-1 (2.71e-2) +	5.2503e-1 (4.45e-2) +	4.9991e-1 (1.71e-2) +	4.6943e-1 (1.23e-2) +	1.9689e-1 (1.48e-1)
WFG4	3	104	10 000	6.0896e-1 (1.31e-2) -	6.0675e-1 (2.29e-2) -	6.0279e-1 (2.25e-2) -	6.4873e-1 (2.24e-2) -	6.6996e-1 (2.39e-2) -	7.8024e-1 (5.13e-2)
WFG5	3	104	10 000	5.6570e-1 (3.12e-2) =	6.0446e-1 (3.94e-2) =	6.1814e-1 (1.91e-2) +	5.9243e-1 (2.37e-2) =	5.8877e-1 (1.85e-2) =	5.8823e-1 (7.52e-2)
WFG6	3	104	10 000	5.8189e-1 (1.66e-2) -	5.6329e-1 (2.76e-2) -	5.8116e-1 (2.25e-2) -	5.7684e-1 (2.44e-2) -	6.0032e-1 (1.20e-2) +	4.6719e-1 (4.82e+2)
WFG7	3	104	10 000	5.3749e-1 (2.10e-2) -	5.6315e-1 (2.25e-2) -	5.6291e-1 (3.07e-2) -	5.5655e-1 (2.48e-2) -	1.4147e-2 (9.19e-2) -	6.4453e-1 (1.58e-2)
WFG8	3	104	10 000	5.6462e-1 (2.41e-2) -	5.7167e-1 (2.57e-2) -	5.6807e-1 (2.95e-2) -	6.2178e-1 (1.55e-2) =	6.1369e-1 (2.80e-2) +	3.9036e-1 (7.66e-2)
WFG9	3	104	10 000	5.9033e-1 (3.25e-2) =	6.1587e-1 (2.85e-2) =	6.1538e-1 (1.89e-2) -	5.5291e-1 (1.67e-2) -	5.9455e-1 (3.62e-2) +	4.1449e-1 (4.36e-2)
+/-=				2/4/3	2/4/3	3/4/2	4/3/2	4/4/1	

TABLE 5 Statistical results of hypervolume values obtained by each algorithm in comparison on 30-D F1 to F9

Problem	M	D	FES	IMMOEA-RF-CV	IMMOEA-RF-Naive	IMMOEA-RF	IMMOEA	RMMEDA	CSEA
F1	2	30	10000	8.3834e-1 (4.14e-3) =	8.3314e-1 (5.61e-1) =	8.3397e-1 (4.41e-3) =	8.1589e-1 (1.42e-2) -	8.2764e-1 (2.39e-2) +	5.9867e-1 (1.23e-1)
F2	2	30	10000	4.9263e-1 (8.27e-3) =	4.8350e-1 (8.82e-3) =	4.8838e-1 (7.93e-3) =	3.3266e-1 (1.02e-1) -	4.8803e-1 (2.50e-2) +	3.3296e-1 (7.14e-2)
F3	2	30	10000	4.2018e-1 (7.35e-4) +	4.2114e-1 (7.23e-4) +	4.2347e-1 (3.97e-4) +	3.8141e-1 (1.64e-2) +	3.0410e-3 (8.04e-3) -	9.2116e-2 (1.33e-2)
F4	3	30	10000	2.4817e-1 (6.45e-2) =	2.5210e-1 (6.26e-2) =	2.5517e-1 (6.32e-2) =	2.3191e-2 (2.91e-2) -	2.4204e-1 (2.30e-1) +	7.7279e-2 (1.38e-1)
F5	2	30	10000	8.1850e-1 (6.28e-3) +	8.2618e-1 (5.30e-3) +	8.2455e-1 (3.88e-3) +	8.1239e-1 (6.22e-3) =	8.0309e-1 (2.77e-2) +	7.7075e-1 (5.03e-2)
F6	2	30	10000	4.5888e-1 (8.58e-3) -	4.7181e-1 (6.93e-3) -	4.6782e-1 (6.85e-3) -	4.6974e-1 (8.39e-3) -	5.0011e-1 (1.84e-2) +	4.5594e-1 (3.53e-2)
F7	2	30	10000	2.7138e-1 (4.18e-2) +	3.1607e-1 (2.96e-2) +	3.1456e-1 (2.08e-2) +	2.4068e-1 (1.06e-2) +	1.7296e-1 (1.08e-2) +	1.1835e-1 (9.49e-3)
F8	3	30	10000	1.4988e-1 (4.86e-2) =	1.5444e-1 (3.22e-2) =	1.6932e-1 (4.60e-2) =	2.4282e-1 (2.58e-2) =	2.4338e-1 (1.88e-1) +	1.6238e-1 (1.46e-1)
F9	2	30	10000	8.5063e-1 (4.18e-3) +	8.5204e-1 (3.28e-3) +	8.5165e-1 (2.81e-3) +	7.3889e-1 (2.97e-2) +	4.2019e-1 (5.73e-2) -	5.9998e-1 (6.57e-2)
+/-=				4/1/4	4/1/4	3/1/5	2/5/2	7/2/0	

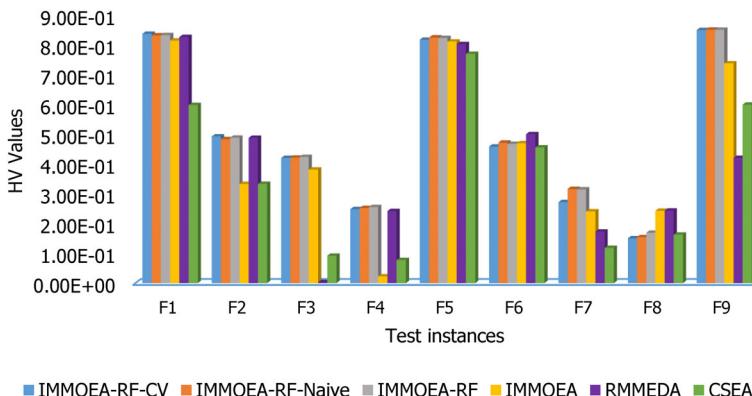


FIGURE 4 Statistical results of hypervolume values obtained by each algorithm in comparison on 30-D F1 to F9 [Color figure can be viewed at wileyonlinelibrary.com]

It can be illustrated that the three proposed methods have the better overall performance and the best result are acquired by IMMOEA-RF-Naive and IMMOEA-RF on F3, F4, F7, and F10 and F5, F8, and F9, respectively, which have strong linear and nonlinear correlations between the decision variables and highly multimodal test instances. The IMMOEA-RF-Naive and the IMMOEA-RF have a performance than the IMMOEA-RF-CV, but the IMMOEA-RF-CV has a better performance than the IMMOEA and the RMMEDA. The CSEA has weakest results among the algorithms compared.

