# A Surrogate-Assisted Offspring Generation Method for Expensive Multi-objective Optimization Problems

Fan Li[1], Liang Gao[1], Weiming Shen[1], Xiwen Cai[1]; Shifeng Huang[2]
[1]State Key Laboratory of Digital Manufacturing Equipment and Technology, School of Mechanical Science and Engineering, Huazhong University of Science &Technology, Wuhan 430074, PR China
[2]National CNC Engineering Technology Research Center, School of Mechanical Science and Engineering, Huazhong University of Science & Technology, Wuhan, 430074, PR China
Email: D201780171@hust.edu.cn; gaoliang@mail.hust.edu.cn; wshen@ieee.org; 864533724@qq.com; d201677154@hust.edu.cn

***Abstract*—Surrogate-assisted multi-objective evolutionary algorithms have been commonly used to solve multi-objective expensive problems. In this paper, we investigate whether the surrogate-assisted offspring generation method can improve the optimization efficiency of multi-objective evolutionary algorithms. We first construct a surrogate model for each objective function. After that, some candidate solutions from the surrogate models are used to produce promising offspring for the multi-objective evolutionary algorithm. In addition, a pre-screening criterion based on reference vectors and the non-dominated rank is used to select the surviving offspring and exactly evaluated individuals. The pre-screening criterion can ensure the diversity and convergence of the offspring, and reduce function evaluations. Benchmark problems with their dimensions varying from 8 to 30 are used to test the effects of the surrogate-assisted offspring generation method under the framework of using the pre-screening criterion. Experimental results show that using the candidate solutions from surrogate models can enhance the performance of its basic algorithm on most of the problems.**

***Keywords—Expensive problem, multi-objective evolutionary algorithm, offspring generation, surrogate model.***

## I. INTRODUCTION

Many real-world optimization problems need to optimize multiple conflicting objectives simultaneously. This means that the improvement of one objective may lead to the deterioration of other objectives. Many evolutionary algorithms have been developed to solve these problems. Commonly used multi-objective evolutionary algorithms (MOEAs) can be roughly classified into three categories: dominance-based, decomposition-based, and indicator-based. Dominance-based MOEAs, such as non-dominated sorting genetic algorithm II (NSGA-II) [1] and the improved strength Pareto evolutionary algorithm [2], use the dominance relationship to guide the evolutionary process such as the selection of parent solutions and the calculation of fitness values. Decomposition-based MOEAs decompose a multi-objective optimization problem into multiple single-objective sub-problems [3]. Commonly used decomposition approaches include the weighted sum approach, the Tchebycheff approach, and the boundary intersection approach. Indicator-based MOEAs infuse some estimation indicators such as hypervolume [4] and ε–indicator [4] into the search process. Although the performance of these MOEAs is satisfactory, they usually require a large number of function evaluations (FEs) to find acceptable non-dominated solutions.

Many real-world multi-objective optimization problems usually involve time-consuming simulations or expensive experiments. One real function evaluation of these problems may consume plenty of time. These expensive multi-objective optimization problems commonly exist in the engineering field such as airfoil shape design and car cab design [5]. Many FEs will be consumed when MOEAs are applied to solve these problems. Surrogate models have been used to assist the search of MOEAs to enhance optimization efficiency. Commonly used surrogate models include Kriging models [6], radial basis function (RBF) models [7], artificial neural networks [8], polynomial regression models [9], and support vector machines [10].

Surrogate models are usually used to approximate the objective functions of expensive problems. The fitness value and uncertainty of an un-evaluated point can be predicted by the surrogate models. In surrogate-assisted decomposition-based MOEAs [3], multi-objective problems are transformed into multiple single-objective problems. Infill criteria such as expected improvement and lower confidence bound used in single-objective problems can also be used in these algorithms.

In surrogate-assisted non-decomposition-based MOEAs, the predicted information from the surrogate models can be used to replace some exact FEs or select promising individuals for exact FEs. For example, in [11], a criterion was used to decide whether the simulation or its Kriging models should be used for evaluating a design point. In [12], two criteria were used to select individuals for exact FEs. The first criterion based on angle penalized distance was used to promote convergence. The second criterion based on the average of standard deviations obtained from Kriging models was used to promote diversity. Some well-distributed reference vectors were used to classify the population into several clusters. A solution with the best value of the above criterion was selected from each cluster, and the variation of empty reference vectors was used to switch between the two criteria. In some studies, each pair of parent solutions produced multiple trial positions. The solutions with good non-dominated rank were selected for exact FEs [13, 14]. For example, each pair of parents produced λ candidate offspring in [15]. Non-dominated solutions were selected from these offspring. Then, the non-dominated offspring dominating most solutions in current non-dominated parents were selected for exact FEs.

In addition to the above methods to utilize surrogate models, the population which evolved on a coarse surrogate model for some iterations was used to produce offspring for a fine surrogate-assisted MOEAs in [16]. The dimension of the coarse surrogate model is a subset of the dimension of the

This work was supported by the National Natural Science Foundation for Distinguished Young Scholars of China [Grant Number 51825502], the 111 Project [Grant Number B16019] and Program for HUST Academic Frontier Youth Team [Grant Number 2017QYTD04].
Corresponding Author: Weiming Shen.

original problem. The algorithm is designed to solve off-line data-driven optimization problems. Inspired by this algorithm, we investigate whether the candidate points from surrogate models can help to produce promising offspring for MOEAs in this paper. We first build a surrogate model for each objective function. Some candidate solutions from the surrogate models are combined with the population in the multi-objective evolutionary algorithm to produce offspring. A pre-screening criterion based on reference vectors and non-dominated ranks is used to select the surviving offspring. The reference vectors can ensure the diversity of the surviving offspring, while the non-dominated rank can ensure the convergence of the surviving offspring. In addition, the non-dominated rank is based on the predicted fitness values from the surrogate models. Therefore, it will not add extra FEs. The pre-screening criterion is also used to select some individuals from the surviving offspring for exact FEs. Some well-distributed reference vectors are used to select parent solutions for the next generation in NSGA-III [17]. This method is beneficial for obtaining well distributed non-dominated solutions. Therefore, NSGA-III is used as the basic algorithm for the proposed algorithm. The main contributions of this paper are as follows.

