

## Research Article

# A Kriging Model-Based Expensive Multiobjective Optimization Algorithm Using R2 Indicator of Expectation Improvement

Ding Han  and Jianrong Zheng 

*School of Mechanical and Power Engineering, East China University of Science and Technology, Shanghai 200237, China*

Correspondence should be addressed to Jianrong Zheng; jrzheng@126.com

Received 20 April 2020; Revised 27 May 2020; Accepted 28 May 2020; Published 27 June 2020

Guest Editor: Juan Carlos Leyva-Lopez

Copyright © 2020 Ding Han and Jianrong Zheng. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Most of the multiobjective optimization problems in engineering involve the evaluation of expensive objectives and constraint functions, for which an approximate model-based multiobjective optimization algorithm is usually employed, but requires a large amount of function evaluation. Aiming at effectively reducing the computation cost, a novel infilling point criterion EIR2 is proposed, whose basic idea is mapping a point in objective space into a set in expectation improvement space and utilizing the R2 indicator of the set to quantify the fitness of the point being selected as an infilling point. This criterion has an analytic form regardless of the number of objectives and demands lower calculation resources. Combining the Kriging model, optimal Latin hypercube sampling, and particle swarm optimization, an algorithm, EIR2-MOEA, is developed for solving expensive multiobjective optimization problems and applied to three sets of standard test functions of varying difficulty and comparing with two other competitive infill point criteria. Results show that EIR2 has higher resource utilization efficiency, and the resulting nondominated solution set possesses good convergence and diversity. By coupling with the average probability of feasibility, the EIR2 criterion is capable of dealing with expensive constrained multiobjective optimization problems and its efficiency is successfully validated in the optimal design of energy storage flywheel.

## 1. Introduction

Science and engineering practice possess a large number of multiobjective optimization problems (MOP) [1] whose objectives often conflict with each other and need to be optimized simultaneously, where the Pareto set is desired. Evolutionary Algorithm (EA) [2] has proven to be very suitable for MOP, some typical algorithms include NSGA-II [3], SPEA2 [4], MOEA/D [5], and MOPSO [6]. However, the success of these algorithms depends heavily on a large number of function evaluations and can only be applied in cases that function evaluation is cheap computationally. To handle some expensive MOPs, it is a desire to develop some efficient algorithms.

To reduce the number of expensive evaluation required, approximate/surrogate model [7], which mimic the input-output relationship of expensive function but is cheap-to-evaluate, is often employed. Polynomial response surfaces

[8], support vector machines [9], neural networks [10], and Kriging [11] are commonly used approximate models.

Currently, there are two ways to integrate the approximate model with EA. The first one is to first construct a static global approximate model based on all sample points already evaluated, and the EA algorithm then searches for the optimal solution. However, with limited computational resources, it is hard to guarantee the approximate model's accuracy over the entire design space. The second one utilizes an approximate model in a dynamic manner by first constructing a rough approximate model with a fraction of sample points generated by some Design of Experiments (DoE) [12] techniques, and then maximizing some infilling point criterion (IPC) [13], a new point is located by EA and is evaluated by expensive function. This process is repeated until resources are exhausted. The second manner has the advantage of high efficiency in using computation resources and is used in this paper.

The infilling point criterion [14] plays a critical role in the approximate model-based optimization method. Expectation Improvement (EI), proposed in Efficient Global Optimization (EGO) [15] by Jones, may be one of the most researched methods in the literature. It uses the Kriging model as an approximate model and considers not only the predicted value but also modeling uncertainty, achieving a good balance between exploration and exploitation in the optimization process. Other IPCs include statistical lower bound [16] and probability of improvement [17].

However, these criteria are proposed originally only for single optimization and have to be modified for MOPs. Knowles first introduced EI into the field of multiobjective optimization and presented ParEGO [18], where the Kriging model is used and multiple objectives are converted into a single objective through a parameterized scalarizing weight vector, and then followed by the direct application of EGO, the weight vector changed randomly as iteration proceeded. Jones pointed out ParEGO is apt to favor model accuracy rather than finding nondominated solutions and demands more iterations before convergence and hence proposed Pareto Expected Improvement index (PEI) [19], which considers both EI and the probability of being Pareto optimal solution. Multi-EGO [20] builds the Kriging model and calculates separately EI for each objective and calls MOGA [21] as a solver to output a set of candidate points, from which a point is selected as an infilling point. MOEA/D-EGO [22] borrowed the ideas from ParEGO, and it generated not a single but a set of weight vectors that distributed uniformly in the objective space to form single objective optimization subproblems involving EI, optimizing jointly those subproblems produces multiple infill points and in this way parallelly infilling is achieved. Recently, some IPCs compatible with multiobjective optimization were proposed, such as MaxMin improvement [23] and expectation hypervolume improvement [24]. Svensson defined MaxMin improvement in the analytic form when the number of the objective function is two and suggested using the Monte Carlo method (MCM) [25] to calculate MaxMin value when the number of the objective is more than two.

In this paper, a novel IPC is proposed for expensive multiobjective optimization. The outstanding feature of the IPC is to map one point in the objective space to a set in the expectation improvement space, where the fitness of the point is defined as a Quality indicator (QI) [26] value of the set. In this way, QI plays a role in guiding optimization towards an infilling point, rather than evaluating the nondominated solution set after optimization. There are a large number of candidate QIs in the literature, such as IGD [27, 28],  $\epsilon$ -indicator [29], SDE indicator [30], and Hypervolume (HV) [31]. However, these QIs do not meet the requirements of easy-using and low-computation cost to some extent. For example, both IGD and  $\epsilon$ -indicator demand Pareto front (PF) as the reference set, which is impossible in practical problems; HV does not require PF but a reference point, for which, if set unreasonably, will mislead the optimization search direction [32]. Besides, HV possesses high computational complexity, which increases exponentially with the number of objectives. Some QIs have a

specific bias, for example,  $\epsilon$ -indicator is more convergence-oriented, and as a result, optimization process may suffer from premature convergence, while SDE prefers diversity [33] which may make the optimization period too long [34]. R2 indicator [35, 36], having some desirable properties in common with HV but is cheaper to evaluate, attracted widespread interest in recent years. In addition, it is weakly monotonic [37] and does not require PF to be provided; hence, it is adopted in this article.

The rest of the paper is organized as follows. In Section 2, the mathematical background of Kriging, EI, and R2 indicator is reviewed. Then, the proposed EIR2 indicator and framework of the whole optimization algorithm are presented in Section 3. To evaluate the performance of the proposed IPC, a series of experiments over three benchmark test function suites against two competitors IPCs is conducted in Section 4, their results are measured by three performance metrics and then analyzed in Section 5, and followed by Section 6, where the proposed EIR2 is modified and applied to a real-world constrained expensive MOP, an optimization design of an energy storage flywheel. Lastly, Section 7 gives the conclusion and future research direction.

## 2. Background

*2.1. Multiobjective Optimization Problem.* Without loss of generality, a multiobjective optimization problem is formulated as follows:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_m(\mathbf{x})]^T \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, p \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \dots, q \\ & \mathbf{x}^{\min} \leq \mathbf{x} \leq \mathbf{x}^{\max}, \end{aligned} \quad (1)$$

where  $\mathbf{F}(\mathbf{x})$  represents the objective function vector which contains  $m$  objective functions,  $g_i$  and  $h_j$  stand for inequality and equality constraint function with the total number  $p$  and  $q$ , respectively, and  $\mathbf{x} = [x_1, \dots, x_d]^T$  stands for design vector with lower bound  $\mathbf{x}^{\min}$  and upper bound  $\mathbf{x}^{\max}$ . A solution  $\mathbf{u}$  is said to dominate another one  $\mathbf{v}$  if and only if  $f_i(\mathbf{u}) \leq f_i(\mathbf{v}) \forall i = 1, \dots, m$  and  $f_i(\mathbf{u}) < f_i(\mathbf{v}) \exists i = 1, \dots, m$ , written as  $\mathbf{u} < \mathbf{v}$ . A solution  $\mathbf{x}^*$  is called Pareto optimal if there is no  $\mathbf{x}$  satisfying  $\mathbf{x} < \mathbf{x}^*$  (such  $\mathbf{x}^*$  is also referred to as nondominated by  $\mathbf{x}$ ). The set of all the Pareto optimal solutions is called the Pareto Set (PS), whose image in the objective space is called Pareto Front (PF).

In practice, objective functions and constraint functions may come from finite element simulation; hence, they have no explicit analytic expressions and are usually computationally expensive to evaluate. Under this condition, we are interested in identifying a set of nondominated solutions, called approximated Pareto Set/Front, to represent the true Pareto Set/Front, meanwhile consuming as less computing resources as possible.

*2.2. Kriging Model and Expectation Improvement.* Kriging model is an unbiased estimation model with smallest estimated variance, excellent high-dimensional, and nonlinear

fitting capability. It assumes the relationship between input  $\mathbf{x}$  and predicted output  $\hat{y}$  as follows:

$$\hat{y}(\mathbf{x}) = \mu + z(\mathbf{x}), \quad (2)$$

where  $\mu$  is a constant and  $z(\mathbf{x}) \sim N(0, \sigma^2)$  is the Gaussian random variable, which represent global trend and local deviation of the output separately.