The IGD indicator in Table 8 compares the statistical results of QIs for investigating of all quality aspects (eg, convergence, spread, uniformity, and cardinality) on WFG1 to WFG9, that acquired by the five algorithms. Also, Figure 8 shows the resulting IGD curves of WFG1 to WFG9, in which the horizontal and vertical axes are test instances and IGD values, respectively. It can be illustrated that all quality aspects (eg, convergence, spread, uniformity, and cardinality) of three proposed algorithms (eg, IMMOEA-RF-CV, IMMOEA-RF-Naive, and IMMOEA-RF) on functions of WFG1 to WFG9 is better than the IMMOEA. In WFG1, the IMMOEA-RF-Naive and IMMOEA-RF are better than IMMOEA-RF-CV and RMMEDA has the worst performance. In WFG2 and WFG3 the performance of IMMOEA-RF-CV, IMMOEA-RF-Naive and IMMOEA-RF is better than IMMOEA and RMMEDA. In WFG4 the IMMOEA has the best performance and the RMMEDA has the worst performance. Also, the performance of three proposed algorithms is approximately equal to the IMMOEA. In WFG5 and WFG6 the IMMOEA-RF has the best performance. IN WFG7 and WFG8, the IMMOEA-RF-Naive has the best performance. In WFG9, IMMOEA-RF has the best performance and the performance of IMMOEA-RF-CV and IMMOEA-RF-Naive is approximately equal to the IMMOEA-RF. Also the performance of CSEA is better than IMMOEA and REMMEDA.

5.7 | Review of compared benchmarks

Table 9 shows the relation ship of categories and benchmarks. In order to cover all categories, the behavior of HV and DM performance metrics is selected and evaluated on our test sets.

In order to evaluate the HV and DM for the six algorithms, the HV values of F2, F3, and WFG4 are shown in Figures 9, 10, and 11, respectively, and the DM values of F2, F3, and WFG4 are plotted in Figures 12, 13, and 14, respectively. By looking at the diagrams in Figures 4 and 5,

TABLE 6 Statistical results of hypervolume values obtained by each algorithm in comparison on 104-D WFG1 to WFG9

Problem	M	D	FEs	IMMOEA-RF-CV	IMMOEA-RF-Naive	IMMOEA-RF	IMMOEA	RMMEDA	CSEA
WFG1	3	104	10 000	1.1678e+0 (1.45e+0) =	2.1246e+0 (1.89e+0) +	2.8851e+0 (2.02e+0) +	2.4765e+0 (1.94e+0) +	4.6630e-1 (3.99e-1) -	1.9821e+1 (3.52e-1)
WFG2	3	104	10 000	4.5613e+1 (8.54e-1) +	5.0057e+1 (5.99e-1) +	4.9710e+1 (8.53e-1) +	4.2465e+1 (7.09e-1) +	3.6501e+1 (7.20e-1) -	4.8179e+1 (5.47e-1)
WFG3	3	104	10 000	4.1002e+0 (1.42e-1) +	4.6023e+0 (2.52e-1) +	4.5926e+0 (3.18e-1) +	2.8966e+0 (1.34e-1) +	1.6105e+0 (1.15e-1) -	3.4213e+0 (1.54e-1)
WFG4	3	104	10 000	2.2935e+1 (1.12e+0) +	2.5111e+1 (1.03e+0) +	2.4454e+1 (1.28e+0) +	2.4698e+1 (5.32e-1) +	1.8500e+1 (2.50e-1) -	2.6549e+1 (4.10e-1)
WFG5	3	104	10 000	1.4431e+1 (1.05e+0) -	2.6682e+1 (3.59e+0) +	2.6820e+1 (2.26e+0) +	1.5992e+1 (4.42e-1) -	1.7445e+1 (7.32e-1) -	2.3165e+1 (4.96e-1)
WFG6	3	104	10 000	2.0259e+1 (2.30e+0) +	2.4630e+1 (2.27e+0) +	2.5242e+1 (2.42e+0) +	1.4179e+1 (5.01e-1) +	1.0281e+1 (1.13e+0) -	2.1194e+1 (5.88e-1)
WFG7	3	104	10 000	2.0236e+1 (6.31e-1) +	2.3507e+1 (8.41e-1) +	2.3279e+1 (9.55e-1) +	1.7775e+1 (6.72e-1) +	1.4567e+1 (4.20e-1) -	1.9611e+1 (6.15e-1)
WFG8	3	104	10 000	1.8444e+1 (7.78e-1) -	2.1316e+1 (6.58e-1) +	2.1029e+1 (5.63e-1) +	1.9473e+1 (3.31e-1) +	1.4022e+1 (1.21e+0) -	2.3311e+1 (3.88e-1)
WFG9	3	104	10 000	2.7635e+1 (1.11e+0) +	2.9058e+1 (1.59e+0) +	2.9519e+1 (7.61e-1) +	1.5686e+1 (1.45e+0) +	1.1785e+1 (8.39e-1) =	1.8897e+1 (1.42e+0)
+/-=				6/2/1	9/0/0	9/0/0	8/1/0	0/8/1	

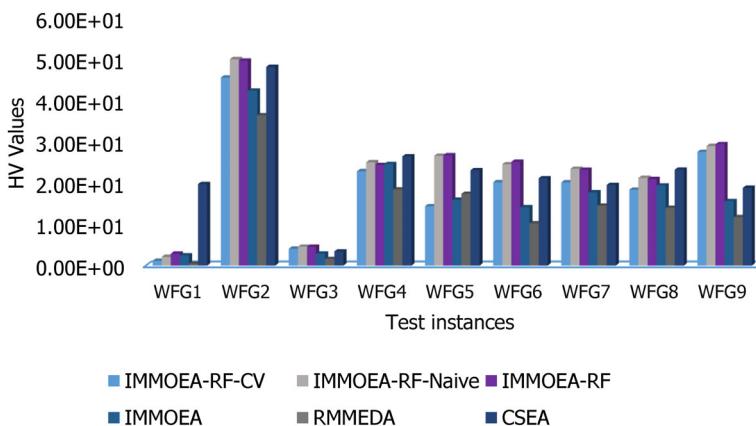


FIGURE 5 Statistical results of hypervolume values obtained by each algorithm in comparison on 104-D WFG1 to WFG9
[Color figure can be viewed at wileyonlinelibrary.com]

it can be seen that the three proposed algorithms perform better than the three compared algorithms in seven out of nine test functions in Figure 4, and in four out of nine test functions in Figure 5, which overall perform better about 61% in the HV performance metric. It can illustrate that convergence, spread and cardinality of three proposed algorithms is better than REMMEDA and CSEA. Also these proposed algorithms have better HV than IMMOEA. In these plots, the rate of HV on IMMOEA is weaker than the three proposed algorithms. Therefore the proposed algorithms have better performance than three comparison algorithms.