1) A surrogate-assisted offspring generation method is proposed to accelerate the search progress of MOEAs. The method can help MOEAs find good solutions with fewer FEs.
2) Different from the current methods of use reference vectors in the pre-screening criterion, the non-dominated ranks and the reference vectors are simultaneously used in the proposed pre-screening criterion. The well-distributed reference vectors can ensure the diversity of the selected individuals, while the non-dominated ranks can ensure the convergence.

The rest of this paper is organized as follows. Section II briefly introduces the knowledge about NSGA-III and the used surrogate model. Section III describes the proposed multi-objective evolutionary algorithm with the surrogate-assisted offspring generation method (SAO-MOEA). Section IV conducts the comparative experiments on the benchmark problems to test the efficiency of the SAO-MOEA. Section V concludes the paper.

## II. Preliminaries

### A. NSGA-III

The framework of NSGA-III [17] is shown in Algorithm 1. The main framework of NSGA-III is similar to NSGA-II except that it uses a set of well-spread reference points to maintain the diversity of the population. We will present the main components of NSGA-III in this section. Before the optimization starts, the parent solutions are initialized. Recombination and mutation strategies are conducted on parent solutions to produce offspring solutions. Then, some strategies are used to select parent solutions from current parent and offspring solutions ($R_t$) for the next iteration. The selection of parent solutions is the main component of NSGA-III.

The non-dominated-sorting method is first conducted on $R_t$. Members from the non-dominated level 1 to $l$ are put into $S_t$. The members of $S_t$ are set as $P_{t+1}$ if $|S_t|=N$. Otherwise, members from 1 to ($l$ - 1) fronts are selected for $P_{t+1}$ and the remaining ($K = N - |P_{t+1}|$) population members are chosen from the last front $F_l$. The objective function values of the population are normalized, and a set of well- spread reference points are set. The members of $S_t$ are associated with a reference point by calculating the perpendicular distance of them to the reference points. The number of members associated with each reference point is also obtained in this way. $K$ members are gradually selected from $F_l$ to construct $P_{t+1}$ by using the niche-preservation operation.

**Algorithm 1** NSGA-III
**Input**: $Kr$ (the number of reference solutions), $N$ (population size), $P$ (initial parent solutions), $T_{\max}$ (maximum iteration number)
**Output**: final population, non-dominated solutions

$t$=1
**While** $t < T_{\max}$
  $S_t = \Phi$; $i$=1
  $Q_t$=Recombination + Mutation ($P_t$)
  $R_t = P_t \cup Q_t$
  ($F_1$, $F_2$, …)=non-dominated-sort($R_t$)
  **Repeat**
    $S_t = S_t \cup F_i$ and $i=i+1$
  **Until** $|S_t|>=N$
  Last front to be included: $F_l=F_i$
  **If** $|S_t|=N$ **then**
    $P_{t+1}=S_t$, **break**
  **Else**
    $P_{t+1}=\cup_{j=1}^{l-1} F_j$
    Points to be chosen from $F_l$: $K=N-|P_{t+1}|$
    Normalize objectives and create the reference set $Zr$:
    Associate each member $\boldsymbol{s}$ of $S_t$ with a reference point
    Compute niche count of reference point
    Choose $K$ members one at a time from $F_l$ to $P_{t+1}$
  **End if**
**End while**

### B. RBF models

The RBF model uses a weighted sum of basic functions to approximate complicated landscapes [7, 18]. For a data set consisting of the values of the input variables and response values at $N$ training points, the true function $y(x)$ can be approximated as

$$\hat{y}(x) = \sum_{i=1}^{N} \lambda_i \varphi(\|x - c_i\|) + p(x) \tag{1}$$

where $\lambda$ are coefficients calculated by solving linear equations; $c_i$ denotes the $i$th center of basis functions; $p$ is either a polynomial model or a constant value, and a linear polynomial is used in this paper [13, 19]; $\varphi$ is a basis function.

As (1) is underdetermined, the orthogonality condition is further imposed on coefficients $\lambda$ as

$$\sum_{i=1}^{N} \lambda_i p_j(x_i) = 0, \quad \text{for } j = 1, 2, ......, m \tag{2}$$

$$\begin{bmatrix} \Phi & P \\ P^{\mathrm{T}} & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ b \end{bmatrix} = \begin{bmatrix} y \\ 0 \end{bmatrix} \tag{3}$$

where $m$ is the number of terms of $p(x)$; $\Phi_{ij} = \varphi(\|x_i - x_j\|)$, ($i$ =1,2,..., $N$), ($j$ = 1,2,..., $N$); $P_{ij} = p_j(x_i)$, ($i$ = 1,2,..., $N$), ($j$ = 1,2,..., $m$); $\lambda = [\lambda_1, \lambda_2, ..., \lambda_N]^{\mathrm{T}}$, and $b = [b_1, b_2, ..., b_m]^{\mathrm{T}}$. Equation (3) consists of ($N + m$) equations and its solution gives coefficients $\lambda$ and $b$ in (1).

The RBF model can approximate nonlinear and high-dimensional problems. It has shown good performance on some surrogate-assisted evolutionary algorithms [13, 20]. Therefore, the model is used in this paper.

## III. PROPOSED MULTI-OBJECTIVE EVOLUTIONARY ALGORITHM WITH SURROGATE-ASSISTED OFFSPRING GENERATION METHOD

A generic diagram of the proposed algorithm is presented in Fig. 1. Before the optimization starts, Latin Hypercube Sampling (LHS) is used to produce initial samples. These samples are evaluated by exact FEs. All the evaluated samples are saved in the database (DB). The initial population is randomly selected from the DB. Then, the RBF surrogate model is constructed for each objective function with all the samples. Some candidate points are obtained from the surrogate models. Next, crossover and mutation strategies are conducted on these candidate points and current parent solutions to produce offspring solutions. A pre-screening criterion based on reference vectors and the non-dominated rank is used to select surviving offspring. The pre-screening criterion is also used to select some offspring for exact evaluations. These exactly evaluated solutions are saved into the DB. After that, the environmental selection method is conducted on current parent and offspring solutions to select parent solutions for the next iteration. The process repeats until the termination condition is met.

