Preprint:

Automatic Algorithm Selection for Evolutionary Multi-Objective Optimization

Ye Tian, Shichen Peng, Tobias Rodemann, Xingyi Zhang, Yaochu Jin

2020

This is an accepted article published in 2019 IEEE Symposium Series on Computational Intelligence (SSCI). The final authenticated version is available online at: https://doi.org/10.1109/SSCI44817.2019.9003018

Automated Selection of Evolutionary Multi-objective Optimization Algorithms

Ye Tian[1], Shichen Peng[2], Tobias Rodemann[3], Xingyi Zhang[2], and Yaochu Jin[4]
[1]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China
[2]Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China *(Corresponding author: Xingyi Zhang)*
[3]Honda Research Institute Europe, Carl-Legien-Strasse 30, 63073 Offenbach/Main, Germany
[4]Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom
Email: field910921@gmail.com; sevenrus.peng@gmail.com; tobias.rodemann@honda-ri.de; xyzhanghust@gmail.com; yaochu.jin@surrey.ac.uk

***Abstract*—In the last two decades, many evolutionary algorithms have shown promising performance in solving a variety of multi-objective optimization problems (MOPs). Since there does not exist an evolutionary algorithm having the best performance on all the MOPs, it is unreasonable to use a single evolutionary algorithm to tackle all the MOPs. Since many real-world MOPs are computationally expensive, selecting the best evolutionary algorithm from multiple candidates via empirical comparisons is also impractical. To address the above issues, this paper proposes an automated algorithm selection method for choosing the most suitable evolutionary algorithm for a given MOP. The proposed method establishes a predictor based on the performance of a set of candidate evolutionary algorithms on multiple benchmark MOPs, where the inputs of the predictor are the explicit and implicit features of an MOP, and the output is the index of the evolutionary algorithm having the best performance on the MOP. Experimental results indicate that the evolutionary algorithm suggested by the proposed method is highly competitive among all the candidate evolutionary algorithms, demonstrating the practical value of the proposed method for engineers to select an evolutionary algorithm for their applications.**

I. Introduction

There exist many real-world optimization problems containing multiple conflicting objectives [1], [2], [3], which are known as multi-objective optimization problems (MOPs) [4]. Since an MOP contains multiple optimal solutions rather than a single optimal solution, the goal of solving an MOP is to find a set of well-converged solutions with good diversity to approximate all the optimal solutions of the MOP. Due to the difficulties of MOPs including black-box functions, multi-modality landscapes, and computationally expensive objectives, MOPs are challenging to be solved by numerical optimization approaches. On the contrary, evolutionary computation has become the most effective technique for tackling MOPs [5], since it has good properties that can overcome the difficulties of MOPs. For example, most evolutionary algorithms solve MOPs in a black-box manner, and they can escape from local optimum by optimizing a set of solutions (i.e., a population) simultaneously. Moreover, the surrogate models can be adopted in evolutionary algorithms to handle MOPs with computationally expensive objectives [6].

Since the vector evaluated genetic algorithm (VEGA) [7] was presented in 1985, a large number of multi-objective evolutionary algorithms (MOEAs) have been proposed and demonstrated high effectiveness in solving MOPs. The first generation of MOEAs is characterized by Pareto ranking selection and fitness sharing mechanism, such as MOGA [8], NSGA [9], and NPGA [10]. From 1999 to 2002, some elitism strategy based MOEAs like NSGA-II [11], SPEA2 [12], and PESA-II [13] were developed, where some of them are still very popular in the community. In 2007, the decomposition based algorithm MOEA/D [14] was proposed, which became one of the most effective techniques in MOEAs. After 2010, MOEAs manifest an explosive growth in the literature [15]. In terms of the operators for generating offsprings, existing MOEAs can be roughly grouped into genetic algorithms [11], particle swarm optimization algorithms [16], differential evolution algorithms [17], estimation of distribution algorithms [18], and so on. While in terms of selection strategies, most existing MOEAs can be classified into dominance relation based algorithms [12], decomposition based algorithms [14], and indicator based algorithms [19].