Also, it can illustrate that the spread and uniformity of three proposed algorithms are faster and better than IMMOEA, RMMEDA and CSEA and three proposed algorithms have similar DM rates. But according to diagram of DM and HV, it seem that the uniformity of these proposed algorithms is weaker than REMMEDA and CSEA. Also, according to the results shown in the tables, these proposed algorithms is suitable for large-scale problems in decision variables. In IMMOEA test instances the dimension of decision variables is equal to 30 and in WFG test instances the dimension of decision variables is equal to 104. The advantages and disadvantages of the proposed algorithms are shown in Table 10.

In general, it can illustrate that the convergence, spread and cardinality proposed algorithm is better than three compared algorithms, but the uniformity is not smooth. The reason why these parameters are better in the proposed algorithm than in IMMOEA is that in this algorithm, we replace three methods of random forest variable importance with random grouping in order to prevent the elimination of some effective inverse models and losing the chance of allocated decision variables, to other objective spaces randomly: the classical permutation variable importance random forest method, the random forest naïve variable importance method and random forest cross validation variable importance method. In addition, random grouping process, cannot determine the best assignment of x_n to f_m and its performance is based on random assignment. In fact, the importance of a variable x_n in decision space for predicting f_m is determined by random forest feature importance and variables are selected that are most importance. This paper proposes a new model-based on random forest, in order to determine the best assignment of x_n to f_m . When the distribution of each gene (x_i) over any of fitness functions (f_j), are known, the best distributions can be select and it is not necessary to be assigned x_n to f_m randomly. On other hands, for decision variables assigned to the same group, their correlations can be implicitly increased by selecting the best distributions and the better HV criterion shows this. In the classical permutation variable importance random forest method, there is no natural cut off that can be discriminated important and nonimportant variables. Also, the predictor variables contain variables that do not

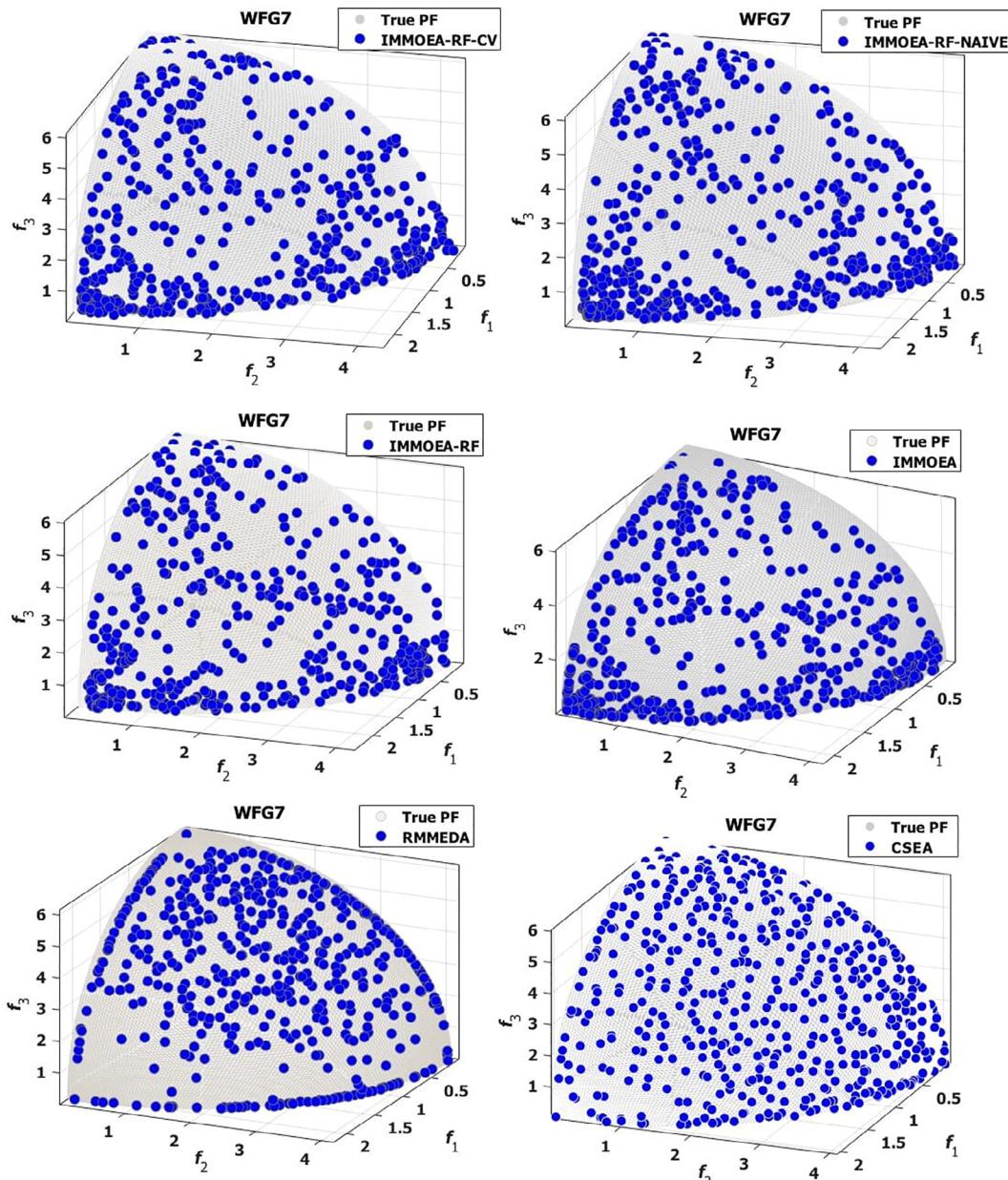


FIGURE 6 The nondominated solutions with the best hypervolume values obtained by each algorithm among 20 runs in the objective space on 104-D WFG7 [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 7 Statistical results of inverted generational distance values obtained by each algorithm in comparison on 30-D F1 to F10