The method to obtain candidate points from the surrogate models is introduced in Section III. A. The pre-screening criterion based on reference vectors and the non-dominated rank is described in Section III.B.

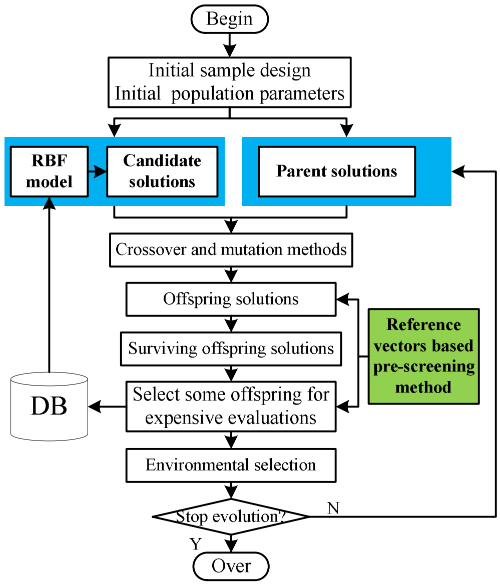


Fig. 1. The framework of the proposed multi-objective evolutionary algorithm with the surrogate-assisted offspring generation method

### A. Method to Obtain Candidate Points

Many infill criteria based on surrogate models can be used to select the candidate points. These infill criteria can be roughly classified into three categories. The performance-based criterion selects solutions with good fitness values. The non-dominated solutions are usually used in multi-objective optimization. The uncertainty-based criterion usually selects solutions with great uncertainties. Predicted variances from Kriging models, predicted errors from multiple different models, and the distance to the evaluated samples can be used to estimate uncertainty. The last criterion considers the fitness values and uncertainties simultaneously. Commonly used criteria include multi-objective EI criteria (Euclidean distance improvement, maximin distance improvement, and hypervolume improvement).

Using the performance-based criterion to select candidate points is beneficial for exploiting current promising areas and accelerate the convergence rate. However, the population may be trapped into the local optima of the surrogate models as the surrogate models are usually not accurate. Using the uncertainty-based criterion to select candidate points are beneficial for exploring sparse areas. However, the convergence rate may be delayed. Using the criterion combining both fitness values and uncertainty to select candidate points considers the exploitation and exploration simultaneously.

In this paper, we consider exploration and exploitation simultaneously. We use a multi-objective evolutionary algorithm to evolve on the current surrogate models for some iterations (*maxiter*). The current parent solutions are set as candidate points as shown in Fig. 2. The obtained candidate points may have great uncertainties if *maxiter* is small, while the candidate points may have good fitness values if *maxiter* is big. Note that any multi-objective evolutionary algorithm can be used in this method. We also use NSGA-III to simplify the algorithm.

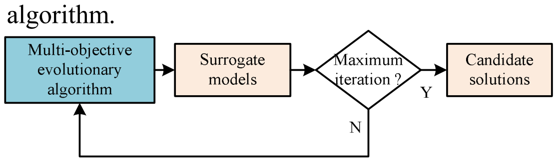


Fig. 2. Method to obtain candidate points

### B. The proposed pre-screening criterion

The proposed pre-screening criterion based on reference vectors and the non-dominated rank is used to select surviving offspring. A set of uniformly distributed reference vectors is first produced. The number of surviving offspring is the same as the population size, so the number of the reference vectors is also set the same as the population size ($N$). Then, associate each offspring to its nearest reference vector in the reference vectors**.** The method of finding the nearest reference vector is similar to the method in NSGA-III. We calculate the perpendicular distances of each offspring to all the reference vectors. Then, each offspring is associated with the reference vector with the smallest value of the distance. For each reference vector, we find non-dominated solutions from its associated solutions ($MO_1$). If the number of $MO_1$ is bigger than one, we find the solution with maximin distance to samples in DB from $MO_1$. This solution is selected as the surviving offspring. The $MO_1$ is directly selected as the surviving offspring if only one solution exists in $MO_1$. All the surviving offspring are selected in this way. Fig. 3 shows solutions in a two-dimension objective space. Nine reference vectors are uniformly distributed in the objective space. The yellow points are evaluated samples in DB, and other points are the produced offspring. The green points are the surviving offspring. The pre-screening criterion compares the dominance relationships of offspring in a local area. Then, select the non-dominated individual far away from the evaluated samples. The diversity of the offspring can be obtained by the reference vectors and the distance criterion, while the convergence can be obtained by the non-dominance relationship.

The pre-screening criterion is also used to select offspring for exact evaluations. However, the number of reference vectors is the same as the number of exactly evaluated individuals ($N_E$). A distance criterion is also used to decide whether the selected individuals will be exactly evaluated. The individual will be exactly evaluated if the minimum Euclidean distance between it and samples in DB is bigger than the threshold $\eta$. The distance is calculated in the variable space.

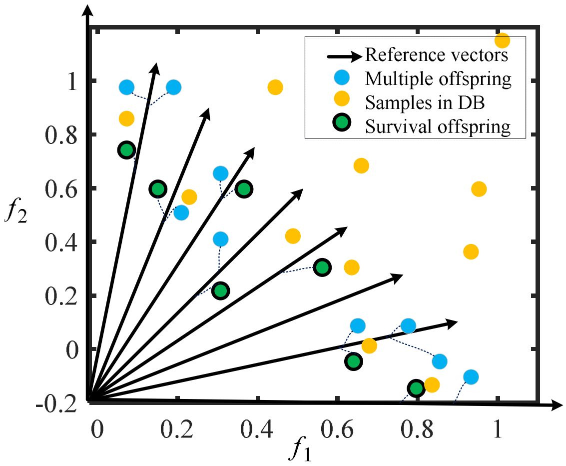


Fig. 3. The proposed pre-screening criterion for selecting surviving offspring

## IV. EXPERIMENTAL STUDY

In this section, empirical studies are conducted to test the performance of the multi-objective evolutionary algorithm with the surrogate-assisted offspring generation method (SAO-MOEA). ZDT instances [21] with two objective functions and DTLZ instances [22] with three objective functions are used. The decision variables are set as 8, 20 and 30 for ZDT instances, and 10, 20 and 30 for DTLZ instances.