In spite of hundreds (or even thousands) of MOEAs proposed in the literature, according to the no free lunch theorem [20], there does not exist an MOEA outperforming any others on all the MOPs. In other words, it is unreasonable to use a single MOEA to solve all the MOPs in real-world applications. Although many researchers are interested in comparing the performance of state-of-the-art MOEAs [21], [22], it is quite difficult to answer which MOEA is most suitable for an unknown MOP. The difficulty is mainly attributed to the diverse performance of existing MOEAs, as they employed different operators and selection strategies that exhibit significantly different performance. For example, the Pareto dominance relation is effective for MOPs with two or three objectives but ineffective for MOPs with more than three objectives [23]; the decomposition based MOEAs are effective for MOPs with any numbers of objectives, but they encounter difficulties in solving MOPs with irregular Pareto fronts [24]; the surrogate models are effective for MOPs with computationally expensive

DOI: 10.1109/SSCI44817.2019.9003018

objectives, but most of them can only solve MOPs with a limited number of decision variables [25].

On the other hand, it is almost impossible to determine the performance of an MOEA on a given MOP in theory, since most MOEAs are stochastic algorithms and have complicated procedures. Moreover, since many real-world MOPs have computationally expensive objectives [26], [27], it is also impractical to empirically compare the performance of a number of candidate MOEAs to determine which one is the best for a given MOP. To address this issue, machine learning models are needed to build a bridge between the performance of MOEAs and the type of MOPs. In contrast to some existing work that focuses on using models to optimize the parameters of algorithms [28], [29], this paper attempts to use a model to predict which MOEA out of a set of candidate MOEAs is most effective for a given MOP. More specifically, the proposed method first empirically compares the performance of a set of candidate MOEAs on multiple benchmark MOPs, then establishes a predictor based on the statistical results, where the inputs of the predictor are the features extracted from an MOP, and the output is the index of the MOEA having the best performance on the MOP. Therefore, the predictor is able to choose the most suitable MOEA from a set of candidate MOEAs for any given MOP.

We want to stress that in industry, only a small number of algorithms can be tested on most real-world application problems. In a recent work [30] one of the authors performed a comparison of just 10 MOEAs on a single application. Despite a reduced test regime, a simplified problem setting, no hyperparameter tuning, and the usage of a computing cluster with 35 cores, the benchmark took several weeks of computer runtime. Therefore any guidance on which MOEA to employ on an unknown problem is highly valuable and will support the usage of MOEAs in industry.

The rest of this paper is organized as follows. Section II introduces some basic definitions related to MOPs. Section III describes the methodologies of the proposed algorithm selection method. Section IV verifies the effectiveness of the proposed method by experimental studies. Finally, Section V draws the conclusions of this paper.

II. Preliminaries

An MOP can be mathematically defined as

$$\begin{aligned} \min_{\mathbf{x}} \ & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_M(\mathbf{x})) \\ \text{s.t.} \ & g_i(\mathbf{x}) \leq 0, \ i = 1, \cdots, I \\ & h_j(\mathbf{x}) = 0, \ j = 1, \cdots, J \\ & \mathbf{x} \in \Omega \end{aligned}, \tag{1}$$

where $\mathbf{x} = (x_1, \cdots, x_D) \in \Omega$ is a solution consisting of D decision variables, $\Omega \subseteq \mathbb{R}^D$ is the decision space, $\mathbf{f} : \Omega \to \Lambda \subseteq \mathbb{R}^M$ consists of M objectives, and Λ is the objective space. Besides, $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ are I inequality constraints and J equality constraints, respectively. This work only considers the MOPs without constraint, hence both I and J are equal to 0. In general, the MOPs with $M \geq 4$ are known as many-objective optimization problems, and the MOPs with $D \geq 100$ are known as large-scale MOPs.