Problem	M	D	FEs	IMMOEA-RF-CV	IMMOEA-RF-Naive	IMMOEA-RF	IMMOEA	RMMEDA	CSEA
F1	2	30	10 000	2.2437e-2 (2.74e-3) =	2.6363e-2 (3.80e-3) =	2.5594e-2 (3.05e-3) =	4.8621e-2 (2.21e-2) +	2.5267e-2 (1.10e-2) -	2.4306e-1 (5.06e-2)
F2	2	30	10 000	2.3043e-2 (3.89e-3) =	2.7422e-2 (4.31e-3) =	2.5106e-2 (3.75e-3) =	1.7698e-1 (1.20e-1) -	3.2720e-2 (1.97e-2) +	4.2168e-1 (7.21e-2)
F3	2	30	10 000	1.0295e-2 (5.07e-4) +	9.7394e-3 (5.22e-4) +	9.5079e-3 (3.09e-4) +	3.6387e-2 (1.18e-2) +	8.1272e-2 (1.98e-1) +	8.4074e-2 (2.45e-2)
F4	3	30	10 000	3.9984e-1 (7.98e-2) =	3.9957e-1 (7.59e-2) =	3.9908e-1 (7.46e-2) =	7.3525e-1 (1.81e-1) -	4.6770e-1 (3.06e-1) =	4.6856e-1 (3.08e-2)
F5	2	30	10 000	2.9277e-2 (3.44e-3) =	2.5916e-2 (3.09e-3) +	2.5996e-2 (2.29e-3) +	2.9292e-2 (3.75e-3) =	3.4339e-2 (1.12e-2) +	6.7773e-2 (1.05e-2)
F6	2	30	10 000	5.1775e-2 (6.63e-3) -	4.1504e-2 (4.46e-3) =	4.4168e-2 (4.30e-3) =	4.0252e-2 (5.28e-3) =	4.6028e-2 (1.55e-2) +	1.3459e-1 (2.17e-2)
F7	2	30	10 000	1.3622e-1 (5.16e-2) +	9.4990e-2 (3.21e-2) +	9.2502e-2 (1.88e-2) +	1.8925e-1 (1.65e-2) +	2.6470e-1 (1.63e-2) -	1.0702e-1 (1.67e-2)
F8	3	30	10 000	3.9385e-1 (7.53e-2) +	3.4087e-1 (3.96e-2) +	3.4091e-1 (4.27e-2) +	3.7556e-1 (3.70e-2) +	6.1511e-1 (4.54e-2) -	3.7255e-1 (7.80e-2)
F9	2	30	10 000	1.3981e-2 (2.69e-3) +	1.3373e-2 (2.14e-3) +	1.3677e-2 (1.74e-3) +	7.8510e-2 (1.76e-2) +	2.9270e-1 (4.54e-2) +	5.4591e-2 (8.23e-2)
F10	2	30	10 000	1.3029e+1 (8.91e+0) +	1.0008e+1 (7.03e+0) +	8.0290e+0 (6.79e+0) +	1.8781e+1 (6.75e+0) +	1.1572e+2 (1.11e+1) -	3.7869e+1 (9.16e+0)
+/-=				5/1/4	6/0/4	6/0/4	5/3/2	5/4/1	

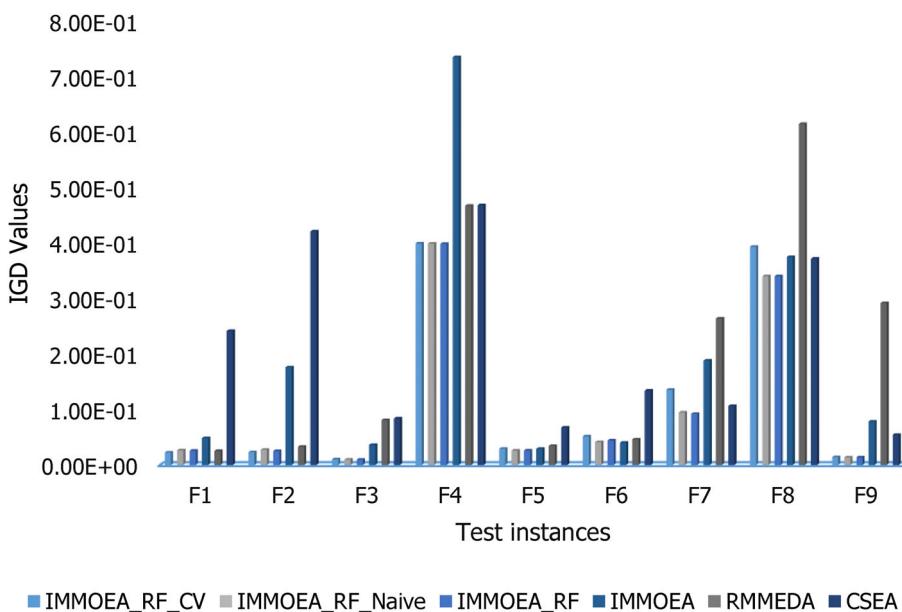


FIGURE 7 Statistical results of inverted generational distance values obtained by each algorithm in comparison on 30-D F1 to F9 [Color figure can be viewed at wileyonlinelibrary.com]

have their “own” effect on the response but are related to the response due to their relationship with truly influential predictor variables. Regarding the values variable importance, the negative values or values of zero, illustrate that variable does not improve the trees’ predictive strength, because the error rates are similar or even larger when applying an unchanged version of the variable. In contrast, a positive value for the variable importance, express that the variable at least some improves the trees’ predictive strength, because the error rates are smaller when applying the original version of the variable for deriving tree predictions. However it cannot be deduced that a positive value for the variable importance demonstrates a relative value. For this purpose, the random forest naïve variable importance method and random forest cross-validation variable importance method are suggested.

6 | CONCLUSIONS

In this paper, an IM-MOEA with Random Forest (eg, IMMOEA-RF-CV, IMMOEA-RF-Naive, and IMMOEA-RF) has been proposed. The idea is to apply the process of random forest variable importance with a random grouping that determines the best assignment of x_n to f_m in inverse model process. The reason for using this process is that some of the effective inverse models may be removed randomly in IM-MOEA, while the feature importance of them that removed, may be higher than the values that selected. Also, random grouping assignment, removes the chance of allocated decision variables, to other objective spaces, while the decision variable x_n maybe more feature importance for several f_n . In addition, IM-MOEA cannot determine the best assignment of x_n to f_m and its function is based on random assignment, while the random forests method can be used to identify predictor variables are the most important and termed as random forest feature importance. Using this method on IM-MOEA, that is, IM-MOEA-RF creates samples in the

TABLE 8 Statistical results of inverted generational distance values obtained by each algorithm in comparison on 104-D WFG1 to WFG9