### *A. Experimental setup*

NSGA-III is taken as the basic search method. The population size is 20. Simulated binary crossover (SBX) and the polynomial mutation in [23] are used to produce offspring. The number of reference vectors is 30/105 for two/three objective problems. For SAO-MOEA, the population size is 80/106 for two/three objective problems. The number of the initial samples is set as the same as the population size. The *maxiter* is set at 20, and the number of exactly evaluated individuals ($N_E$) is set at 5. The effects of $maxiter/N_E$ on the performance of SAO-MOEA are discussed in Section IV.C. $\eta$ is min(sqrt(0.0012$D$), 5.0e-4$D$×min(ub-lb)) to avoid samples being too close, where $D$ is the dimension of the problem. The other parameters are the same as NSGA-III.

Each algorithm runs 20 times for each test instance. The termination criterion for each algorithm is a predefined maximum function evaluations (*maxNFEs*). *maxNFEs* is 200/300 for two/three objective problems. Inverted generational distance (IGD) [24, 25] is used to evaluate the performance of the compared algorithms. The IGD metric can indicate diversity and convergence of the obtained non-dominated solutions. The reference point sets for IGD metrics are obtained from PlatEMO [23]. The number of the reference points is 500/950 for two/three objective problems.

### *B. Comparative experiments on benchmark problems*

In this section, we compare SAO-MOEA with its variance SAO-MOEA-WM to investigate the effects of the surrogate-assisted offspring method. SAO-MOEA-WM is the same as SAO-MOEA except that it uses the method in NSGA-III to produce offspring. We also compare SAO-MOEA with NSGA-III to investigate the effects of the surrogate models.

#### *1) Experimental results on ZDT instances*

The results obtained by the three algorithms over 20 independent runs are collected in TABLE I. Wilcoxon rank-sum test calculated at a significance level of $\alpha$ =0.05 is also used to test the algorithms, where'≈' indicates that there is no statistically significant difference between the results obtained by SAO-MOEA and the compared algorithm, '+' indicates that SAO-MOEA is significantly better than the compared algorithm, while '−' means that SAO-MOEA is significantly outperformed by the compared algorithms.

SAO-MOEA and SAO-MOEA-WM obtain the best mean values on 12 and 3 problems respectively. SAO-MOEA obtains significantly better results than SAO-MOEA-WM on 12 problems, while it is outperformed by SAO-MOEA-WM on 8-, 20-, and 30-dimension ZDT4. These results indicate that using the surrogate-assisted offspring method can efficiently improve the searching performance of the algorithm on most of the problems. ZDT4 is a multi-modal problem with many local Pareto solutions. This problem is hard to be approximated, so the candidates from surrogate models may not provide real promising evolutionary information. Using the candidates from surrogate models can only accelerate the search towards promising areas in the surrogate models. Therefore, using the surrogate-assisted offspring method may mislead the population to the Pareto solutions of the surrogate models.

SAO-MOEA obtains significantly better results than NSGA-III on all 15 problems, while they perform comparably on the remaining four problems. These results indicate that using the information from the surrogate models can obtain better solutions within the same computational resource. In addition, SAO-MOEA-WM also achieves better results than NSGA-III on all 15 problems. SAO-MOEA-WM is the same as NSGA-III except that it uses the pre-screening criterion to select offspring for exact evaluations. The better results obtained by SAO-MOEA-WM indicate that the pre-screening criterion can significantly reduce unnecessary function evaluations.

In summary, using the proposed pre-screening criterion can significantly improve the performance of the basic algorithm under the same computational resource. In addition, using the surrogate-assisted offspring method can further enhance the search efficiency under the framework of using the pre-screening criterion.

#### *2) Experimental results on DTLZ instances*

The results obtained by the three algorithms on DTLZ instances over 20 independent runs are collected in TABLE II. SAO-MOEA and NSGA-III obtain the best mean values on 12 and 9 problems respectively.

SAO-MOEA obtains significantly better results than SAO-MOEA-WM on 12 out of 21 problems, while it is outperformed by SAO-MOEA-WM on six problems. SAO-MOEA obtains significantly better results than NSGA-III on 12 out of 21 problems, while it is outperformed by NSGA-III on the remaining nine problems. SAO-MOEA-WM also obtains better results than NSGA-III on 12 out of 21 problems, while it obtains worse results than NSGA-III on the remaining nine problems. These results indicate that using the candidates from the surrogate models and the pre-screening criterion can produce positive effects for the algorithm on most of the problems. However, the algorithms using surrogate models cannot produce better results than NSGA-III on DTLZ1, DTLZ3, and DTLZ4. DTLZ1 and DTLZ3 have many local Pareto-optimal fronts, while DTLZ4 has a biased density of solutions at some regions in the search space. The good performance of SAO-MOEA and SAO-MOEA-WM on DTLZ2, DTLZ5, DTLZ6, and DTLZ7 indicates that using the pre-screening criterion and surrogate-assisted offspring method is beneficial for obtaining better solutions within the same computational resource. However, if the problems are too complex, the surrogate models cannot provide promising information. Using the information from the surrogate models may mislead the population to the optima of the surrogate models for the three problems.

In summary, using the candidates from surrogate models can hardly enhance the performance of NSGA-III on the three complex problems (DTLZ1, DTLZ3, and DTLZ4). However, the two strategies can efficiently improve the performance of the NSGA-III on most of the simple problems.