A solution $\mathbf{x}$ is said to dominate another solution $\mathbf{y}$ if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ for every $i \in \{1, \cdots, M\}$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ for at least one $j \in \{1, \cdots, M\}$. A solution is called a Pareto optimal solution if it is not dominated by any solution in Ω. All the Pareto optimal solutions for an MOP constitute the Pareto set, and the image of the Pareto set in objective space is called the Pareto front. The goal of solving an MOP is to find a set of solutions with good convergence and diversity along the Pareto front.

The inverted generational distance (IGD) [31] can be adopted to quantitatively measure the quality of a solution set, which calculates the similarity between the solution set and a reference point set on the Pareto front:

$$IGD(P, R) = \frac{\sum_{\mathbf{r} \in R} \min_{\mathbf{p} \in P} \|\mathbf{p} - \mathbf{r}\|}{|R|}, \tag{2}$$

where P is the objective values of a solution set, R is a set of reference points uniformly sampled on the Pareto front, and $\|\cdot\|$ is the L_2-norm. In short, IGD calculates an average minimum distance from each point in R to the solutions in P, where a smaller IGD value indicates a better convergence and diversity of P. The methods for sampling a set of uniformly distributed reference points on the Pareto fronts of benchmark MOPs can be found in [32].

III. Methodologies

A. Extracting Features from MOPs

As a matter of fact, it is quite difficult to precisely represent the characteristics of an MOP by several numeric features, since most real-world MOPs have complex objective functions and are pursued based on large datasets [33]. Moreover, the objective functions and datasets of some real-world MOPs are even unknown to engineers [34]. Fortunately, we are interested in the performance of MOEAs on a given MOP rather than the characteristics of the given MOP, hence two types of features related to the performance of MOEAs are considered, namely, the explicit features and implicit features.

The explicit features of an MOP refer to the number of objectives M and the number of decision variables D. These two parameters can be easily observed from an MOP, and they are crucial to the performance of MOEAs. For example, some many-objective evolutionary algorithms like GrEA [35] and KnEA [36] outperform classical MOEAs (e.g., NSGA-II [11] and SPEA2 [12]) when $M > 3$, but they underperform classical MOEAs when $M \leq 3$; some large-scale MOEAs like MOEA/DVA [37] and LMEA [38] have better convergence performance than classical MOEAs when $D \geq 100$, but they converge more slowly than classical MOEAs when $D < 50$. As a consequence, these two parameters need to be considered in the proposed predictor.

The implicit features of an MOP refer to the landscape and Pareto front of the MOP. The landscape and Pareto front are closely related to the convergence performance and diversity performance of MOEAs, respectively. For example,

particle swarm optimization algorithms converge faster than genetic algorithms on MOPs with unimodal landscapes, but they usually have worse convergence performance than genetic algorithms on MOPs with multimodal landscapes [39]; decomposition based MOEAs have excellent diversity performance on MOPs with regular Pareto fronts, but they exhibit bad diversity performance on MOPs with irregular Pareto fronts [24]. However, both the landscape and Pareto front are determined by the objective functions of MOPs, which are difficult to be observed. In the proposed method, the following strategy is used to extract features to approximate the implicit features of an MOP. First, L solutions are generated in the decision space of the MOP by using Latin hypercube sampling [40]. Then, the objective values of the L solutions are regarded as the features. It is worth to note that although the features extracted by this strategy cannot directly represent the landscape and Pareto front of an MOP, they can be used to measure the similarity between a given MOP and a set of benchmark MOPs, so that the landscape and Pareto front of the most similar benchmark MOP can be regarded as those of the given MOP, and the performance of MOEAs on the most similar benchmark MOP can be regarded as the performance of MOEAs on the given MOP.

Based on the above features, a predictor is established in the proposed method to predict the best MOEA for a given MOP. In the next subsection, the procedure of the proposed method is elaborated.

B. Procedure of the Proposed Method

To begin with, the proposed method executes N_A candidate MOEAs on N_P benchmark MOPs, where the number of objectives of each MOP is set to N_M different values and the number of decision variables of each MOP is set to N_D different values. Therefore, there are $N_A \times N_P \times N_M \times N_D$ results in total. Then, a support vector machine (SVM) [41] is trained for each MOP, where the inputs of the SVM are the number of objectives and the number of decision variables, and the output is the index of the MOEA having the best IGD value on the benchmark MOP with the same settings to the inputs. As a result, there are $N_M \times N_D$ training samples for each SVM.