Problem	M	D	FEs	IMMOEA-RF-CV	IMMOEA-RF-Naive	IMMOEA-RF	IMMOEA	RMMEDA	CSEA
WFG1	3	104	10000	2.0469e+0 (6.26e-2) +	2.0134e+0 (7.08e-2) -	1.9715e+0 (1.03e-1) +	1.9805e+0 (1.22e-1) +	2.2946e+0 (4.04e-2) -	1.4863e+0 (1.31e-2)
WFG2	3	104	10000	5.3169e-1 (3.76e-2) -	3.5352e-1 (2.00e-2) +	3.5955e-1 (2.41e-2) +	5.6389e-1 (2.59e-2) +	7.9901e-1 (3.45e-2) -	4.1657e-1 (7.29e-2)
WFG3	3	104	10000	4.1200e-1 (2.43e-2) +	3.0137e-1 (3.72e-2) +	3.0560e-1 (4.24e-2) +	5.7255e-1 (2.42e-2) +	8.3472e-1 (2.87e-2) -	4.8811e-1 (2.76e-2)
WFG4	3	104	10000	4.7528e-1 (3.53e-2) +	4.1461e-1 (2.72e-2) +	4.3135e-1 (2.86e-2) +	4.1298e-1 (1.09e-2) +	6.22275e-1 (1.71e-2) -	3.6582e-1 (1.40e-2)
WFG5	3	104	10000	7.2670e-1 (7.35e-2) =	3.8454e-1 (8.31e-2) +	3.7418e-1 (4.58e-2) +	6.4266e-1 (1.33e-2) =	6.3369e-1 (1.71e-2) -	4.3396e-1 (1.20e-2)
WFG6	3	104	10000	5.4553e-1 (5.66e-2) +	4.4530e-1 (4.77e-2) +	4.2735e-1 (4.92e-2) +	7.0454e-1 (1.66e-2) +	9.2834e-1 (2.63e-2) -	4.9416e-1 (1.24e-2)
WFG7	3	104	10000	5.3437e-1 (1.98e-2) +	4.5245e-1 (2.03e-2) +	4.6078e-1 (2.60e-2) +	6.0593e-1 (1.74e-2) +	7.3551e-1 (1.64e-2) -	5.4535e-1 (1.61e-2)
WFG8	3	104	10000	5.8410e-1 (3.22e-2) +	4.9421e-1 (1.97e-2) +	5.0365e-1 (1.85e-2) +	5.3515e-1 (1.24e-2) -	7.6786e-1 (3.60e-2) -	4.3938e-1 (1.68e-2)
WFG9	3	104	10000	3.9950e-1 (2.99e-2) +	3.5147e-1 (2.99e-2) +	3.4209e-1 (1.76e-2) +	6.5560e-1 (4.38e-2) +	9.0111e-1 (3.11e-2) -	5.3632e-1 (2.92e-2)
+/-=			8/1/0	9/0/0	9/0/0	8/0/1	8/0/0	0/9/0	

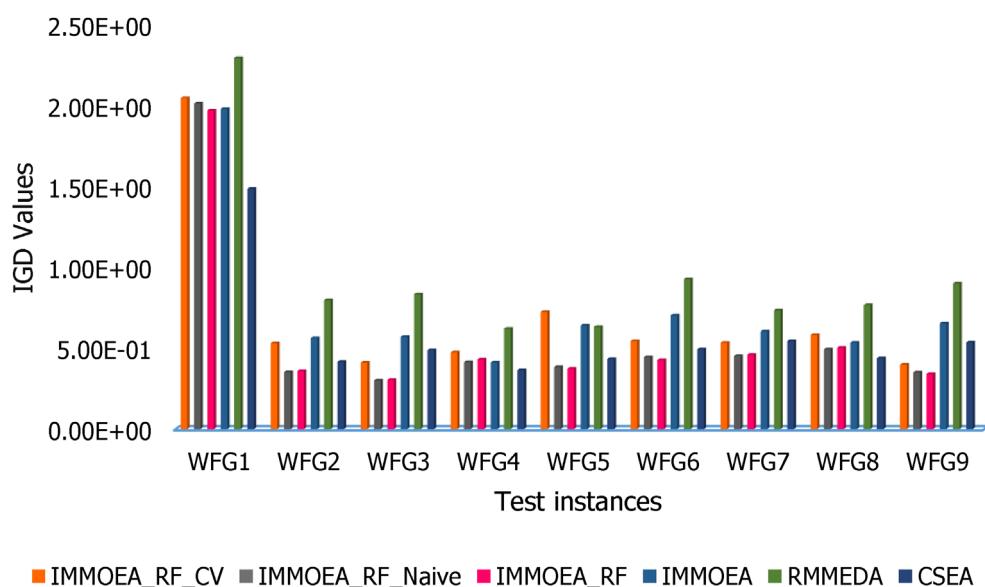


FIGURE 8 Statistical results of inverted generational distance values obtained by each algorithm in comparison on 104-D WFG1 to WFG9 [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 9 Relationship of categories with the performance metrics

Category	Performance Metrics					
	HV	IGD	DM	GD	PD	SP
Convergence	✓	✓		✓		
Spread	✓	✓	✓		✓	
Cardinality	✓					
Uniformity			✓			✓

Abbreviations: DM, diversity metric; HV, hypervolume; IGD, inverted generational distance; PD, pure diversity; SP, spacing.

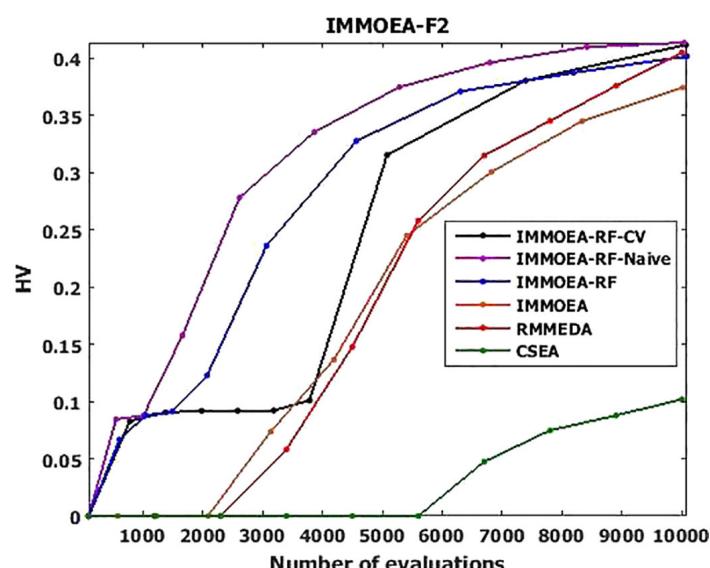


FIGURE 9 The hypervolume value on F2 for six comparison algorithms [Color figure can be viewed at wileyonlinelibrary.com]