### C. Discussion

#### 1) Effects of the parameter *maxiter*

The population evolves in the surrogate models for *maxiter* iterations. Then, the current population information is used to produce offspring for the SAO-MOEA. In this section, we investigate the effects of *maxiter* on the performance of SAO-MOEA. *maxiter* is set at the value of 0, 5, 10, 20, 50 and 100 respectively, while *maxiter*=0 means no candidates are used. A higher value of *maxiter* is not tested as it will significantly increase the computational time. ZDT instances with $D$=8, 20 and 30 are used to test the algorithms. The results including mean values and standard deviations of IGD values on ZDT instances are presented in TABLE III. Better results can be obtained when *maxiter* is set as a bigger value for most problems except ZDT4. The obtained candidate points will be closer to the Pareto front of the approximate problems when *maxiter* is taken as a bigger value. A bigger value will accelerate the search to more promising areas, so the algorithm with a bigger value of *maxiter* can obtain better results in the same computational budget. However, the algorithm with no candidate solutions from surrogate models being used is better for ZDT4. This indicates that using the candidate solutions from surrogate models can only accelerate the search towards the promising areas in the surrogate models. If the surrogate models cannot provide promising information, using the candidates may mislead the search.

#### 2) Effects of the parameter $N_E$

To reduce function evaluations, only $N_E$ individuals are selected for exact evaluations at each iteration. In this section, we investigate the effects of the parameter $N_E$ on the performance of the proposed SAO-MOEA. $N_E$ is set at the value of 2, 5, 10 and 20 respectively. ZDT instances with $D$=8, 20 and 30 are used to test the algorithms. The obtained results including the mean values and standard deviations of IGD values on ZDT instances are presented in TABLE IV. SAO-MOEA with $N_E$=2 obtains the best mean values in 12 out of 15 problems. SAO-MOEA with $N_E$=20 obtains the worst mean values in 11 out of 15 problems. For most of the ZDT1, ZDT2, ZDT3 and ZDT6 problems, the obtained mean values by the SAO-MOEA gradually get worse with the increase of the value of $N_E$. The accuracies of the surrogate models are relatively low in the early iterations. Evaluating too many individuals at each iteration may leave less FEs for exploitation, so SAO-MOEA with $N_E$=2 performs the best on most of the problems. Overall, the value of 2 is recommended for $N_E$.

## V. Conclusion

In this paper, we use NSGA-III as the basic algorithm to investigate whether the candidate points from surrogate models can produce more promising offspring solutions for NSGA-III. A multi-objective algorithm evolves on the surrogate models for some iterations. Then, the current population from the multi-objective algorithm is taken as the candidate points. Crossover and mutation strategies are conducted on all candidate points and current parent solutions of NSGA-III to produce offspring solutions. To reduce the number of function evaluations, a pre-screening criterion based on reference vectors and the non-dominated rank is used to select the surviving offspring and exactly evaluated offspring. ZDT and DTLZ instances are used to test the proposed algorithm and NSGA-III.

Experimental results show that using the candidate solutions from surrogate models and the proposed pre-screening criterion can enhance the performance of NSGA-III on most of the problems. In addition, the obtained candidate points closer to the Pareto front of the approximate problems will produce more positive effects as it can accelerate the search to more promising areas. However, for some complex problems, using the candidates from the surrogate models may produce negative effects for NSGA-III. It may mislead the search towards the wrong promising areas in the surrogate models.

TABLE I. STATISTICAL RESULTS OF IGD VALUES ON ZDT PROBLEMS

| Problem | $D$ | SAO-MOEA | | SAO-MOEA-WM | | NSGA-III | |
|---|---|---|---|---|---|---|---|
| | | MEAN | STD | MEAN | STD | MEAN | STD |
| ZDT1 | 8 | **0.0052** | 0.0006 | 0.2979(+) | 0.0972 | 4.0341(+) | 1.4283 |
| | 20 | **0.0302** | 0.0248 | 8.5035(+) | 1.6351 | 33.4872(+) | 4.4495 |
| | 30 | **0.1661** | 0.0909 | 24.0398(+) | 4.4482 | 59.5749(+) | 6.1505 |
| ZDT2 | 8 | **0.0074** | 0.0011 | 0.9060(+) | 0.2266 | 6.4941(+) | 1.5522 |
| | 20 | **0.6978** | 0.0696 | 11.1590(+) | 1.6793 | 37.2719(+) | 3.3262 |
| | 30 | **1.4893** | 0.5764 | 23.6781(+) | 3.2754 | 65.7386(+) | 6.0366 |
| ZDT3 | 8 | **0.1544** | 0.1593 | 0.4988(+) | 0.2548 | 4.9030(+) | 1.8920 |
| | 20 | **0.2163** | 0.1339 | 8.6659(+) | 1.7980 | 33.0964(+) | 5.1226 |
| | 30 | **0.3437** | 0.2087 | 23.4543(+) | 3.6421 | 62.1631(+) | 5.9274 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ZDT4 | 8 | 18.3336 | 3.6936 | **10.2689**(-) | 4.0224 | 45.2509(+) | 7.0734 |
| | 20 | 98.6197 | 11.9300 | **72.5124**(-) | 11.1418 | 204.3676(+) | 13.6271 |
| | 30 | 159.1358 | 21.2396 | **122.9268**(-) | 16.1278 | 368.6394(+) | 20.0478 |
| ZDT6 | 8 | **0.6459** | 0.2702 | 5.0900(+) | 0.7997 | 8.6720(+) | 0.6422 |
| | 20 | **3.4909** | 1.1985 | 10.7448(+) | 0.4380 | 13.2636(+) | 0.3756 |
| | 30 | **6.2738** | 1.3999 | 12.8830(+) | 0.4983 | 15.4037(+) | 0.4893 |
| +/≈/- | | N/A | | 12/0/3 | | 15/0/0 | |