After the SVMs are trained, they can be used to predict the best MOEA for a given MOP, the procedure of which is given in Algorithm 1. First, L solutions are generated by using Latin hypercube sampling (Line 1), and the objective values of the L solutions on the given MOP and all the benchmark MOPs are calculated (Lines 3 and 6). Note that since the MOPs may have different decision spaces, the decision variables of X should be scaled according to the decision space of each MOP before calculating objective values (Lines 2 and 5). Then, the similarity between the given MOP and each benchmark MOP can be calculated as follows (Line 7):

$$Sim(P,Q) = \frac{1}{|P|}\sum_{i=1}^{|P|}\|\mathbf{p}_i - \mathbf{q}_i\|, \tag{3}$$

Algorithm 1: Predicting the best MOEA for a given MOP by the proposed method

Input: $\mathbf{f}$ (objective functions of a given MOP), M (number of objectives), D (number of decision variables)
Output: $best$ (index of the best MOEA)

1 $X \leftarrow$ Generate L solutions by Latin hypercube sampling;
2 $X' \leftarrow$ Scale the decision variables of X according to the decision space of the given MOP;
3 $P \leftarrow$ Calculate the objective values of X' on $\mathbf{f}$;
4 **for** $i \in \{1, \cdots, N_P\}$ **do**
5 $\quad X' \leftarrow$ Scale the decision variables of X according to the decision space of the i-th benchmark MOP;
6 $\quad Q_i \leftarrow$ Calculate the objective values of X' on the objective functions of the i-th benchmark MOP;
7 $\quad Sim_i \leftarrow$ Calculate the similarity between P and Q_i by (3);
8 $S \leftarrow$ The indices of the K smallest elements in Sim;
9
10 $A \leftarrow \emptyset$;
11 **for** $s \in S$ **do**
12 $\quad A \leftarrow A \cup \{SVM_s(M,D)\}$; `//` $SVM_s()$ `denotes the output of the SVM trained on the` s`-th benchmark MOP`
13 **for** $a \in A$ **do**
14 $\quad Rank_a \leftarrow 0$;
15 $\quad$ **for** $s \in S$ **do**
16 $\quad\quad r \leftarrow$ The rank of the a-th MOEA on the s-th benchmark MOP with M objectives and D decision variables;
17 $\quad\quad Rank_a \leftarrow Rank_a + r$;
18 $best \leftarrow \text{argmin}_a Rank_a$;
19 **return** $best$;

where $\mathbf{p}_i$ and $\mathbf{q}_i$ denote the objective values of the i-th solution in the population P for the given MOP and the population Q for a benchmark MOP, respectively. Afterwards, the K most similar benchmark MOPs to the given MOP are determined (Line 8), and the MOEAs having the best performance on the K benchmark MOPs are identified (Line 12). Finally, the average ranks of these MOEAs on the K benchmark MOPs are calculated (Line 16), and the index of the MOEA having the best average rank is considered as the final output.

To summarize, the proposed method adopts the implicit features of MOPs to find the most similar benchmark MOPs to a given MOP by using a similar strategy to k-means clustering. Afterwards, it adopts the explicit features of MOPs to determine the best MOEA for the given MOP by using SVM. In the next section, the effectiveness of the proposed algorithm selection method is empirically verified.

IV. Experimental Studies

A. Candidate MOEAs

In the experiments, ten state-of-the-art MOEAs are selected as candidate MOEAs, i.e., AR-MOEA [42], BCE-IBEA [43], GFM-MOEA [44], MOEA/D [14], MOEA/DD [45], NMPSO [46], NSGA-III [47], NSGA-II/SDR [23], RVEA [48], SPEA2+SDE [49]. A brief introduction to them is given in the following.