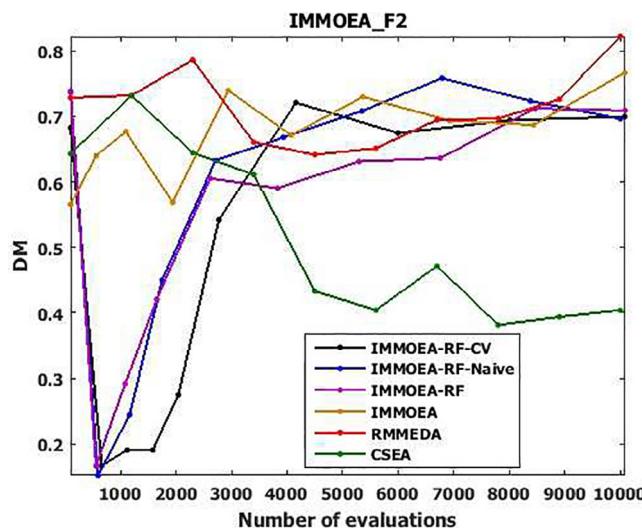
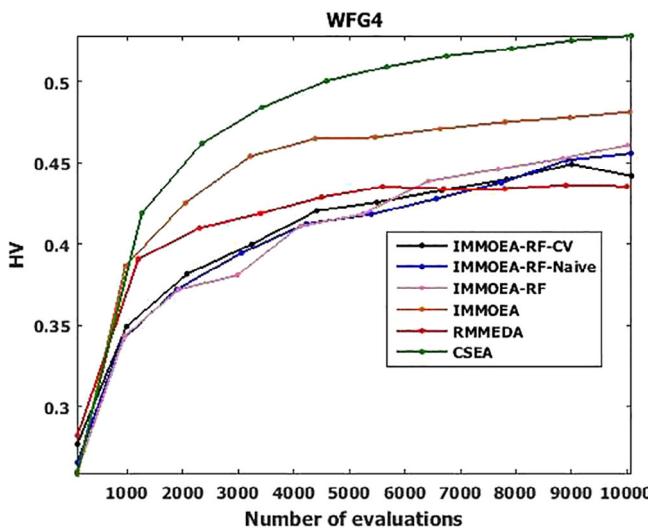
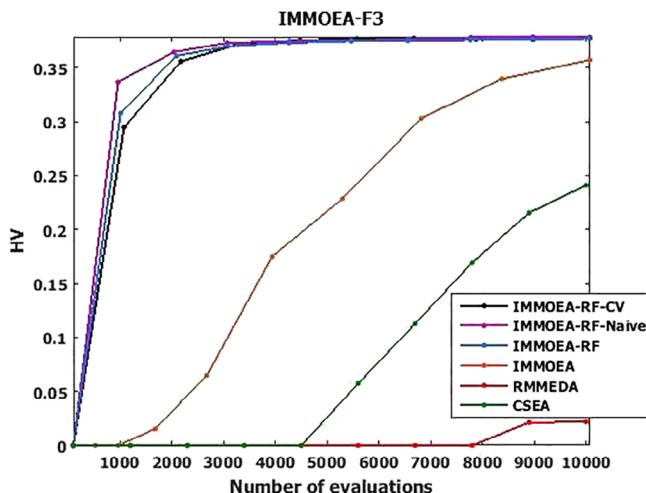




FIGURE 13 The diversity metric value on F3 for six comparison algorithms [Color figure can be viewed at wileyonlinelibrary.com]

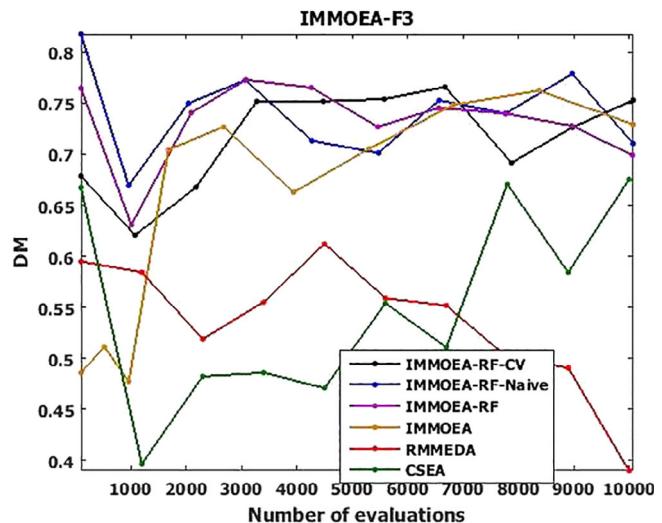


FIGURE 14 The diversity metric value on WFG4 for six comparison algorithms [Color figure can be viewed at wileyonlinelibrary.com]

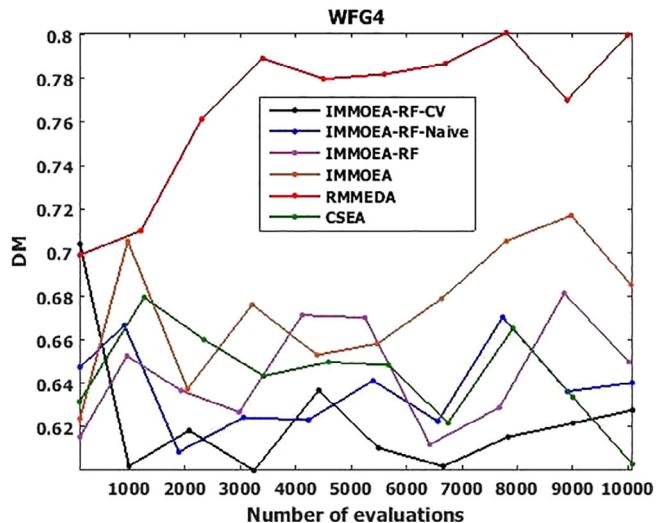


TABLE 10 The advantages and disadvantages of the proposed algorithms

Advantages	Disadvantages
Suitable for large-scale problems in decision variables.	The rate of uniformity performance metric is not better than compared algorithms, because the Gaussian distribution is more suitable for the exploitation stage owing to its narrow sampling range. Therefore in order to improve the uniformity performance, in future work, other distributions can be used, such as the Cauchy distribution along with the Gaussian distribution.
It can determine the best assignment of x_n to f_m in regression model (in IM-MOEA, this process is randomize).	
The best distributions of x_n to f_m can be selected in regression model.	
Increase of their correlations.	
The effective inverse models, have not be removed.	
Increase rate of convergence, spread and cardinality.	