TABLE II. STATISTICAL RESULTS OF IGD VALUES ON DTLZ PROBLEMS

| Problem | *D* | SAO-MOEA | | SAO-MOEA-WM | | NSGA-III | |
|---|---|---|---|---|---|---|---|
| | | MEAN | STD | MEAN | STD | MEAN | STD |
| DTLZ1 | 10 | 84.1012 | 17.5563 | 86.3628(≈) | 17.2910 | **57.5261**(-) | 10.8629 |
| | 20 | 320.0955 | 21.9744 | 287.0331(-) | 46.7537 | **232.4035**(-) | 22.7655 |
| | 30 | 635.1275 | 52.0513 | 601.1073(-) | 42.9266 | **507.7794**(-) | 61.1570 |
| DTLZ2 | 10 | **0.0685** | 0.0013 | 0.0741(+) | 0.0028 | 0.2246(+) | 0.0364 |
| | 20 | **0.1002** | 0.0050 | 0.1621(+) | 0.0151 | 0.5225(+) | 0.0559 |
| | 30 | **0.1847** | 0.0228 | 0.4520(+) | 0.1019 | 1.0441(+) | 0.1045 |
| DTLZ3 | 10 | 202.3472 | 33.3737 | 243.9448(+) | 46.7760 | **155.5265**(-) | 29.6164 |
| | 20 | 851.1609 | 129.5639 | 826.9687(≈) | 96.1780 | **659.9093**(-) | 66.8736 |
| | 30 | 1670.9452 | 137.0149 | 1566.1248(-) | 159.3108 | **1300.8422**(-) | 185.9474 |
| DTLZ4 | 10 | 0.3517 | 0.0345 | 0.2984(-) | 0.0472 | **0.2786**(-) | 0.0656 |
| | 20 | 0.7054 | 0.1169 | 0.6159(-) | 0.0624 | **0.5686**(-) | 0.0639 |
| | 30 | 1.6443 | 0.2041 | 1.2897(-) | 0.2410 | **1.0751**(-) | 0.1736 |
| DTLZ5 | 10 | **0.0203** | 0.0021 | 0.0210(≈) | 0.0026 | 0.1486(+) | 0.0346 |
| | 20 | **0.0440** | 0.0052 | 0.0755(+) | 0.0121 | 0.4362(+) | 0.0740 |
| | 30 | **0.0950** | 0.0120 | 0.2622(+) | 0.0707 | 0.9158(+) | 0.1199 |
| DTLZ6 | 10 | **1.5840** | 0.4040 | 2.6775(+) | 0.6240 | 3.4928(+) | 0.5476 |
| | 20 | **8.5764** | 0.8537 | 12.1853(+) | 0.7419 | 11.3020(+) | 1.3797 |
| | 30 | **18.0696** | 1.1442 | 21.4369(+) | 0.9741 | 20.2673(+) | 0.8970 |
| DTLZ7 | 10 | **0.0902** | 0.0066 | 0.3249(+) | 0.2152 | 1.7268(+) | 0.8585 |
| | 20 | **0.1337** | 0.0206 | 0.3994(+) | 0.1045 | 4.1341(+) | 0.7830 |
| | 30 | **0.2201** | 0.0520 | 0.6759(+) | 0.1139 | 4.1449(+) | 0.8547 |
| +/≈/- | | N/A | | 12\3\6 | | 12\0\9 | |

TABLE III. STATISTICAL RESULTS OF IGD VALUES OF ALGORITHMS WITH DIFFERENT VALUES OF THE MAXIMUM ITERATION

| problem | D/maxiter | 0 | | 5 | | 20 | | 50 | | 100 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | MEAN | STD | MEAN | STD | MEAN | STD | MEAN | STD | MEAN | STD |
| ZDT1 | 8 | 0.2979 | 0.0972 | 0.0598(+) | 0.0675 | 0.0052(+) | 0.0006 | **0.0045**(+) | 0.0003 | **0.0045**(+) | 0.0002 |
| | 20 | 8.5035 | 1.6351 | 1.6726(+) | 0.7674 | 0.0302(+) | 0.0248 | 0.0069(+) | 0.0013 | **0.0059**(+) | 0.0008 |
| | 30 | 24.0398 | 4.4482 | 9.4333(+) | 3.4113 | 0.1661(+) | 0.0909 | 0.0082(+) | 0.0015 | **0.0062**(+) | 0.0009 |
| ZDT2 | 8 | 0.9060 | 0.2266 | 0.4238(+) | 0.1285 | 0.0074(+) | 0.0011 | 0.0056(+) | 0.0007 | **0.0052**(+) | 0.0005 |
| | 20 | 11.1590 | 1.6793 | 5.1689(+) | 1.7923 | 0.6978(+) | 0.0696 | 0.1135(+) | 0.1837 | **0.0143**(+) | 0.0086 |
| | 30 | 23.6781 | 3.2754 | 15.1198(+) | 2.9683 | 1.4893(+) | 0.5764 | 0.3878(+) | 0.2459 | **0.0844**(+) | 0.1821 |
| ZDT3 | 8 | 0.4988 | 0.2548 | 0.1765(+) | 0.1386 | 0.1544(+) | 0.1593 | 0.1225(+) | 0.0938 | **0.1070**(+) | 0.0857 |
| | 20 | 8.6659 | 1.7980 | 1.8435(+) | 1.0182 | 0.2163(+) | 0.1339 | 0.1143(+) | 0.1184 | **0.1116**(+) | 0.0778 |
| | 30 | 23.4543 | 3.6421 | 9.7708(+) | 2.5860 | 0.3437(+) | 0.2087 | **0.1320**(+) | 0.0845 | 0.1392(+) | 0.1263 |
| ZDT4 | 8 | **10.2689** | 4.0224 | 10.5938(≈) | 4.3817 | 18.3336(-) | 3.6936 | 18.6601(-) | 4.0013 | 17.4950(-) | 4.8695 |
| | 20 | **72.5124** | 11.1418 | 75.3970(≈) | 12.1665 | 98.6197(-) | 11.9300 | 99.8635(-) | 12.1428 | 96.5022(-) | 10.2963 |
| | 30 | **122.9268** | 16.1278 | 123.7458(≈) | 19.7316 | 159.1358(-) | 21.2396 | 167.0485(-) | 16.0474 | 170.2750(-) | 10.5344 |
| ZDT6 | 8 | 5.0900 | 0.7997 | 1.6224(+) | 0.4845 | 0.6459(+) | 0.2702 | 0.5372(+) | 0.2489 | **0.3878**(+) | 0.3040 |
| | 20 | 10.7448 | 0.4380 | 8.3276(+) | 1.0160 | 3.4909(+) | 1.1985 | **3.3089**(+) | 1.4150 | 3.7704(+) | 1.2450 |
| | 30 | 12.8830 | 0.4983 | 10.5763(+) | 0.7499 | 6.2738(+) | 1.3999 | 5.3952(+) | 1.2467 | **5.1819**(+) | 1.0594 |
| +/≈/- | | N/A | | 12\3\0 | | 12\0\3 | | 12\0\3 | | 12\0\3 | |