1) AR-MOEA is a recently proposed MOEA based on indicator. It improves the convergence speed by adopting an enhanced inverted generational distance (IGD-NS) [50] metric as the selection criterion, and enhances the versatility of reference point based MOEAs on MOPs with irregular Pareto fronts by suggesting a reference point adaptation strategy.
2) BCE-IBEA indicates the IBEA [19] with bi-criterion evolution. This MOEA enhances IBEA by adopting a Pareto criterion evolution, which maintains a set of representative non-dominated solutions and explores promising areas that are undeveloped by IBEA. As a result, it benefits from both Pareto dominance based MOEAs and indicator based MOEAs.
3) GFM-MOEA is a recently proposed MOEA, which adopts a generic front modeling method in the environmental selection. More specifically, it selects solutions according to their convergence and diversity along an estimated Pareto front, thus maintaining a good diversity on MOPs with different Pareto front shapes.
4) MOEA/D is a classical MOEA based on the decomposition of objective space. It converts the original MOP into a number of single-objective optimization problems and solves them simultaneously. In general, MOEA/D can obtain populations with excellent diversity on MOPs with regular Pareto fronts, but it has poor diversity performance on MOPs with irregular Pareto fronts [24].
5) MOEA/DD is a combination of MOEA/D and Pareto dominance. It follows the general framework of MOEA/D and adopts both penalty based boundary intersection (PBI) and Pareto dominance to update solutions. Experimental results demonstrated that MOEA/DD can strike a good balance between convergence and diversity.
6) NMPSO is a recently proposed multi-objective particle swarm optimization algorithm. It generates offspring by both particle swarm optimization and genetic operators, and uses a balanceable fitness estimation method in the environmental selection. Experimental results indicated that NMPSO has good performance in solving many-objective optimization problems.
7) NSGA-III can be regarded as a combination of NSGA-II [11] and MOEA/D, i.e., using both non-dominated sorting [51] and decomposition based selection. It sorts the solutions according to Pareto dominance, and distinguishes the solutions in the same level by a decomposition based selection strategy. As a result, NSGA-III can benefit from both NSGA-II and MOEA/D, thus exhibiting more robust performance.
8) NSGA-II/SDR indicates the NSGA-II with strengthened dominance relation. The strengthened dominance relation is a new dominance relation tailored for many-objective optimization problems, which can improve the selection pressure and better balance convergence and diversity. It was empirically verified that the strengthened dominance relation greatly improves the performance of NSGA-II on many-objective optimization problems.
9) RVEA is an effective MOEA based on decomposition. It has a similar framework to MOEA/D, but is equipped with a more effective aggregation function and a reference vector adaptation strategy. RVEA has a low computational complexity and a high convergence speed, and it has been applied to some complex MOPs in real-world applications [52], [53].
10) SPEA2+SDE indicates the SPEA2 [12] with shift based density estimation. The shift based density estimation covers both the distribution and convergence information of solutions when estimating the density of solutions, which can significantly improve the performance of SPEA2 on many-objective optimization problems.

B. Benchmark MOPs

The widely used problems WFG4–WFG9 [54] are adopted as benchmark MOPs in the experiments. WFG4–WFG9 have different transformation functions between decision space and objective space, thus posing various challenges to MOEAs. To diversify the shapes of the Pareto fronts, we inverse the Pareto fronts of WFG4–WFG9 by a similar strategy to that in [55] and construct six new benchmark MOPs, namely, IWFG4-IWFG9. As a result, there are 12 benchmark MOPs in total.

C. Parameter Settings

For the parameter settings of the candidate MOEAs and benchmark MOPs, we follow the default settings provided in PlatEMO [15]. Besides, the number of objectives for all the MOPs is set to 3, 4, 5, 6, 10, 11, 15 and 16, the number of decision variables for all the MOPs is set to 20, 25, 30, 35, 40, 45, 50 and 55, the population size for all the MOEAs is set to 105, 120, 105, 112, 110, 77, 120 and 152 when the number of objectives is 3, 4, 5, 6, 10, 11, 15 and 16, respectively, and the number of generations for all the MOEAs is set to 1, 10 and 100. The best candidate MOEA on each benchmark MOP is determined according to the IGD values averaged over 20 runs.