Having a set of known observations  $S = \{\mathbf{x}^i, y^i, i = 1, \dots, N\}$ , our job is to search for the best values of  $\mu$  and  $\sigma$  when predicting at a new point  $\mathbf{x}$ . In the design space, the relationship between two different points  $\mathbf{x}^i$  and  $\mathbf{x}^j$  is characterized by correlation functions, such as the Gaussian function:

$$R(z(\mathbf{x}^i), z(\mathbf{x}^j); \boldsymbol{\theta}) = \prod_{k=1}^d \exp\left(-\theta_k |x_k^i - x_k^j|\right), \quad (3)$$

where hyperparameter  $\boldsymbol{\theta} = [\theta_1, \dots, \theta_d]^T$  controls the smoothness of prediction function and is usually determined through maximizing a likelihood function of  $R$  by some intelligent optimization algorithms such as particle swarm optimization. According to optimization theory, analytical form of  $\mu$  and  $\sigma$  are obtained:

$$\begin{aligned} \hat{\mu} &= \frac{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}}, \\ \hat{\sigma}^2 &= \frac{(\mathbf{y} - \mathbf{1}\hat{\mu})^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu})}{N}, \end{aligned} \quad (4)$$

where  $\mathbf{y} = [y_1, \dots, y_d]^T$  is known as the output vector and  $\mathbf{R}$  is a matrix of covariance between any two points in  $S$ , that is,  $\mathbf{R}_{i,j} = R(z(\mathbf{x}^i), z(\mathbf{x}^j))$ . Finally, at the new point  $\mathbf{x}$ , we get the predicted value and variance:

$$\begin{aligned} \hat{y}(\mathbf{x}) &= \hat{\mu} + \mathbf{r}^T \mathbf{R}^{-1} (\mathbf{y} - \mathbf{1}\hat{\mu}), \\ \hat{s}^2(\mathbf{x}) &= \hat{\sigma}^2 \left[ 1 - \mathbf{r}^T \mathbf{R} \mathbf{r} + \frac{(1 - \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r})^2}{\mathbf{1}^T \mathbf{R}^{-1} \mathbf{1}} \right], \end{aligned} \quad (5)$$

where  $\mathbf{r}$  is a correlation vector between point  $\mathbf{x}$  and all points in  $S$ . Unlike other approximate models such as response surface [38] and neural network [39], which can only give out the predicted value, Kriging is capable of outputting predicted variance as a byproduct, which can be regarded as a predicting uncertainty at  $\mathbf{x}$ .

With Kriging, the value at a new point  $\mathbf{x}$  is treated as a Gaussian random variable, that is,  $y(\mathbf{x}) \sim N(\hat{\mu}, \hat{s}^2)$ . An improvement quantity is designed as  $I(y(\mathbf{x}), y_{\min}) = \max(y(\mathbf{x}) - y_{\min}, 0)$ , where  $y_{\min}$  is the smallest  $y$  value in  $S$ . It is noted that only those  $y(\mathbf{x})$  that are less than  $y_{\min}$  produce improvement. The expectation of improvement (EI) [40] has a closed form, that is,

$$EI(y(\mathbf{x}), y_{\min}) = \begin{cases} (y_{\min} - \hat{y}(\mathbf{x})) \Phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right) + \hat{s} \phi\left(\frac{y_{\min} - \hat{y}(\mathbf{x})}{\hat{s}(\mathbf{x})}\right), & \text{if } \hat{s} > 0, \\ 0, & \text{if } \hat{s} = 0, \end{cases} \quad (6)$$

where  $\Phi$  and  $\phi$  are the probability distribution and probability density functions, respectively.

**2.3. R2 Indicator.** The R2 indicator [41] is a unitary indicator, which was proposed to evaluate the performance of the nondominated solution set  $Q$ , given as

$$R2(Q, U) = \frac{1}{|U|} \sum_{u \in U} \max_{q \in Q} \{u(q)\}, \quad (7)$$

where  $U$  is a discrete and finite set of utility functions  $u$ , for which there are many options to choose, such as weight sums and penalty-based boundary intersection [42].

### 3. Proposed Method

**3.1. Expectation Improvement R2 Indicator.** The main difficulty in extending EI infilling criterion to MOPs lies in the fact that we have to specify  $y_{\min}$  before calculating the EI value, which is impractical as no such solution exists as being optimal with respect to every objective, but a set of

nondominated solution found during previous optimization process. It is now highly desired to find an infilling point having large EI values with respect to all points in the current nondominated set. This motivates us to treat each nondominated solution as current optima and then calculate the EI value in each objective, resulting in a vector function:

$$H(\mathbf{x}, \mathbf{p}) = [EI_1(\mathbf{x}, \mathbf{p}), \dots, EI_m(\mathbf{x}, \mathbf{p})]^T, \quad (8)$$

where  $\mathbf{p} = [p_1, \dots, p_m]^T$  is a point in current nondominated set  $P$  and  $EI_j(\mathbf{x}, \mathbf{p}) = EI(y_j(\mathbf{x}), p_j)$ . By this function, we obtain a vector consisting of all EI values for which  $p_j$  is deemed as  $y_{\min}$  in objective  $j$ . Obviously, we need to build  $m$  Kriging models, each corresponding to one objective function. After continually applying function  $H$  to all nondominated solutions, we get a set composed of  $|P|$  points, defined as

$$Q(\mathbf{x}) = \{\mathbf{q} \mid \mathbf{q} = H(\mathbf{x}, \mathbf{p}), \mathbf{p} \in P\}, \quad (9)$$

where  $\mathbf{q} = [q_1, \dots, q_m]^T$ . In essence, we mapped point  $\mathbf{x}$  in design space to a set  $Q(\mathbf{x})$  in another space, called

expectation improvement space, where the R2 indicator of  $Q(\mathbf{x})$  is defined as fitness of  $\mathbf{x}$  being selected as the infilling point. The mapping from objective space to the EI space is illustrated in Figure 1.

Combining EI and R2 indicator, a novel indicator for MOP, Expectation Improvement R2 Indicator (EIR2), is defined as follows:

$$\text{EIR2}(\mathbf{x}, P, U) = -\frac{1}{|U|} \sum_{u \in U} \max_{\mathbf{q} \in Q(\mathbf{x})} \{u(\mathbf{q})\}. \quad (10)$$

As EIR2 is a scalar indicator, some single optimization algorithm can be employed to determine the optimal solution as the infilling point. Besides, EIR2 actually defines an indicator template in which other utility functions can be specified. In this paper, Tchebyche function is employed as utility function because it can tackle MOPs with convex or concave Pareto front, defined as

$$u_{\lambda}(\mathbf{q}) = -\max_{j \in \{1, \dots, m\}} \lambda_j |z_j^* - q_j|, \quad (11)$$

where  $\lambda = [\lambda_1, \dots, \lambda_m]^T$  is weight vector satisfying  $\lambda_j \geq 0$  and  $\sum_{j=1}^m \lambda_j = 1$ . These weight vectors are uniformly distributed in the objective space and are collected in the weight set  $\Lambda$ .  $\mathbf{z}^* = [z_1^*, \dots, z_m^*]^T$  is the ideal point, that is,  $z_j^* = \min(q_j), \forall \mathbf{q} \in Q$ . As all EI values are nonnegative, a natural selection is  $\mathbf{z}^* = \mathbf{0}$ . To summarize, the EIR2 indicator has the form

$$\text{EIR2}(\mathbf{x}, P, \Lambda) = \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} \min_{\mathbf{p} \in P} \{ \max_{j=1, \dots, m} \{ \lambda_j \times \text{EI}_j(\mathbf{x}, \mathbf{p}) \} \}. \quad (12)$$

The point that maximizes the EIR2 indicator is recognized as an infilling point for expensive evaluation.

**3.2. Weight Vectors Generating.** The utility function is actually an optimization subproblem parameterized by the weight vector, by which the optimization process can be controlled. If the geometry characteristics of PF are known, such as being convex/concave, a set of weight vectors can be distributed consistent with the PF in the objective space, by which a more accurate approximate PF with uniform distribution is more likely to be obtained. However, there is often no information about the PF in priori, which is common for practical MOP; hence, a better choice in this situation is to treat all objectives equally important and generate a set of weight vectors evenly distributed in the objective space.

To help generate weight vector  $\lambda$ , an auxiliary set is introduced  $L = \{0/H, 1/H, \dots, H/H\}$ , where  $H$  is a user-specified positive integer. We can choose randomly  $m-1$  element from  $L$  as the first  $m-1$  elements of  $\lambda$  and assign the value of  $\lambda_m$  to ensure  $\sum_{i=1}^m \lambda_i = 1$ . Exhausting every possible combination, there are totally  $C_{H+m-1}^{m-1}$  weight vectors generated. Figure 2 illustrates all 21 weight vectors generated using parameter  $m=3$  and  $H=5$ .

**3.3. Comparison with Other IPCs.** The novelty of EIR2 is that fitness assessment of point is carried out not in the objective

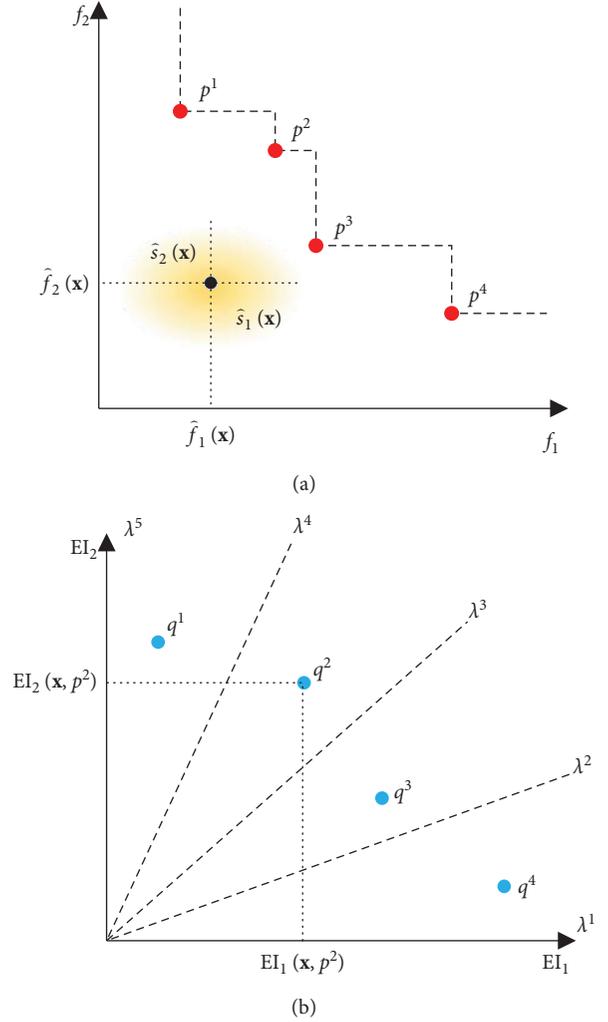


FIGURE 1: (a) A point with its predicted value and standard deviation in 2D objective space and (b) its corresponding mapping set in 2D EI space.