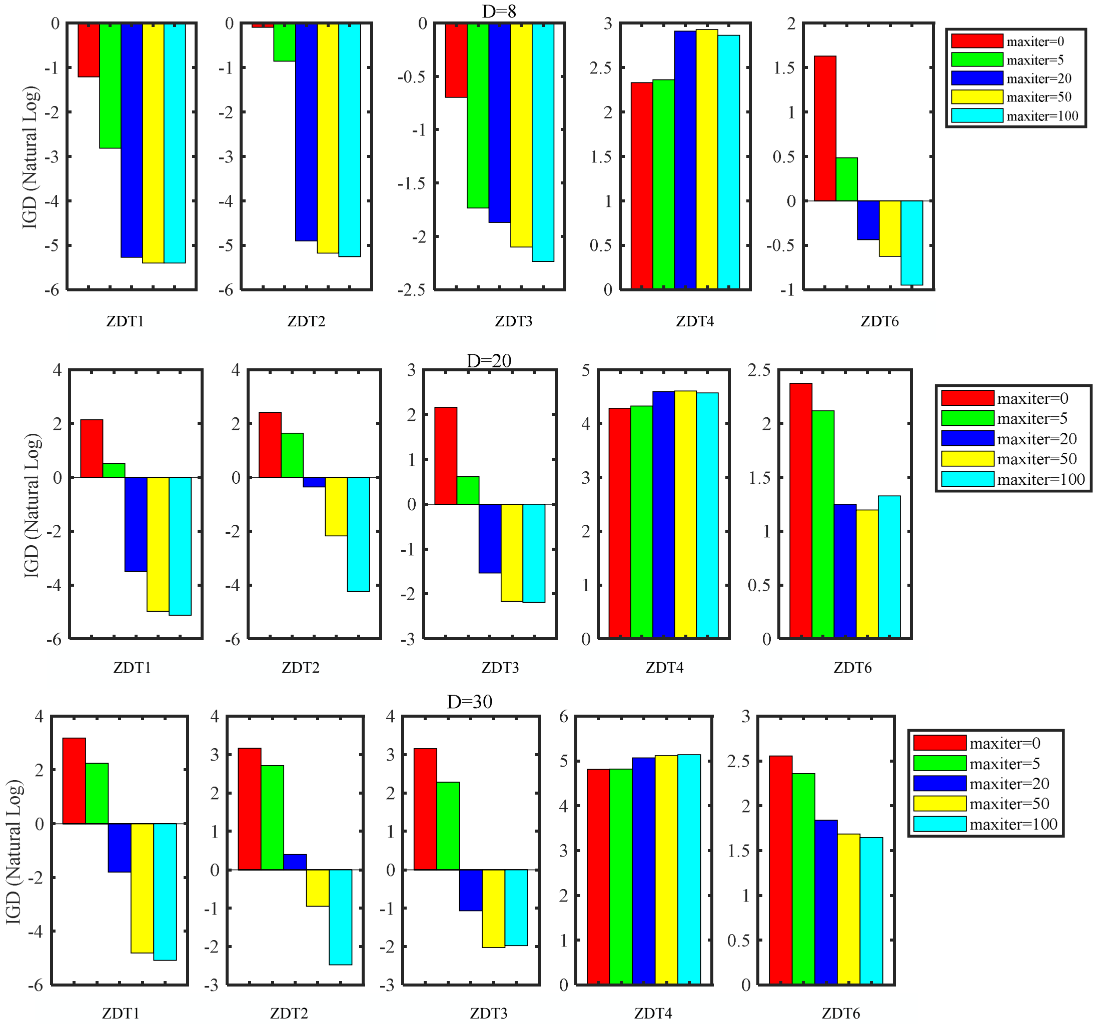


Fig. 4. Mean values of IGD values on ZDT problems for SAO-MOEA with different *maxiter*