objective space in creating offspring, which is very convenient for generating solutions in preferred regions. The proposed method performs robustly competitive on a variety of test instances compared to four representative MOEAs, as long as the shapes of Pareto set are not too complex. The rate of uniformity performance metric is not better than compared algorithms, because the Gaussian distribution is more suitable for the exploitation stage owing to its narrow sampling range. Therefore in order to improve the uniformity performance, in future work, other distributions can be used, such as the Cauchy distribution along with the Gaussian distribution. The computational efficiency of proposed method is not ideal and it is worth further investigation to decrease the run time in solving small to medium optimization problems as future work. Also, its potential ability this method to solve many-objective optimization problems. The proposed method can be applied in the interdisciplinary fields for future research directions. Some of these are as follows: modeling and optimization of flexible manufacturing systems,⁵⁴ visual analytics solution for scheduling processing phases,⁵⁵ the number of out-of-cash ATMs and duration of out-of-cash status,⁵⁶ optimization of hybrid wind and solar renewable energy system,⁵⁷ optimization of parqueting of the concentrator of photovoltaic thermal module,⁵⁸ verifying the gaming strategy of self-learning game,⁵⁹ optimization of the process of anaerobic bioconversion of liquid organic wastes,⁶⁰ efficiency optimization of indoor air disinfection by radiation exposure for poultry breeding.⁶¹

ACKNOWLEDGMENTS

Financial support from research office of Department of Computer, South Tehran Branch, Islamic Azad University is acknowledged.

ORCID

Ali Broumandnia  <https://orcid.org/0000-0001-5145-2013>

REFERENCES

- Janitza S, Celik E, Boulesteix A-L. A computationally fast variable importance test for random forests for high-dimensional data. *ADAC*. 2018;12(4):885-915.
- Zhang Q, Zhou A, Jin Y. RM-MEDA: a regularity model-based multiobjective estimation of distribution algorithm. *IEEE Trans Evol Comput*. 2008;12(1):41-63.
- Krishnamoorthy Murugan, Suresh Sailakshmi, Alagappan Solaiappan. Deep learning techniques and optimization strategies in big data analytics: automated transfer learning of convolutional neural networks using Enas algorithm. Paper presented at: Proceedings of the Deep Learning Techniques and Optimization Strategies in Big Data Analytics. IGI Global; 2020:142-153.
- Krejca Martin. Theoretical analyses of evolutionary algorithms with a focus on estimation of distribution algorithms. Proceedings of the 10th Ph. D. Retreat of the HPI Research School on Service-oriented Systems Engineering; 2018, Vol. 111:129.
- Gao S, Silva CW. Estimation distribution algorithms on constrained optimization problems. *Applied Mathematics and Computation*. 2018;339:323-345.
- Cheng R, He C, Jin Y, Yao X. Model-based evolutionary algorithms: a short survey. *Complex Intell Syst*. 2018;4(4):283-292.
- Allmendinger R, Emmerich MTM, Hakanen J, Jin Y, Rigoni E. Surrogate-assisted multicriteria optimization: complexities, prospective solutions, and business case. *J Multi-Criteria Decis Anal*. 2017;24(1-2):5-24.
- Jin Yaochu, Sendhoff Bernhard. Connectedness, regularity and the success of local search in evolutionary multi-objective optimization. Paper presented at: Proceedings of the 2003 Congress on Evolutionary Computation, 2003 CEC'03; 2003:1910-1917; IEEE.
- Giagkiozis Ioannis, Fleming Peter J. Increasing the density of available Pareto optimal solutions; 2012.
- Giagkiozis I, Fleming PJ. Pareto front estimation for decision making. *Evol Comput*. 2014;22(4):651-678.



11. Cheng R, Jin Y, Narukawa K, Sendhoff B. A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling. *IEEE Trans Evol Comput.* 2015;19(6):838-856.
12. Schulz E, Speekenbrink M, Krause A. A tutorial on Gaussian process regression: modelling, exploring, and exploiting functions. *J Math Psychol.* 2018;85:1-16.
13. Ma X, Li X, Zhang Q, et al. A survey on cooperative co-evolutionary algorithms. *IEEE Trans Evol Comput.* 2018;23(3):421-441.
14. Khan N, Goldberg DE, Pelikan M. *Multi-objective Bayesian optimization algorithm (IlliGAL Report No. 2002009)*. Urbana, IL: University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory; 2002.
15. Schwarz J, Ocenasek J. Multiobjective bayesian optimization algorithm for combinatorial problems: theory and practice. *Neural Netw World.* 2001;11(5):423-442.
16. Laumanns Marco, Ocenasek Jiri. Bayesian optimization algorithms for multi-objective optimization. Paper presented at: Proceedings of the International Conference on Parallel Problem Solving from Nature; 2002:298-307; Springer.
17. Zhong Xiaoping, Li Weiji. A decision-tree-based multi-objective estimation of distribution algorithm. Paper presented at: Proceedings of the 2007 International Conference on Computational Intelligence and Security (CIS 2007); 2007:114-118; IEEE.
18. Karshenas H, Santana R, Bielza C, Larranaga P. Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables. *IEEE Trans Evol Comput.* 2013;18(4):519-542.
19. Bosman PAN, Thierens Dirk. Multi-objective optimization with diversity preserving mixture-based iterated density estimation evolutionary algorithms. *Int J Approx Reason.* 2002;31(3):259-289.
20. Costa Mario, Minisci Edmondo. MOPED: a multi-objective parzen-based estimation of distribution algorithm for continuous problems. Paper presented at: Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization; 2003:282-294; Springer.
21. Pelikan Martin, Sastry Kumara, Goldberg David E. Multiobjective HBOA, clustering and scalability. Tech. rep., Proceedings of the Genetic and Evolutionary; 2005.
22. Sastry Kumara, Goldberg David E, Pelikan Martin. Limits of scalability of multiobjective estimation of distribution algorithms. Paper presented at: Proceedings of the 2005 IEEE Congress on Evolutionary Computation; 2005:2217-2224; IEEE.
23. Pan L, He C, Ye T, Wang H, Xingyi Z, Yaochu J. A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. *IEEE Trans Evol Comput.* 2018;23(1):74-88.
24. Jin Y. Surrogate-assisted evolutionary computation: recent advances and future challenges. *Swarm Evol Comput.* 2011;1(2):61-70.
25. Liu B, Zhang Q, Gielen GGE. A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems. *IEEE Trans Evol Comput.* 2013;18(2):180-192.
26. Li X, Yao X. Cooperatively coevolving particle swarms for large scale optimization. *IEEE Trans Evol Comput.* 2011;16(2):210-224.
27. Yang Z, Tang K, Yao X. Large scale evolutionary optimization using cooperative coevolution. *Inf Sci.* 2008;178(15):2985-2999.
28. Gee SB, Tan KC, Alippi C. Solving multiobjective optimization problems in unknown dynamic environments: an inverse modeling approach. *IEEE Trans Cybern.* 2016;47(12):4223-4234.
29. Tian Y, Wang H, Zhang X, Jin Y. Effectiveness and efficiency of non-dominated sorting for evolutionary multi-and many-objective optimization. *Complex Intell Syst.* 2017;3(4):247-263.
30. Louppe G, Wehenkel L, Sutera A, Geurts P. Understanding variable importances in forests of randomized trees. In: Burges CJC, Bottou L, Welling M, Ghahramani Z, Weinberger KQ, eds. *Advances in neural information processing systems*. Red Hook, NY: Curran Associates, Inc.; 2013:431-439.
31. Couronné R, Probst P, Boulesteix A-L. Random forest versus logistic regression: a large-scale benchmark experiment. *BMC Bioinform.* 2018;19(1):270.
32. Zhou Aimin, Zhang Qingfu, Zhang Guixu. A multiobjective evolutionary algorithm based on decomposition and probability model. Paper presented at: Proceedings of the 2012 IEEE Congress on Evolutionary Computation; 2012:1-8; IEEE.
33. Denisko D, Hoffman MM. Classification and interaction in random forests. *Proc Natl Acad Sci.* 2018;115(8):1690-1692.