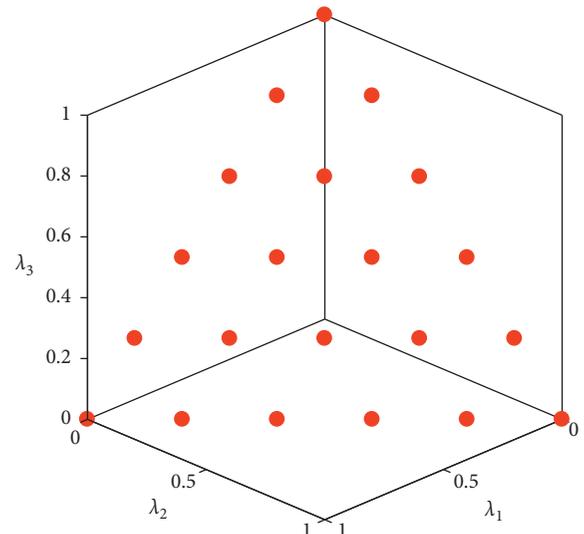


FIGURE 2: Evenly distributed weight vectors in objective space with  $m=3$  and  $H=5$ .

space but in the EI space, where the R2 indicator is calculated for the mapping set of the point. Since the EI indicator has a closed function form, so does EIR2 regardless of the number of objective functions. In addition, the computational complexity of EIR2 is  $O(m * |P| * |\Lambda|)$ , which is considerably less than that of hypervolume [26].

ParEGO also uses the Chebyshev function; however, its purpose is to integrate multiple objectives into one single objective before applying the EGO algorithm. Therefore, in essence, the infilling point criterion used in ParEGO is EI.

With regard to the MaxMin indicator, despite analytic form being existing, its calculation has to resort to dividing the objective space into many blocks on which tedious multivariable integration is carried out. Even worse, the number of blocks increases exponentially with the growth of the number of objective. So, in practice, we only apply this analytic form when  $m = 2$  while using the Monte Carlo method when  $m > 2$ .

Figure 3 shows contour plots of three IPCs, namely, ParEGO, MaxMin, and EIR2 applied in test function TEST2 defined in Table 1, where lighter color denotes higher value. In these plots, 20 known sample points are represented by red points, and blue points represent the true Pareto set, which are composed of three disconnected point subsets. It is easy to find that ParEGO can identify only one Pareto subset out of three, while MaxMin and EIR2 successfully locate the three Pareto subsets. In the region, where no Pareto solution exists, the function value of EIR2 is almost constant, while the function landscape of MaxMin shows a ladder-like trend.

Deutz [43] recently proposed an IPC, also called EIR2, which is however very different from ours, proposed in this article. The main difference is that the R2 indicator is calculated in different spaces. Deutz's R2 indicator is defined in objective space and the expectation of R2 indicators is carried out through multivariate integration. Therefore, the analytic form can only be obtained when  $m = 2$  but does not exist when  $m \geq 3$ , where the Monte Carlo method has to be adopted. In contrast, EIR2 in this paper calculates the R2 indicator in the EI space and hence having the analytic form for any  $m$ . Most importantly, our EIR2 involves no multivariable integration and hence is cheap to evaluate.

**3.4. Overview of the EIR2-EMOA.** Combining the devised EIR2 infilling point criterion with the Kriging model, optimal Latin hypercube sampling method [44], and particle swarm optimization [45], an optimization algorithm for multiobjective problems with expensive functions is proposed, referred to as EIR2-MOEA. The detailed procedure is shown in Algorithm 1.

In the initial stage of EIR2-MOEA, a set of sample points of small size need to be selected for establishing the initial Kriging model. As a design of experiment method, the optimal Latin Hypercube Sampling (LHS) method is utilized as it has flexibility in setting the number of sampling points and can ensure these points filling in the entire design space.

When maximizing the EIR2 value and looking for optimal hyperparameters of the Kriging model, an

optimization method is required. However, the landscapes of these two functions are nonlinear, for which traditional gradient-based optimization algorithms are prone to falling into local minima if the initial point is not selected wisely and has to be rerun multiple times. Particle swarm optimization algorithm, as an intelligent optimization algorithm, does not need information about gradient and has fewer parameter settings, high efficiency of search, and lower complexity, hence is employed to find the optimal parameters.

## 4. Experiment Setup

In order to assess the performance of the proposed algorithm, we applied it to three sets of standard test function suites and compared it with another two algorithms, ParEGO and MaxMin.

**4.1. Test Problems.** Three benchmark function suites, namely, simple set, ZDT set, and DTLZ set, are selected to verify the performance of the IPCs in different aspects, whose definitions are shown in Table 1. The simple set includes three biobjective optimization problems TEST1, TEST2, and FON with the number of variables less than three, whose corresponding PFs are convex, concave, and discontinuous, respectively. The ZDT set contains three biobjective test functions, ZDT1, ZDT2, and ZDT3, with convex, concave, and discontinuous PF, respectively. We set the number of variables to 5 to test IPCs' performance on problems having more decision variables. DTLZ2, DTLZ5, and DTLZ7 form the DTLZ test set, which all have three objectives and five decision variables, and they are selected to assess the performance of the IPCs when the number of objectives is large.

**4.2. Performance Metric.** We hope that the approximated Pareto front found close enough to the true Pareto front and is evenly distributed in the objective space, that is, to meet the requirement on convergence and diversity. For this purpose, we choose three performance metrics to quantify the performance of the approximated Pareto front.

- (1) Inverted Generational Distance (IGD) [28]:

$$\text{IGD}(T, P) = \frac{1}{|T|} \sum_{\mathbf{x} \in T} \min_{\mathbf{p} \in P} \{d(\mathbf{x}, \mathbf{p})\}, \quad (13)$$

where  $T$  and  $P$  are true and approximated Pareto set, respectively,  $|T|$  is the number of solutions in  $T$ , and  $d(\mathbf{x}, \mathbf{p})$  represents the Euclidean distance between  $\mathbf{x}$  and  $\mathbf{p}$  in objective space. IGD measures the average distance between the true and approximated Pareto set. Therefore, the smaller IGD value is preferred.

- (2) HyperVolume (HV) [31]:

$$\text{HV}(P) = \text{Vol} \left( \bigcup_{\mathbf{x} \in P} [f_1(\mathbf{x}), r_1] \times \dots \times [f_m(\mathbf{x}), r_m] \right), \quad (14)$$

where  $\mathbf{r} = [r_1, \dots, r_m]^T$  represents reference point which is dominated by all solutions in  $P$  and Vol

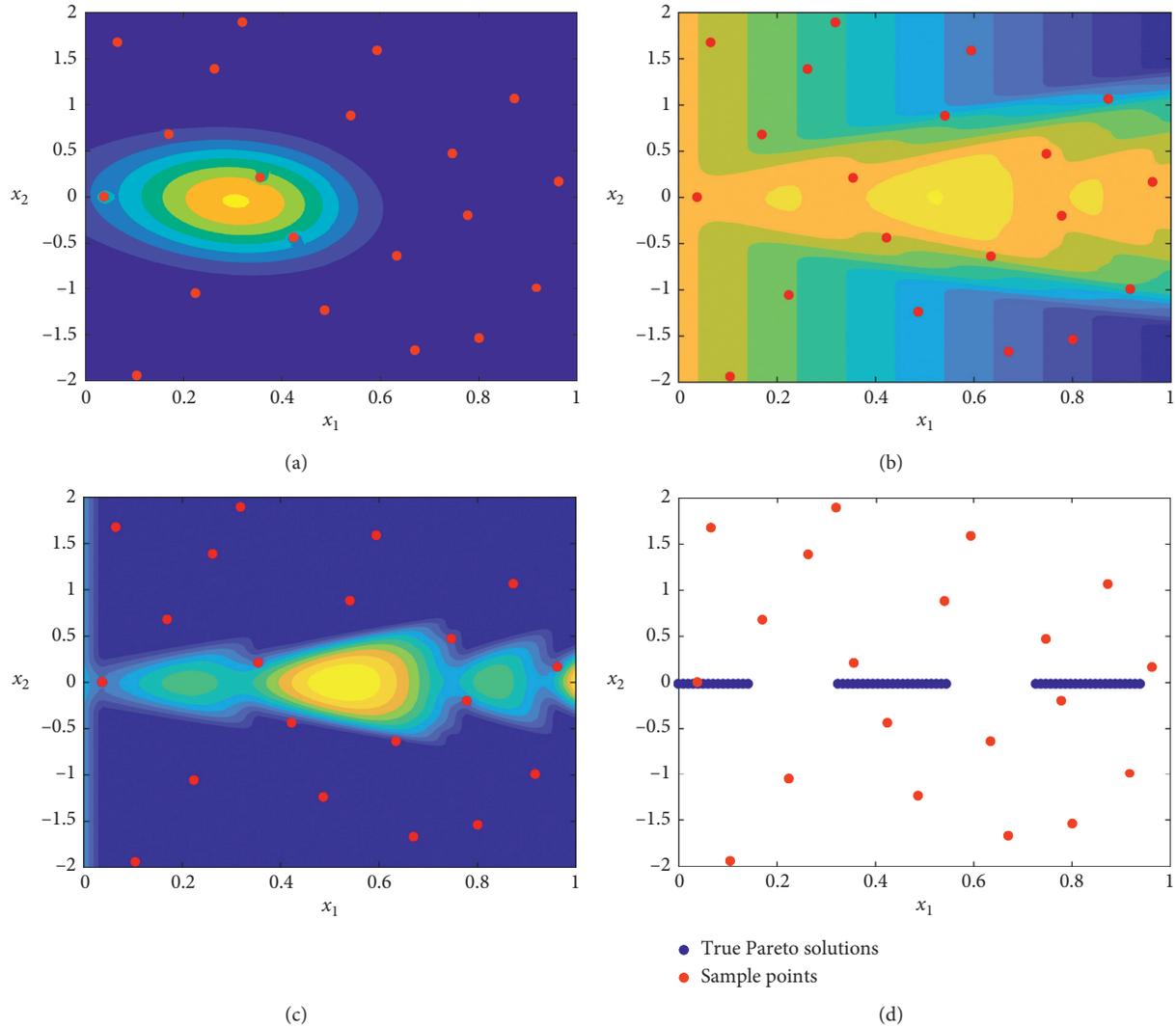


FIGURE 3: The landscape of infilling point criteria in 2D decision space for TEST2 by (a) ParEGO with weight vector  $[0.8, 0.2]$ , (b) MaxMin, and (c) EIR2 with 20 uniformly distributed weight vectors. (d) Both the true Pareto set and 20 sample points generated by the optimal Latin hypercube sampling method.

stands for Lebesgue measure. Geometrically, HV is the volume enclosed by the approximated Pareto set  $P$  and the reference point  $\mathbf{r}$ . HV can reflect both convergence and uniformity. If the approximate Pareto set performances well with respect to both aspects, then the HV value will be high.