As for the parameter settings of the proposed method, the number of sampling solutions L is set to 100 and the number of most similar benchmark MOPs K is set to 3.

D. Effectiveness of the Implicit Features in the Proposed Method

This subsection verifies the effectiveness of the implicit features in the proposed method. For this aim, the results on a benchmark MOP are used as the test samples each time, and

TABLE I
THE TRUE RANK VALUES OBTAINED BY THE PROPOSED METHOD AND THE BASELINE METHOD, WHERE THE TRAINING SAMPLES AND TEST SAMPLES ARE WITH DIFFERENT BENCHMARK MOPS. THE NUMBER OF GENERATIONS IS SET TO 1. THE BETTER RESULT IN EACH ROW IS HIGHLIGHTED.

Settings	Proposed method	Baseline method
$M = 3, D = 20$	1.0000	1.0000
$M = 3, D = 30$	2.1667	1.5000
$M = 3, D = 40$	1.5833	1.5833
$M = 3, D = 50$	2.5833	2.0000
$M = 5, D = 20$	2.4167	2.4167
$M = 5, D = 30$	4.0000	5.5000
$M = 5, D = 40$	2.7500	3.5000
$M = 5, D = 50$	3.5833	3.0000
$M = 10, D = 20$	3.9167	3.5000
$M = 10, D = 30$	3.0833	6.2500
$M = 10, D = 40$	3.0833	4.0000
$M = 10, D = 50$	4.0000	4.0833
$M = 15, D = 20$	4.9167	5.1667
$M = 15, D = 30$	3.6667	4.1667
$M = 15, D = 40$	4.7500	4.2500
$M = 15, D = 50$	3.8333	4.2500
Average	3.2083	3.5104

TABLE II
THE TRUE RANK VALUES OBTAINED BY THE PROPOSED METHOD AND THE BASELINE METHOD, WHERE THE TRAINING SAMPLES AND TEST SAMPLES ARE WITH DIFFERENT BENCHMARK MOPS. THE NUMBER OF GENERATIONS IS SET TO 10. THE BETTER RESULT IN EACH ROW IS HIGHLIGHTED.

Settings	Proposed method	Baseline method
$M = 3, D = 20$	1.4167	1.6667
$M = 3, D = 30$	1.6667	1.6667
$M = 3, D = 40$	1.3333	1.6667
$M = 3, D = 50$	1.5833	1.7500
$M = 5, D = 20$	1.4167	1.4167
$M = 5, D = 30$	1.1667	1.3333
$M = 5, D = 40$	1.2500	1.5000
$M = 5, D = 50$	1.1667	1.3333
$M = 10, D = 20$	1.7500	2.4167
$M = 10, D = 30$	2.0833	2.0833
$M = 10, D = 40$	1.6667	1.7500
$M = 10, D = 50$	1.2500	1.8333
$M = 15, D = 20$	1.2500	1.5833
$M = 15, D = 30$	1.3333	1.9167
$M = 15, D = 40$	1.3333	1.5333
$M = 15, D = 50$	1.9167	1.8333
Average	1.4740	1.7052

TABLE III
THE TRUE RANK VALUES OBTAINED BY THE PROPOSED METHOD AND THE BASELINE METHOD, WHERE THE TRAINING SAMPLES AND TEST SAMPLES ARE WITH DIFFERENT BENCHMARK MOPS. THE NUMBER OF GENERATIONS IS SET TO 100. THE BETTER RESULT IN EACH ROW IS HIGHLIGHTED.