34. Rasmussen CE. Gaussian processes in machine learning. *Summer School on Machine Learning*. New York, NY: Springer; 2003:63-71.
35. Wackernagel H. *Multivariate Geostatistics: An Introduction with Applications*. Berlin, Heidelberg / Germany: Springer Science & Business Media; 2013.
36. Cressie N. Statistics for spatial data: Wiley series in probability and statistics Wiley-interscience. John Wiley and Sons, New York, NY 1993;15:105–209.
37. Barber D. *Bayesian Reasoning and Machine Learning*. Cambridge, MA: Cambridge University Press; 2012.
38. Rasmussen Carl Edward, Williams Christopher KI. Gaussian Processes for Machine Learning; 2006.
39. Freedman DA. *Statistical Models: Theory and Practice*. Cambridge, MA: Cambridge University Press; 2009.
40. Umlauf Nikolaus, Adler Daniel, Kneib Thomas, Lang Stefan, Zeileis Achim. Structured additive regression models: An R interface to BayesX. Working Papers in Economics and Statistics; 2012.
41. Do Chuong B. Gaussian Processes. Stanford University, Stanford, CA; 2017. Accessed December 5, 2007.
42. Pham H, Olafsson S. Bagged ensembles with tunable parameters. *Comput Intell*. 2019;35(1):184-203.
43. Singh PK, Sarkar R, Nasipuri M. Correlation-based classifier combination in the field of pattern recognition. *Comput Intell*. 2018;34(3):839-874.
44. Kohavi Ron. A study of cross-validation and bootstrap for accuracy estimation and model selection. Ijcai; 1995:1137-1145; Montreal, Canada.
45. James G, Witten D, Hastie T, Tibshirani R. *An Introduction to Statistical Learning*. New York, NY: Springer; 2013.
46. Huband S, Hingston P, Barone L, While L. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans Evol Comput*. 2006;10(5):477-506.
47. Li M, Yao X. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Comput Surv (CSUR)*. 2019;52(2):26.
48. Van Veldhuizen David A, Lamont Gary B. Evolutionary computation and convergence to a pareto front. Paper presented at: Late Breaking Papers at the Genetic Programming 1998 Conference; 1998:221-228.
49. Wang H, Jin Y, Yao X. Diversity assessment in many-objective optimization. *IEEE Trans Cybern*. 2016;47(6):1510-1522.
50. Schott Jason R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization No. AFIT/CI/CIA-95-039. Air force inst of tech Wright-Patterson afb OH; 1995.
51. Goli A, Zare HK, Tavakkoli-Moghaddam R, Sadegheih A. Multiobjective fuzzy mathematical model for a financially constrained closed-loop supply chain with labor employment. *Comput Intell*. 2019;36(1):4-34.
52. Zhou Aimin, Zhang Qingfu, Jin Yaochu, Tsang Edward, Okabe Tatsuya. A model-based evolutionary algorithm for bi-objective optimization. Paper presented at: Proceedings of the 2005 IEEE Congress on Evolutionary Computation; 2005:2568-2575; IEEE.
53. Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*. Hoboken, NJ: John Wiley & Sons; 2001.
54. Lechuga, G. P., & Sánchez, F. M. Modeling and optimization of flexible manufacturing systems: a stochastic approach. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2018:539-546; Springer.
55. Thomas J Joshua, Belaton Bahari, Khader Ahamad Tajudin. Visual analytics solution for scheduling processing phases. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2018:395-408; Springer.
56. Ozer Fazilet, Toroslu Ismail Hakki, Karagoz Pinar, Yucel Ferhat. Dynamic Programming Solution to ATM Cash Replenishment Optimization Problem. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2018:428-437; Springer.
57. Geleta Diriba Kajela, Manshahia Mukhdeep Singh. Optimization of hybrid wind and solar renewable energy system by iteration method. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2018:98-107; Springer.
58. Sinitsyn Sergey, Panchenko Vladimir, Kharchenko Valeriy, Vasant Pandian. Optimization of Parqueting of the Concentrator of Photovoltaic Thermal Module. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2019:160-169; Springer.
59. Zaw Hein Htoo, Hlaing Swe Zin. Verifying the gaming strategy of self-learning game by using PRISM-games. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2019:148-159; Springer.



60. Kovalev Andrey, Kovalev Dmitriy, Panchenko Vladimir, Kharchenko Valeriy, Vasant Pandian. Optimization of the process of anaerobic bioconversion of liquid organic wastes. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2019; :170-176; Springer.
61. Dovlatov Igor, Yuferov Leonid, Pavkin Dmitriy. Efficiency optimization of indoor air disinfection by radiation exposure for poultry breeding. Paper presented at: Proceedings of the International Conference on Intelligent Computing & Optimization; 2019:177-189; Springer.

How to cite this article: Gholamnezhad P, Broumandnia A, Seydi V. An inverse model-based multiobjective estimation of distribution algorithm using Random-Forest variable importance methods. *Computational Intelligence*. 2020;1-39.

<https://doi.org/10.1111/coin.12315>