(3) Nondominated solution ratio (NR):

$$\text{NR}(P) = \frac{|P|}{N_{\max}}, \quad (15)$$

where  $|P|$  is the number of nondominated solutions and  $N_{\max}$  the maximum number of evaluations by expensive function. NR is used to measure the proportion of the nondominated solutions out of the total sample points evaluated by expensive functions. In other words, NR represents the efficiency of IPC in using computing resources. The larger the NR value, the more nondominated solutions found by the IPC

and the higher the utilization efficiency of computing resources.

**4.3. Parameter Setting.** We hope to evaluate the performance of each IPC with limited computing resources, that is, for each test function we set the maximum number of evaluations involving expensive functions  $N_{\max}$ , shown in Table 2. The size of the initial sample set is specified to be a fraction of  $N_{\max}$ , namely,  $\alpha N_{\max}$ , where  $\alpha$  is a proportional coefficient which is  $1/3$  in this paper. To ensure fairness, all three IPCs start with the same initial sample point set generated by optimal LHS in each run, and ten runs are executed for each test function.

Reference points are required for calculating the HV value. All biobjective optimization problems share the same reference point  $[1.1, 1.1]$  except TEST1 for which  $[20.5, 20.5]$  is used; all three-objective optimization problems are assigned the same reference point  $[1.1, 1.1, 1.1]$  except DTLZ7, where  $[0.95, 0.95, 6.6]$  is utilized.

TABLE 1: The definition of test functions.

Name	Dimension	Boundary	Objective function	Comments on PF
TEST1	2	$x_1 \in [0.4, 1.6]$ $x_2 \in [2, 5]$	$f_1 = (x_1 - 2)^2 + (x_2 - 1)^2$ $f_2 = x_1^2 + (x_2 - 6)^2$	Convex, simple
TEST2	2	$x_1 \in [0, 1]$ $x_2 \in [-2, 2]$	$f_1 = x_1$ $f_2 = 1 + x_2^2 - x_1 - 0.1 \sin(5\pi x_1)$	Convex, disconnected
FON	3	$x_i \in [-4, 4]$	$f_1 = 1 - \exp(-\sum_{i=1}^n (x_i - (1/\sqrt{3}))^2)$ $f_2 = 1 - \exp(-\sum_{i=1}^n (x_i + (1/\sqrt{3}))^2)$	Nonconvex
ZDT1	5	$x_i \in [0, 1]$	$f_1 = x_1$ $f_2 = z(1 - \sqrt{x_1/z})$ $z = 1 + (9/(n-1)) \sum_{i=2}^n x_i$	Convex
ZDT2	5	$x_i \in [0, 1]$	$f_1 = x_1$ $f_2 = z(1 - (x_1/z)^2)$ $z = 1 + (9/(n-1)) \sum_{i=2}^n x_i$	Nonconvex
ZDT3	5	$x_i \in [0, 1]$	$f_1 = x_1$ $f_2 = z(1 - \sqrt{x_1/z} - (x_1/z)\sin(10\pi x_1))$ $z = 1 + (9/(n-1)) \sum_{i=2}^n x_i$	Convex, disconnected
DTLZ2	5	$x_i \in [0, 1]$	$f_1 = (1+g)\cos(\pi x_1/2)\cos(\pi x_2/2)$ $f_2 = (1+g)\cos(\pi x_1/2)\sin(\pi x_2/2)$ $f_3 = (1+g)\sin(\pi x_1/2)$ $g = \sum_{i=\{3,4,5\}} (x_i - 0.5)^2$	Nonconvex
DTLZ5	5	$x_i \in [0, 1]$	$f_1 = (1+g)\cos(\pi\theta_1/2)\cos(\pi\theta_2/2)$ $f_2 = (1+g)\cos(\pi\theta_1/2)\sin(\pi\theta_2/2)$ $f_3 = (1+g)\sin(\pi\theta_1/2)$ $\theta_i = (\pi/(4(1+g)))(1+2gx_i), i = 1, 2$ $g = \sum_{i=\{3,4,5\}} (x_i - 0.5)^2$	Convex
DTLZ7	5	$x_i \in [0, 1]$	$f_1 = x_1$ $f_2 = x_2$ $f_3 = (1+g)h$ $g = 1 + 3\sum_{i=\{3,4,5\}} x_i$ $h = 3 - \sum_{i=1}^2 [(f_i/(1+g))(1 + \sin(3\pi f_i))]$	Disconnected

As for particle swarm optimization, the population size is set to 100, the evolution generation is 100, and the scaling factor is 1.2. Other parameters of MaxMin and ParEGO are taken as the recommended values in original papers.

## 5. Result and Discussion

For each test function, ten independent runs are performed, and the HV, IGD, and NR indicators of the resulting approximated Pareto set are calculated. Finally, their mean and standard deviation of the ten runs are calculated and listed, respectively, in Tables 3–5. For each test function, among the three IPCs, the winning one is shown in boldface with a gray background.

**5.1. Simple Tests.** TEST1, TEST2, and FON are all biobjective test functions but have, respectively, convex, piecewise continuous, and concave Pareto front. The comparison of mean and std value for HV, IGD, and NR metric can be

found at the top part of Tables 3–5, respectively. It is shown that ParEGO achieved the worst performance on all three performance metrics, while EIR2 and MinMax achieve similar results but both significantly better than ParEGO. Specifically, for TEST1 and TEST2, EIR2 is better than MaxMin in terms of HV and IGD but is slightly worse concerning NR. For FON, EIR2 ranked first on all three performance metrics.

Figure 4 plots the approximated Pareto front obtained by the three IPCs on TEST1, TEST2, and FON test function, respectively, with its HV value being median in ten runs. It is evident that the number of nondominated solutions obtained by ParEGO is small and they are not evenly distributed, which is more obvious on TEST2 and FON. This phenomenon can be foreseen because ParEGO lacks a mechanism to guarantee the uniform distribution of the solutions, that is, its weight vector is randomly generated at each iteration. In contrast, the approximated Pareto fronts obtained by EIR2 and MaxMin seem

**Input:**

- $N_{\max}$ : Maximum number of evaluations by expensive function  
 $\alpha$ : Ratio of the number of initial samples to  $T_{\max}$   
 $H$ : Integer used to generate weight vector

**Output:**

- $P$ : Nondominated solution set
- (1) Initialization: Obtain design variable limits  $\mathbf{x}^{\max}$  and  $\mathbf{x}^{\min}$ , number of objective  $m$ , etc. according to MOP to be solved. Set  $S = \emptyset$  and  $P = \emptyset$ .
  - (2) Generate  $\alpha N_{\max}$  sample points in design space  $[\mathbf{x}^{\min}, \mathbf{x}^{\max}]$  using optimal Latin hypercube sampling.
  - (3) **for**  $k = 1$  to  $\alpha N_{\max}$  **do**
  - (4) Calculate the expensive objective function values  $\mathbf{y}^k = \mathbf{F}(\mathbf{x}^k)$  for sample point  $\mathbf{x}^k$
  - (5)  $S = S \cup \{(\mathbf{x}^k, \mathbf{y}^k)\}$ , i.e., add sample point to  $S$
  - (6) **end for**
  - (7) Generate  $c = C_{H+m-1}^{m-1}$  weight vectors in objective space and save them in set  $\Lambda = \{\lambda^1, \dots, \lambda^c\}$ .
  - (8) Find non-dominated solutions in  $S$  and put them into  $P$
  - (9) **for**  $k = 1$  to  $(1 - \alpha)N_{\max}$  **do**
  - (10) **for**  $k = 1$  to  $m$  **do**
  - (11) Using PSO to find the best hyperparameter  $\theta$
  - (12) Build Kriging model of the  $m$ -th objective function based on  $S$
  - (13) **end for**
  - (14) According to EIR2 indicator, apply PSO to find the best infilling point  $\mathbf{x}^{\text{inf}}$
  - (15) Calculate expensive objective function  $\mathbf{y}^{\text{inf}} = \mathbf{F}(\mathbf{x}^{\text{inf}})$
  - (16) Update  $S = S \cup \{(\mathbf{x}^{\text{inf}}, \mathbf{y}^{\text{inf}})\}$
  - (17) Find the non-dominated solutions in  $S$  and put them into  $P$
  - (18) **end for return** Output  $P$  as approximated Pareto set

ALGORITHM 1: EIR2-MOEA.

TABLE 2: Maximum number of evaluations involving expensive functions.

	TEST1	TEST2	FON	ZDT1	ZDT2	ZDT3	DTLZ2	DTLZ5	DTLZ7
$N_{\max}$	45	60	90	90	90	120	210	210	210

TABLE 3: The mean and standard deviation of the HV value of approximated Pareto set found by ParEGO, MaxMin, and EIR2-EMOA on all test functions.