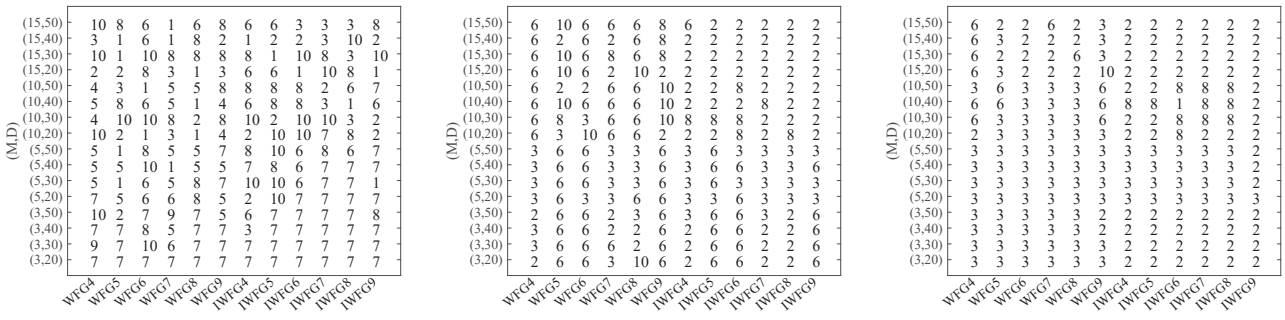
Settings	Proposed method	Baseline method
$M = 3, D = 20$	1.0000	2.0000
$M = 3, D = 30$	1.0000	2.0000
$M = 3, D = 40$	1.0833	1.4167
$M = 3, D = 50$	1.0833	1.4167
$M = 5, D = 20$	1.0000	1.0000
$M = 5, D = 30$	1.0833	1.0833
$M = 5, D = 40$	1.0833	1.0833
$M = 5, D = 50$	1.0833	1.0833
$M = 10, D = 20$	1.2500	1.7500
$M = 10, D = 30$	1.4167	2.2500
$M = 10, D = 40$	1.4167	2.4167
$M = 10, D = 50$	1.4167	2.3333
$M = 15, D = 20$	1.2500	1.2500
$M = 15, D = 30$	1.8333	1.3333
$M = 15, D = 40$	1.2500	1.2500
$M = 15, D = 50$	1.5833	1.2500
Average	1.2396	1.5573

the results on all the remaining benchmark MOPs are used as the training samples. In this way, the true rank of the MOEA predicted for each test sample can indicate the performance of the proposed method, where a lower true rank value indicates a better performance. Note that in this experiment, the number of objectives M is set to 3, 5, 10 and 15 and the number of decision variables D is set to 20, 30, 40 and 50 for all the training samples and test samples.

To better measure the performance of the proposed method, a baseline method is compared with the proposed method in this experiment, where the baseline method regards the MOEA having the lowest rank value averaged over all the training samples as the best MOEA for the test sample. That is, the baseline method does not use the implicit features to find the most similar benchmark MOPs to the test MOP.

Table I lists the true rank values obtained by the proposed method and the baseline method, averaged over all the test samples with the same number of objectives and the same number of decision variables. Besides, the number of generations is set to 1. It can be seen from Table I that the true rank values obtained by the two methods are relatively high, while the proposed method exhibits a slightly better overall performance than the baseline method. Table II gives the true rank values when the number of generations is set to 10. It can be found that the true rank values in Table II are lower than those in Table I, and the proposed method shows significantly better overall performance than the baseline method. Furthermore, Table III lists the true rank values when the number of generations is set to 100, where the proposed method still outperforms the baseline method, having achieved an average true rank value of 1.2396. This value means that the proposed method generally selects the best or the second best MOEA for the test MOP among ten candidate MOEAs. As a consequence, the implicit features are effective for the proposed method.