TABLE IV. STATISTICAL RESULTS OF IGD VALUES OF ALGORITHMS WITH DIFFERENT NUMBER OF EXACT EVALUATED INDIVIDUALS

| problem | D/ $N_E$ | 2 | | 5 | | 10 | | 20 | |
|---|---|---|---|---|---|---|---|---|---|
| | | MEAN | STD | MEAN | STD | MEAN | STD | MEAN | STD |
| ZDT1 | 8 | **0.0046** | 0.0003 | 0.0052 | 0.0006 | 0.0084 | 0.0016 | 0.0462 | 0.0155 |
| | 20 | **0.0064** | 0.0004 | 0.0302 | 0.0248 | 0.3041 | 0.1853 | 2.1309 | 1.7116 |
| | 30 | **0.0091** | 0.0013 | 0.1661 | 0.0909 | 1.3101 | 0.6888 | 9.5897 | 4.8537 |
| ZDT2 | 8 | **0.0052** | 0.0004 | 0.0074 | 0.0011 | 0.1473 | 0.1477 | 0.3139 | 0.2264 |
| | 20 | **0.0347** | 0.0457 | 0.6978 | 0.0696 | 2.6317 | 1.1896 | 8.5548 | 4.3980 |
| | 30 | **0.5885** | 0.0832 | 1.4893 | 0.5764 | 6.3395 | 2.4469 | 15.7699 | 5.1169 |
| ZDT3 | 8 | **0.1286** | 0.0983 | 0.1544 | 0.1593 | 0.1489 | 0.1440 | 0.1513 | 0.1013 |
| | 20 | **0.1832** | 0.1128 | 0.2163 | 0.1339 | 0.5399 | 0.2163 | 2.5249 | 1.9294 |
| | 30 | **0.1552** | 0.0774 | 0.3437 | 0.2087 | 1.8244 | 1.5434 | 6.0287 | 2.4885 |
| ZDT4 | 8 | 21.9696 | 0.6427 | 18.3336 | 3.6936 | **18.0736** | 4.3444 | 17.5122 | 4.4852 |
| | 20 | 100.9556 | 13.6015 | 98.6197 | 11.9300 | **92.8270** | 10.7113 | 96.6937 | 14.0679 |
| | 30 | 166.9070 | 13.3733 | **159.1358** | 21.2396 | 163.7079 | 12.4904 | 165.8066 | 11.6172 |
| ZDT6 | 8 | **0.2289** | 0.0932 | 0.6459 | 0.2702 | 0.9552 | 0.2909 | 1.5661 | 0.7167 |
| | 20 | **1.7816** | 0.6021 | 3.4909 | 1.1985 | 6.6053 | 1.2889 | 8.6168 | 1.0578 |
| | 30 | **2.4917** | 0.7985 | 6.2738 | 1.3999 | 9.2227 | 1.4029 | 10.6802 | 1.0377 |

## REFERENCES

[1] 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, Apr, 2002.

[2] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," TIK-report, vol. 103, 2001.

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

[4] E. Zitzler, and S. Kunzli, "Indicator-based selection in multiobjective search," Parallel Problem Solving From Nature - Ppsn Viii, vol. 3242, pp. 832-842, 2004.

[5] R. Allmendinger, M. T. M. Emmerich, J. Hakanen, Y. Jin, and E. Rigoni, "Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case," Journal of Multi-Criteria Decision Analysis, vol. 24, no. 1-2, pp. 5-24, 2017.

[6] 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, Dec, 1998.
[7] R. L. Hardy, “Multiquadric equations of topography and other irregular surfaces,” Journal of Geophysical Research, vol. 76, no. 8, pp. 1905-1915, 1971.
[8] S.-C. Horng, and S.-Y. Lin, “Evolutionary algorithm assisted by surrogate model in the framework of ordinal optimization and optimal computing budget allocation,” Information Sciences, vol. 233, pp. 214-229, 2013.
[9] T. Goel, R. T. Hafkta, and W. Shyy, “Comparing error estimation measures for polynomial and kriging approximation of noise-free functions,” Structural and Multidisciplinary Optimization, vol. 38, no. 5, pp. 429-442, 2008.
[10] S. M. Clarke, J. H. Griebsch, 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.
[11] M. Li, G. Li, and S. Azarm, “A Kriging Metamodel Assisted Multi-Objective Genetic Algorithm for Design Optimization,” Journal of Mechanical Design, vol. 130, no. 3, pp. 031401, 2008.
[12] T. Chugh, Y. C. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, “A Surrogate-Assisted Reference Vector Guided Evolutionary Algorithm for Computationally Expensive Many-Objective Optimization,” IEEE Transactions on Evolutionary Computation, vol. 22, no. 1, pp. 129-142, Feb, 2018.
[13] R. G. Regis, “Particle swarm with radial basis function surrogates for expensive black-box optimization,” Journal of Computational Science, vol. 5, no. 1, pp. 12-23, 2014.
[14] R. G. Regis, “Surrogate-Assisted Particle Swarm with Local Search for Expensive Constrained Optimization,” in International Conference on Bioinspired Methods and Their Applications, Paris, France, 2018, pp. 246-257.
[15] A. Syberfeldt, H. Grimm, A. Ng, and R. I. John, “A Parallel Surrogate-Assisted Multi-Objective Evolutionary Algorithm for Computationally Expensive Optimization Problems,” in 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 2008, pp. 3177-3184.
[16] C. Yang, J. Ding, Y. Jin, and T. Chai, “Off-line Data-driven Multi-objective Optimization: Knowledge Transfer between Surrogates and Generation of Final Solutions,” IEEE Transactions on Evolutionary Computation ( Early Access), 2019.
[17] K. Deb, and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints,” IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 577-601, 2013.
[18] A. I. J. Forrester, and A. J. Keane, “Recent advances in surrogate-based optimization,” Progress in Aerospace Sciences, vol. 45, no. 1-3, pp. 50-79, 2009.
[19] R. G. Regis, and C. A. Shoemaker, “Combining radial basis function surrogates and dynamic coordinate search in high-dimensional expensive black-box optimization,” Engineering Optimization, vol. 45, no. 5, pp. 529-555, 2013.
[20] H. Yu, Y. Tan, J. Zeng, C. Sun, and Y. Jin, “Surrogate-assisted hierarchical particle swarm optimization,” Information Sciences, vol. 454-455, pp. 59-72, 2018.
[21] K. Deb, “Multi-objective genetic algorithms: Problem difficulties and construction of test problems,” Evolutionary Computation, vol. 7, no. 3, pp. 205-230, 1999.
[22] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multi-objective optimization test problems,” in Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600), Honolulu, HI, USA, USA, 2002, pp. 825-830.
[23] T. Ye, C. Ran, X. Zhang, and Y. Jin, “PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum],” IEEE Computational Intelligence Magazine, vol. 12, no. 4, pp. 73-87, 2017.
[24] P. Czyżżak, and A. Jaszkiewicz, “Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization,” Journal of Multi-Criteria Decision Analysis, vol. 7, no. 1, pp. 34-47, 1998.
[25] Z. Chen, Y. Zhou, and X. He, “Handling expensive multi-objective optimization problems with a cluster-based neighborhood regression model,” Applied Soft Computing, vol. 80, pp. 211-225, 2019.