	ParEGO		MaxMin		EIR2	
	Mean	Std	Mean	Std	Mean	Std
TEST1	272.2024	1.71E+00	278.8626	1.03E-01	<b>279.0228</b>	8.64E-02
TEST2	0.6900	8.02E-03	0.7227	1.49E-03	<b>0.7233</b>	7.44E-04
FON	0.4583	3.10E-02	0.5324	2.21E-03	<b>0.5364</b>	3.08E-03
ZDT1	0.8468	5.53E-03	0.8463	5.01E-03	<b>0.8643</b>	1.08E-03
ZDT2	0.5057	2.14E-02	0.5121	4.63E-03	<b>0.5312</b>	1.04E-03
ZDT3	1.2925	8.52E-03	1.2946	4.72E-02	<b>1.3230</b>	1.14E-03
DTLZ2	0.6499	9.69E-03	0.7078	5.60E-03	<b>0.7378</b>	2.85E-03
DTLZ5	0.4098	5.02E-03	0.4151	3.90E-03	<b>0.4358</b>	3.45E-04
DTLZ7	1.5410	2.31E-02	1.6333	2.72E-02	<b>1.6858</b>	3.39E-03

to have completely covered the true Pareto fronts and have good distribution performance, regardless of the nature of the true Pareto front of the test function.

Overall, EIR2 and MaxMin have comparable performance and are much better than ParEGO on simple test functions with a low number of variables.

**5.2. ZDT Tests.** The middle part of Tables 3–5 presents the IGD, HV, and NR indicator comparison on ZDT test suites. Figure 5 illustrates the approximated Pareto fronts obtained by

each IPC on ZDT1~ZDT3 whose HV value being median in ten runs. It is clear that EIR2 is the best in terms of HV, IGD, and NR indicators for all ZDT test functions. Specifically, the HV value and IGD value of EIR2 are much better than that of ParEGO and MaxMin. It is worth noting that the advantages of EIR2 are quite obvious for IGD and NR indicators. For example, for ZDT3 EIR2 gets an NR value of 0.5438, while the other two algorithms only have 0.1924 and 0.3686; meanwhile, it gets an IGD value of 0.0143 while the other two 0.0544 and 0.0591.

According to Figure 5, it is seen that both ParEGO and MaxMin successfully converged to the true Pareto front of

TABLE 4: The mean and standard deviation of the IGD value of approximated Pareto set found by ParEGO, MaxMin, and EIR2-EMOA on all test functions.

	ParEGO		MaxMin		EIR2	
	Mean	Std	Mean	Std	Mean	Std
TEST1	0.4347	4.88E-02	0.1939	5.19E-03	<b>0.1908</b>	4.49E-03
TEST2	0.0415	1.02E-02	<b>0.0097</b>	5.06E-04	0.0097	5.49E-04
FON	0.0640	3.47E-02	0.0121	1.05E-03	<b>0.0109</b>	1.98E-03
ZDT1	0.0211	2.42E-03	0.0232	4.17E-03	<b>0.0096</b>	8.15E-04
ZDT2	0.0285	1.51E-02	0.0249	3.26E-03	<b>0.0126</b>	1.16E-03
ZDT3	0.0591	9.69E-03	0.0544	1.21E-02	<b>0.0143</b>	1.79E-03
DTLZ2	0.0999	5.40E-03	0.0992	6.36E-03	<b>0.0700</b>	4.21E-03
DTLZ5	0.0289	3.01E-03	0.0334	4.89E-03	<b>0.0084</b>	4.72E-04
DTLZ7	0.1393	1.70E-02	0.0851	1.25E-02	<b>0.0576</b>	4.23E-03

TABLE 5: The mean and standard deviation of the NR value of approximated Pareto set found by ParEGO, MaxMin, and EIR2-EMOA on all test functions.

	ParEGO		MaxMin		EIR2	
	Mean	Std	Mean	Std	Mean	Std
TEST1	0.6689	5.78E-02	<b>0.8711</b>	2.52E-02	0.8644	2.86E-02
TEST2	0.2883	2.95E-02	<b>0.6183</b>	2.66E-02	0.6017	2.54E-02
FON	0.1867	4.50E-02	0.4200	3.22E-02	<b>0.4367</b>	7.33E-02
ZDT1	0.2611	1.76E-02	0.2556	3.19E-02	<b>0.4522</b>	3.31E-02
ZDT2	0.2478	6.87E-02	0.2033	2.10E-02	<b>0.4311</b>	3.77E-02
ZDT3	0.1242	2.17E-02	0.1283	1.77E-02	<b>0.2958</b>	2.95E-02
DTLZ2	0.2752	1.81E-02	0.2733	2.49E-02	<b>0.4381</b>	2.00E-02
DTLZ5	0.1652	1.27E-02	0.1262	1.64E-02	<b>0.3529</b>	1.09E-02
DTLZ7	0.1924	2.43E-02	0.3686	6.87E-02	<b>0.5438</b>	2.53E-02

ZDT1 and ZDT2; however, they failed in making them evenly distributed. This phenomenon is even worse on ZDT3 whose true Pareto front consists of five segmented curves. The majority of solutions found by ParEGO and MaxMin are clustered in one segment, while only a few solutions located in other segments. In contrast, the approximated Pareto set obtained by EIR2 are uniformly and equally distributed among all five segments.

In summary, EIR2 is capable of finding more non-dominated solutions than ParEGO and MaxMin using the same amount of computing resources and keeping them evenly distributed along the approximated Pareto front.

**5.3. DTLZ Tests.** The DTLZ test function suites are used aiming at testing an algorithm's ability to three-objective optimization problems. The metric comparisons are listed at the bottom part of Tables 3–5, respectively. Figure 6 shows each IPC's results on DTLZ2, DTLZ5, and DTLZ7 with their HV value being median in ten runs. Same as the ZDT case, on the three test functions, the EIR2 is still significantly ahead of its two competitors in terms of all three performance metrics. To be more specific, take DTLZ2 as an instance, the NR value of EIR2 is 0.3529, which is nearly three times 0.1262 of MaxMin and more than twice

0.1652 of ParEGO, which shows again EIR2 is more efficient in using computation resources compared to other IPCs. The larger HV value and smaller IGD value obtained by EIR2 also indicate its capacity in achieving strong convergence ability.

The same conclusion is also reached by inspecting Figure 6. Specifically, ParEGO has difficulty in converging to the true Pareto front, while MaxMin, despite converged to PF, suffered from being not able to spread its approximated Pareto front uniformly distributed along the true Pareto front which is a 1/8 sphere of radius 1, and there are only a few solutions in the middle part and most of the solutions are found in three edges. In contrast, the approximated PF obtained by EIR2 is adequate to converge to and cover entirely and evenly the 1/8 spheres surface.

## 6. Engineering Application

In addition to boundary constraints, most MOPs also contain equality or inequality constraints. We assume that constraint functions are expensive; hence, the Kriging model need to be established for each constraint. On the purpose of measuring the degree to which the solution satisfies the constraints, probability of feasibility (PoF) [46, 47] is introduced as follows:

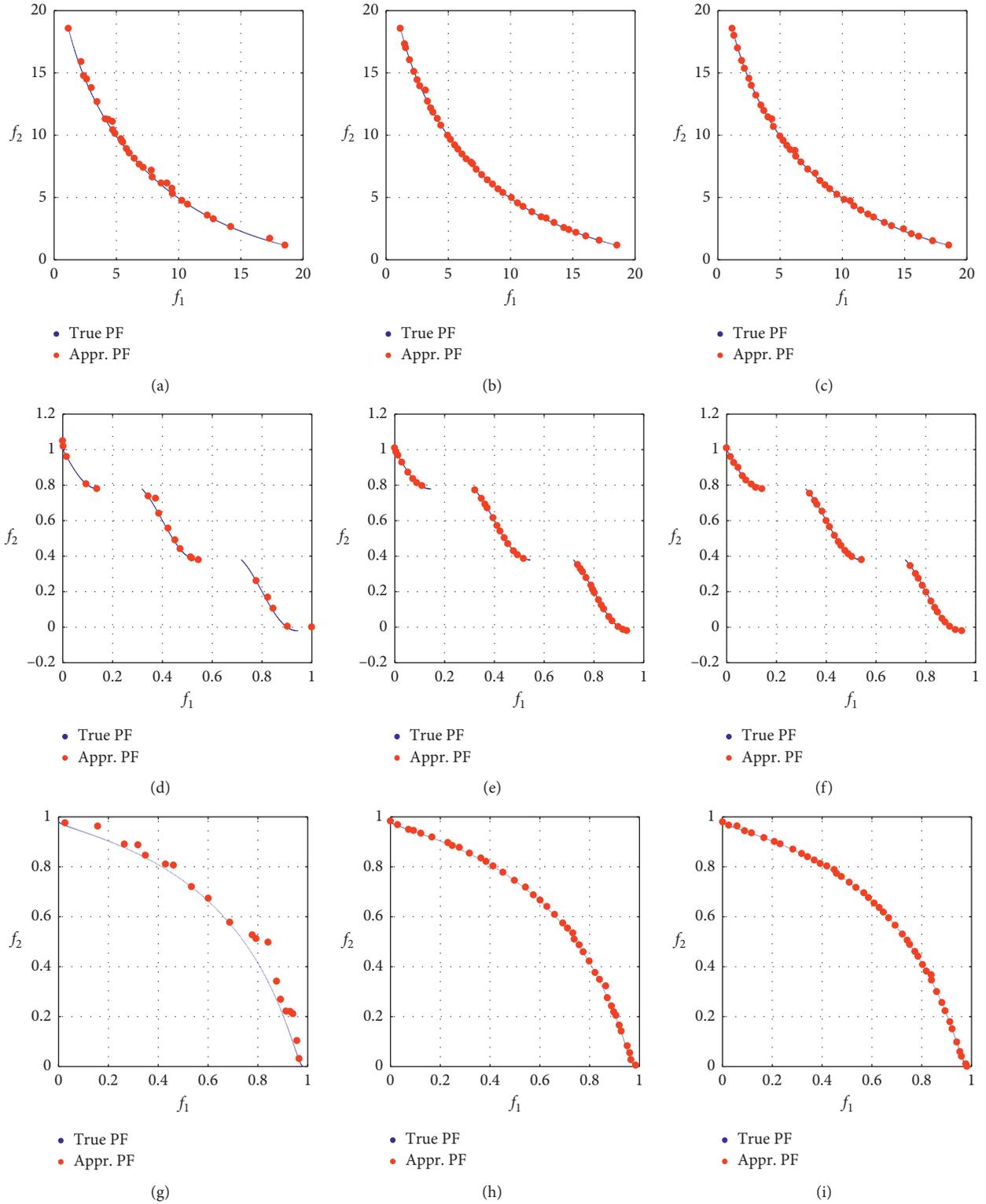


FIGURE 4: Approximated Pareto front found by three infilling criteria ParEGO(left), MaxMin(middle), and EIR2(right) on test function TEST1 (a~c), TEST2 (d~f), and FON (g~i).

$$\text{PoF}_i(\mathbf{x}) = P(g_i(\mathbf{x}) \leq 0) = \Phi\left(\frac{\widehat{g}_i(\mathbf{x})}{\widehat{s}_i(\mathbf{x})}\right), \quad (16)$$

where  $\widehat{g}_i(\mathbf{x})$  and  $\widehat{s}_i(\mathbf{x})$  is the predicted value and standard deviation of constraint function  $g_i(\mathbf{x})$ . Based on PoF, the average probability of feasibility (APoF) is defined as follows:

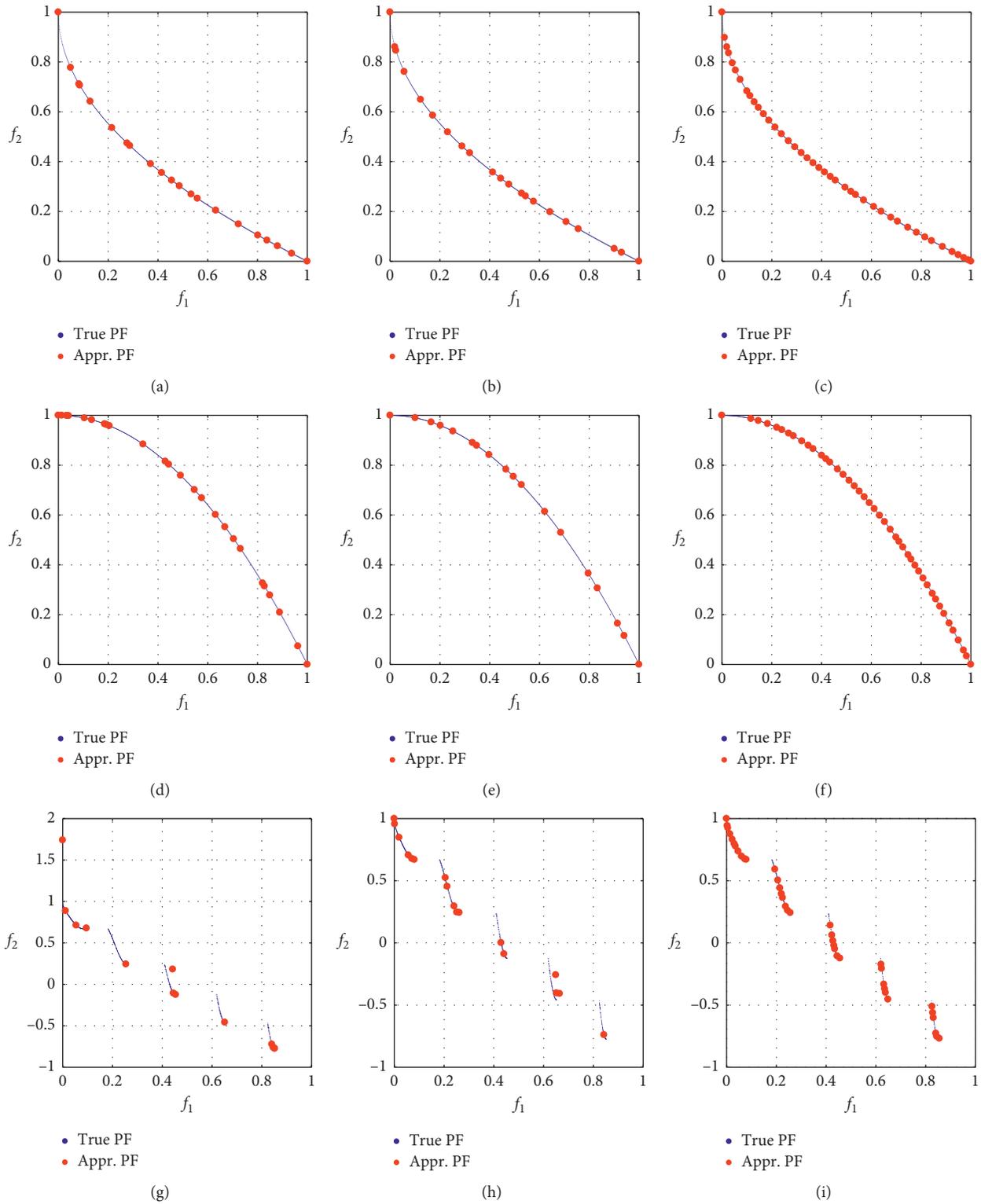


FIGURE 5: Approximated Pareto front found by three infilling criteria ParEGO(left), MaxMin(middle), and EIR2(right) on test function ZDT1 (a~c), ZDT2 (d~f), and ZDT3 (g~i).

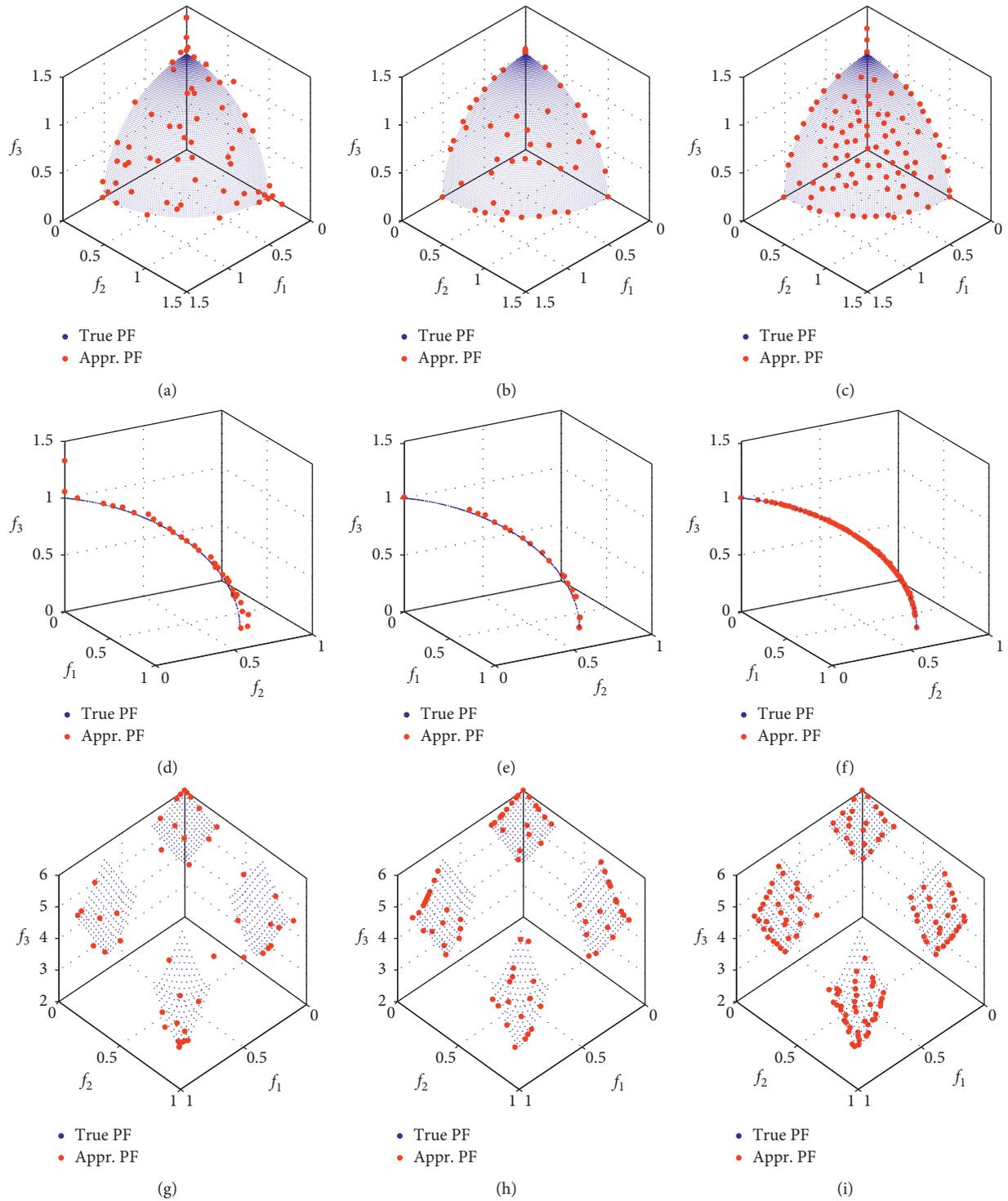


FIGURE 6: Approximated Pareto front found by three infilling criteria ParEGO(left), MaxMin(middle) and EIR2(right) on test function DTLZ1 (a~c), DTLZ3 (d~f), and DTLZ5 (g~i).

$$\text{APoF} = \frac{1}{\nu} \sum_{i=1}^{\nu} \text{PoF}_i(\mathbf{x}), \quad (17)$$

where  $\nu$  is the number of all inequality constraints. The APoF value of an infeasible solution is low, but it will increase as more constraints get satisfied. Clearly, the value of APoF varies in interval  $[0, 1]$ . The closer the value of APoF is to 1, the higher the probability that the solution is feasible, and hence more preferred.

In combination with APoF, EIR2 is extended to constrained MOPs

$$\begin{aligned} \text{CEIR2}(\mathbf{x}) &= \text{EIR2}(\mathbf{x}, P, \Lambda) \times \text{APoF}(\mathbf{x}) \\ &= \frac{1}{|\Lambda|} \sum_{\lambda \in \Lambda} \min_{\mathbf{p} \in P} \left\{ \max_{j=1, \dots, m} \left\{ \lambda_j \times \text{EI}_j(\mathbf{x}, \mathbf{p}) \right\} \right\} \\ &\quad \times \frac{1}{\nu} \sum_{i=1}^{\nu} \Phi \left( \frac{\widehat{g}_i(\mathbf{x})}{\widehat{s}_i(\mathbf{x})} \right). \end{aligned} \quad (18)$$

In this section, the proposed CEIR2 indicator is applied to the optimization design of energy storage flywheel [48] to test its effectiveness.

The energy storage flywheel [49] uses a flywheel in high-speed rotating to store kinetic energy and convert it into electrical energy when needed. Given a constant speed, the maximum energy storage is proportional to the moment of inertia of flywheel. Larger moment of inertia can be obtained by adjusting the geometric parameters of the flywheel, but at the same time on the risk of increasing the total mass. In this paper, our job is to maximize the flywheel rotational inertia as well as minimize total mass.

Figure 7 is a cross-sectional profile of an energy storage flywheel. There are six decision variables:  $r_1 \sim r_4$  and  $t_1$  and  $t_2$ , representing the radial sizes and thickness of the hub and spoke of the flywheel, respectively. Three inequality constraints involve restrictions on radial stress, radial deformation, and hoop stress. The overall optimization problem is expressed as follows:

$$\begin{aligned} \min \quad & \mathbf{F}(\mathbf{x}) = [-I(\mathbf{x}), M(\mathbf{x})] \\ \text{s.t.} \quad & g_1(\mathbf{x}) = \sigma_r^{\max}(\mathbf{x}) - [\sigma_r] \leq 0 \\ & g_2(\mathbf{x}) = \sigma_{\theta}^{\max}(\mathbf{x}) - [\sigma_{\theta}] \leq 0 \\ & g_3(\mathbf{x}) = \Delta_r^{\max}(\mathbf{x}) - [\Delta_r] \leq 0, \end{aligned} \quad (19)$$

where  $\mathbf{x} = [r_1, r_2, r_3, r_4, t_1, t_2]$ , whose upper and lower boundary are listed in Table 6.  $I(\mathbf{x})$  and  $M(\mathbf{x})$  represent the rotational inertia and total mass of the flywheel, respectively.  $\sigma_r^{\max}$ ,  $\sigma_{\theta}^{\max}$ , and  $\Delta_r^{\max}$ , respectively, represent the maximum radial stress, the maximum hoop stress, and the maximum radial deformation of the outer ring when the flywheel rotates, accompanied by their corresponding allowable values,  $[\sigma_r]$ ,  $[\sigma_{\theta}]$ , and  $[\Delta_r]$ .

Although formulas of the objective functions  $I(\mathbf{x})$  and  $M(\mathbf{x})$  can be obtained directly from the geometric parameters, the calculation of the constraint functions has to resort to finite element analysis, considering the complexity of flywheel structure and external force conditions; hence, this problem is a constrained expensive multiobjective optimization problem.

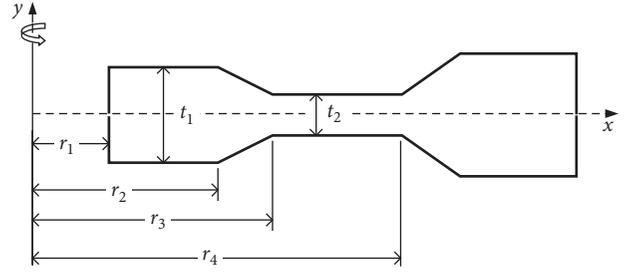


FIGURE 7: Diagram of energy storage flywheel.

TABLE 6: Range of decision variables.

Range (mm)	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$\mathbf{x}^{\max}$	114.22	128.19	159.94	203.12	36.75	8.35
$\mathbf{x}^{\min}$	104.22	118.19	149.94	193.12	26.75	4.35

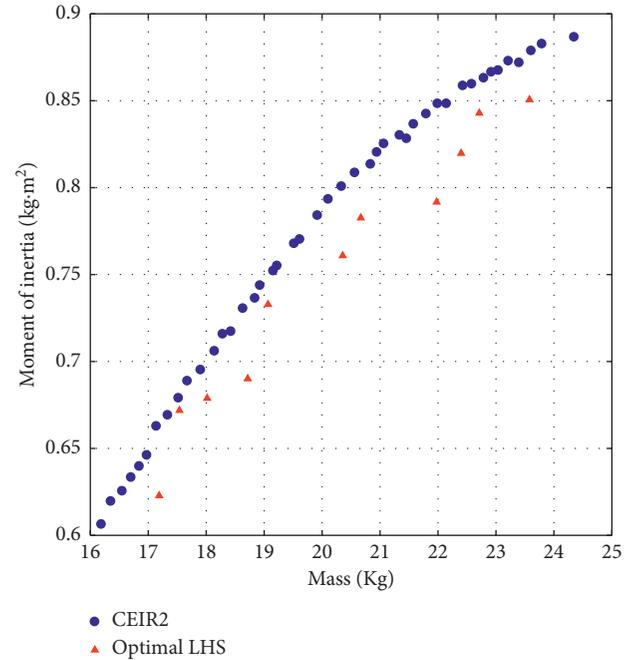


FIGURE 8: Nondominated solutions obtained by CEIR2 and optimal LHS method for the design of the energy storage flywheel.

The material used for the flywheel is Ti-6Al-4V titanium alloy, with corresponding allowable stress  $[\sigma_r] = [\sigma_{\theta}] = 800$  MPa. We set  $[\Delta_r] = 10$  mm at the working rotation speed  $\omega = 20,000$  rpm. ANSYS is employed for calculating values of objective and constraint functions.

As for parameter setting, the maximum number of function evaluations is 360,  $\alpha$  is set to 1/4, and other parameters remain unchanged as for the test function case.

The resulting feasible nondominated solutions are obtained by applying CEIR2 to this engineering problem, as shown in Figure 8. For comparison, we use the optimal LHS method to get 900 sample points at once and then evaluate values of the expensive objective and constraint functions,

out of which the feasible nondominated solutions are also shown in Figure 8.

CEIR2 found a total of 44 feasible nondominated solutions, which have a large distribution range in the objective space, namely,  $I(\mathbf{x}) \sim [16.1, 24.7]$  kg and  $M(\mathbf{x}) \sim [0.68, 0.89]$  kg · m<sup>2</sup>, and uniformly distributed along the approximated PF. In comparison, the optimal LHS spend 900 expensive function evaluations but found only 11 feasible nondominated solutions, which are inferior to that of CEIR2 both in terms of convergence and diversity, indicating CEIR2's advantage in using computing resources.

## 7. Conclusion and Future Work

In this paper, a novel infilling point criterion EIR2 is proposed, which is combined with the Kriging approximate model, optimal Latin hypercube sampling, and PSO algorithm for dealing with expensive multiobjective optimization problems. The basic idea of EIR2 is to map a point in the objective space to a set in the EI space, and the R2 indicator of the set is defined as the fitness of this point. Due to its analytic form, the computation required for this criterion is very low. The proposed algorithm is applied to three suites of standard test functions. The results show that the algorithm can get more nondominated solutions with higher convergence and dispersion under the equal computing resources. Combined with the average probability of feasibility indicator, EIR2 can also be applied to constrained expensive MOPs. In the engineering optimization of energy storage flywheel, the nondominated solutions having a wider range and diversity are obtained exhausting relatively low computation cost.

Future work will have to extend EIR2 to a parallel version that is capable of finding multiple infilling points [50–54] in one iteration with the help of parallel computers to further accelerate the speed of the optimization process. Besides, regarding the parameterization of the utility function, we use a fixed set of uniformly distributed weight vectors. This strategy runs smoothly if the geometry of PF is relatively regular, such as a simplex-like shape, but may struggle [55] if the true PF turns out to be irregular, such as having a sharp peak of a low tail or discontinuous, because multiple weight vectors cloud corresponds to one same solution. As a result, the diversity of the nondominated solution set cannot be guaranteed. A potential promising strategy [56–58] is to dynamically adjust the weight vectors during the optimization process so that the solutions are directed towards each part of the PF. Future work will be incorporating this strategy to our IPC to further improve its ability to maintain diversity.

## Abbreviations

MOPs:	Multiobjective Optimization Problems
MOEA:	Multiobjective Optimization Evolutionary Algorithm
EI:	Expectation Improvement
EGO:	Efficient Global Optimization
IPC:	Infilling Point Criterion

DoE:	Design of Experiments
MCM:	Monte Carlo Method
EIR2:	Expectation Improvement R2 Indicator
IGD:	Inverted Generational Distance
LHS:	Latin hypercube sampling
HV:	Hypervolume
NR:	Nondominated solution ratio
PS:	Pareto set
PF:	Pareto front
PSO:	Particle swarm optimization
PoF:	Probability of Feasibility
APoF:	Average Probability of Feasibility
QI:	Quality indicators.

## Data Availability

The source code of the EIR2-MOEA algorithm proposed in this article can be obtained by contacting the corresponding author.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Authors' Contributions

Both authors have contributed equally to this paper and have read and approved the final manuscript.

## Acknowledgments

This research was supported by Key Projects in the National Science & Technology Pillar Program (2011BAF08B03) and Shanghai's Plan for Absorption and Innovation of Imported Technology (industry-20; direction-28).

## References

- [1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Hoboken, NJ, USA, 2001.
- [2] A. Abraham and L. Jain, "Evolutionary multiobjective optimization," in *Advanced Information and Knowledge Processing*, A. Abraham, L. Jain, and R. Goldberg, Eds., Springer, London, UK, pp. 1–6, 2005.
- [3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: improving the strength pareto evolutionary algorithm," in *Proceedings of the EUROGEN'2001*, Athens, Greece, September 2001.
- [5] Q. Zhang and H. Li, "MOEA/D: a multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712–731, 2007.
- [6] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 Congress on Evolutionary Computation*, IEEE Computer Society, Washington, DC, USA, pp. 1051–1056, 2002.
- [7] Y. S. Ong, P. B. Nair, and A. J. Keane, "Evolutionary optimization of computationally expensive problems via

- surrogate modeling,” *AIAA Journal*, vol. 41, no. 4, pp. 687–696, 2003.
- [8] D. R. Jones, “Taxonomy of global optimization methods based on response surfaces,” *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.
- [9] S. M. Clarke, J. H. Griebisch, and T. W. Simpson, “Analysis of support vector regression for approximation of complex engineering analyses,” *Journal of Mechanical Design*, vol. 127, no. 6, pp. 1077–1087, 2005.
- [10] A. Gaspar-Cunha and A. Vieira, “A multi-objective evolutionary algorithm using neural networks to approximate fitness,” *Evaluations*, vol. 6, p. 20, 2005.
- [11] J. P. C. Kleijnen, “Kriging metamodeling in simulation: a review,” *European Journal of Operational Research*, vol. 192, no. 3, pp. 707–716, 2009.
- [12] J. Antony, *Design of Experiments for Engineers and Scientists*, Elsevier, Amsterdam, Netherlands, 2014.
- [13] J. Parr, C. M. Holden, A. I. Forrester, and A. J. Keane, “Review of efficient surrogate infill sampling criteria with constraint handling,” in *Proceedings of the 2nd International Conference on Engineering Optimization*, pp. 1–10, Lisbon, Portugal, 2010.
- [14] M. T. M. Emmerich, K. Yang, and A. H. Deutz, “Infill criteria for multiobjective bayesian optimization,” in *High-Performance Simulation-Based Optimization*, T. Bartz-Beielstein, B. Filipi, P. Korosec et al., Eds., Springer International Publishing, Cham, Switzerland, pp. 3–16, 2020.
- [15] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, 1998.
- [16] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley & Sons, Hoboken, NJ, USA, 2008.
- [17] F. Viana and R. Haftka, “Surrogate-based optimization with parallel simulations using the probability of improvement,” in *Proceedings of the 13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, American Institute of Aeronautics and Astronautics, Fort Worth, TX, USA, 2010.
- [18] J. Knowles, “ParEGO: a hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 50–66, 2006.
- [19] J. Davins-Valldaura, S. Moussaoui, G. Pita-Gil, and F. Plestan, “ParEGO extensions for multi-objective optimization of expensive evaluation functions,” *Journal of Global Optimization*, vol. 67, no. 1–2, pp. 79–96, 2017.
- [20] J. Shinkyu and O. Shigeru, “Efficient global optimization (EGO) for multi-objective problem and data mining,” in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 3, pp. 2138–2145, Scotland, UK, 2005.
- [21] T. Murata and H. Ishibuchi, “MOGA: multi-objective genetic algorithms,” in *Proceedings of the 1995 IEEE International Conference on Evolutionary Computation*, vol. 1, p. 289, Perth, Western Australia, 1995.
- [22] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, “Expensive multiobjective optimization by MOEA/D with Gaussian process model,” *IEEE Transactions on Evolutionary Computation*, vol. 14, pp. 456–474, 2010.
- [23] J. Svenson and T. Santner, “Multiobjective optimization of expensive-to-evaluate deterministic computer simulator models,” *Computational Statistics & Data Analysis*, vol. 94, pp. 250–264, 2016.
- [24] K. Shimoyama, S. Jeong, and S. Obayashi, “Kriging-surrogate-based optimization considering expected hypervolume improvement in non-constrained many-objective test problems,” in *Proceedings of the 2013 IEEE Congress on Evolutionary Computation*, pp. 658–665, IEEE, Cancun, MX, USA, 2013.
- [25] K. M. Decker, “The Monte Carlo method in science and engineering: theory and application,” *Computer Methods in Applied Mechanics and Engineering*, vol. 89, no. 1–3, pp. 463–483, 1991.
- [26] M. Li and X. Yao, “Quality evaluation of solution sets in multiobjective optimisation: a survey,” *ACM Computing Surveys*, vol. 52, no. 2, pp. 1–38, 2019.
- [27] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, “Reference point specification in inverted generational distance for triangular linear Pareto front,” *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 6, pp. 961–975, 2018.
- [28] M. R. Sierra and C. A. Coello Coello, “Improving PSO-based multi-objective optimization using crowding, mutation and epsilon-dominance. evolutionary multi-criterion optimization,” in *Lecture Notes in Computer Science*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds., Springer, Berlin, Germany, pp. 505–519, 2005.
- [29] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. da Fonseca, “Performance assessment of multiobjective optimizers: an analysis and review,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.
- [30] B. Li, K. Tang, J. Li, and X. Yao, “Stochastic ranking algorithm for many-objective optimization based on multiple indicators,” *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 924–938, 2016.
- [31] J. Bader and E. Zitzler, “HypE: an algorithm for fast hypervolume-based many-objective optimization,” *Evolutionary Computation*, vol. 19, pp. 45–76, 2010.
- [32] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, “How to specify a reference point in hypervolume calculation for fair performance comparison,” *Evolutionary Computation*, vol. 26, no. 3, pp. 411–440, 2018.
- [33] T. Pamulapati, R. Mallipeddi, and P. N. Suganthan, “ $I_{SDE+}$ —an indicator for multi and many-objective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, pp. 346–352, 2019.
- [34] R. Wang, S. Chen, L. Ma, S. Cheng, and Y. Shi, “Multi-indicator bacterial foraging algorithm with kriging model for many-objective optimization,” in *Advances in Swarm Intelligence*, vol. 10941, pp. 530–539, Springer International Publishing, Cham, Switzerland, 2018.
- [35] D. H. Phan and J. Suzuki, “R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1836–1845, Cancun, MX, USA, 2013.
- [36] R. H. Gómez and C. A. C. Coello, “MOMBI: a new meta-heuristic for many-objective optimization based on the R2 indicator,” in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 2488–2495, Cancun, MX, USA, 2013.
- [37] D. Brockhoff, T. Wagner, and H. Trautmann, “On the properties of the R2 indicator,” in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, pp. 465–472, Association for Computing Machinery, Philadelphia, PA, USA, 2012.
- [38] A. Sobester, S. J. Leary, and A. J. Keane, “On the design of optimization strategies based on global response surface approximation models,” *Journal of Global Optimization*, vol. 33, no. 1, pp. 31–59, 2005.

- [39] I. Pan, M. Babaei, A. Korre, and S. Durucan, "Artificial neural network based surrogate modelling for multi-objective optimisation of geological CO<sub>2</sub> storage operations," *Energy Procedia*, vol. 63, pp. 3483–3491, 2014.
- [40] J. M. Parr, A. I. J. Forrester, A. J. Keane, and C. M. E. Holden, "Enhancing intill sampling criteria for surrogate-based constrained optimization," *Journal of Computational Methods in Sciences and Engineering*, vol. 12, no. 1–2, pp. 25–45, 2012.
- [41] F. Li, J. Liu, S. Tan, and X. Yu, "R2-MOPSO: a multi-objective particle swarm optimizer based on R2-indicator and decomposition," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 3148–3155, Sendai, Japan, May 2015.
- [42] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, pp. 645–652, Association for Computing Machinery, Vancouver, BC, Canada, 2014.
- [43] A. Deutz, M. Emmerich, and K. Yang, *The Expected R2-Indicator Improvement for Multi-Objective Bayesian Optimization. Evolutionary Multi-Criterion Optimization*, K. Deb, E. Goodman, C. A. Coello Coello et al., Eds., Springer International Publishing, Cham, Switzerland, pp. 359–370, 2019.
- [44] J.-S. Park, "Optimal Latin-hypercube designs for computer experiments," *Journal of Statistical Planning and Inference*, vol. 39, no. 1, pp. 95–111, 1994.
- [45] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, 1995.
- [46] S. u. Rehman and M. Langelaar, "Expected improvement based infill sampling for global robust optimization of constrained problems," *Optimization and Engineering*, vol. 18, pp. 723–753, 2017.
- [47] Z. Han, F. Liu, C. Xu, K. Zhang, and Q. Zhang, "Efficient multi-objective evolutionary algorithm for constrained global optimization of expensive functions," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, pp. 2026–2033, Wellington, New Zealand, 2019.
- [48] L. Jiang, W. Zhang, G. J. Ma, and C. W. Wu, "Shape optimization of energy storage flywheel rotor," *Structural and Multidisciplinary Optimization*, vol. 55, no. 2, pp. 739–750, 2017.
- [49] S. C. Mohan and D. K. Maiti, "Structural optimization of rotating disk using response surface equation and genetic algorithm," *International Journal for Computational Methods in Engineering Science and Mechanics*, vol. 14, no. 2, pp. 124–132, 2013.
- [50] G. Sun, Y. Tian, R. Wang, J. Fang, and Q. Li, "Parallelized multiobjective efficient global optimization algorithm and its applications," *Structural and Multidisciplinary Optimization*, vol. 61, no. 2, pp. 763–786, 2020.
- [51] D. Han and J. R. Zheng, "Solving expensive multi-objective optimization problems by kriging model with multi-point updating strategy," *Applied Mechanics and Materials*, vol. 685, pp. 667–670, 2014.
- [52] K. Yang, P. S. Palar, M. Emmerich, K. Shimoyama, and T. Bäck, "A multi-point mechanism of expected hypervolume improvement for parallel multi-objective bayesian global optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 656–663, Association for Computing Machinery, Prague, Czech Republic, 2019.
- [53] Y. He, J. Sun, P. Song, X. Wang, and A. S. Usmani, "Preference-driven Kriging-based multiobjective optimization method with a novel multipoint infill criterion and application to airfoil shape design," *Aerospace Science and Technology*, vol. 96, Article ID 105555, 2020.
- [54] J. Liu, W.-P. Song, Z.-H. Han, and Y. Zhang, "Efficient aerodynamic shape optimization of transonic wings using a parallel infilling strategy and surrogate models," *Structural and Multidisciplinary Optimization*, vol. 55, no. 3, pp. 925–943, 2017.
- [55] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2017.
- [56] S. Xu, H. Chen, X. Liang, and M. He, "A modified MOEAD with an adaptive weight adjustment strategy," in *Proceedings of the International Conference on Intelligent Computing, Automation and Systems (ICICAS)*, pp. 184–188, Chongqing, China, 2019.
- [57] C. Dai, X. Lei, and X. He, "A decomposition-based evolutionary algorithm with adaptive weight adjustment for many-objective problems," *Soft Computing*, 2019.
- [58] M. Li and X. Yao, "What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multiobjective optimisation," *Evolutionary Computation*, vol. 28, no. 2, pp. 227–253, 2020.