It is interesting to note that the true rank values in Tables II and III are much lower than those in Table I. This is attributed to the larger computational budget provided for the MOEAs, which makes the difference between the performance of MOEAs more significant and easier to be detected by the proposed method. As can be further observed from Fig. 1, the indices of the best MOEAs on the benchmark MOPs become



(a) 1 generation (b) 10 generations (c) 100 generations

Fig. 1. Indices of the best MOEA on WFG4–WFG9 and IWFG4–IWFG9 with $M = 3, 5, 10, 15$ and $D = 20, 30, 40, 50$. (1. AR-MOEA 2. BCE-IBEA 3. GFM-MOEA 4. MOEA/D 5. MOEA/DD 6. NMPSO 7. NSGA-III 8. NSGA-II/SDR 9. RVEA 10. SPEA2+SDE)

TABLE IV
THE TRUE RANK VALUES OBTAINED BY THE PROPOSED METHOD, WHERE THE TRAINING SAMPLES AND TEST SAMPLES ARE WITH DIFFERENT NUMBERS OF OBJECTIVES AND DECISION VARIABLES. THE NUMBER OF GENERATIONS IS SET TO 1.

M \ D	$D = 25$	$D = 35$	$D = 45$	$D = 55$
$M = 4$	3.5833	2.8333	3.9167	3.0000
$M = 6$	4.7500	3.1667	2.9167	4.3333
$M = 11$	3.7500	3.8333	4.5000	6.1667
$M = 16$	3.1667	3.1667	3.6667	2.4167
Average	3.6979			

TABLE V
THE TRUE RANK VALUES OBTAINED BY THE PROPOSED METHOD, WHERE THE TRAINING SAMPLES AND TEST SAMPLES ARE WITH DIFFERENT NUMBERS OF OBJECTIVES AND DECISION VARIABLES. THE NUMBER OF GENERATIONS IS SET TO 10.

M \ D	$D = 25$	$D = 35$	$D = 45$	$D = 55$
$M = 4$	1.3333	1.2500	1.3333	1.1667
$M = 6$	2.0000	1.8333	2.0000	2.0000
$M = 11$	2.6667	2.3333	2.6667	2.8333
$M = 16$	2.2500	2.4167	2.0833	2.2500
Average	2.0260			

TABLE VI
THE TRUE RANK VALUES OBTAINED BY THE PROPOSED METHOD, WHERE THE TRAINING SAMPLES AND TEST SAMPLES ARE WITH DIFFERENT NUMBERS OF OBJECTIVES AND DECISION VARIABLES. THE NUMBER OF GENERATIONS IS SET TO 100.

M \ D	$D = 25$	$D = 35$	$D = 45$	$D = 55$
$M = 4$	1.4167	1.4167	1.4167	1.4167
$M = 6$	1.0000	1.1667	1.1667	1.1667
$M = 11$	2.1667	2.6667	1.9167	2.9167
$M = 16$	1.9167	1.2500	1.1667	1.2500
Average	1.5885			

more regular with the increase of the number of generations. On the other hand, it can be seen from Fig. 1(c) that there are more than one best MOEAs, hence it is unreasonable to use a single MOEA to solve all the benchmark MOPs. In fact, BCE-IBEA obtains the best average rank value on all the benchmark MOPs, but this value (1.7604) is larger than that obtained by the proposed method (1.2396).

E. Effectiveness of the Explicit Features in the Proposed Method

The effectiveness of the explicit features in the proposed method is verified in this subsection. In this experiment, the results on all the benchmark MOPs with $M = 3, 5, 10, 15$ and $D = 20, 30, 40, 50$ are used as the training samples, and the results on all the benchmark MOPs with $M = 4, 6, 11, 16$ and $D = 25, 35, 45, 55$ are used as the test samples.

Table IV lists the true rank values obtained by the proposed method, averaged over all the test samples with the same number of objectives and the same number of decision variables. Besides, the number of generations is set to 1. It can be seen from Table IV that the true rank values are slightly larger than those in Table I, which is mainly caused by the error of the SVMs. Moreover, Tables V and VI show the true rank values when the number of generations is set to 10 and 100, respectively. It is obvious that the true rank values decrease with the increase of the number of generations, which is consistent with those shown in Tables I–III. As a consequence, the explicit features are also effective for the proposed method, and they can lead to an average true rank value of 1.5885 when the number of generations is 100.

V. CONCLUSIONS AND FUTURE WORK

This paper has proposed a method for automatically choosing the best MOEA from a set of candidate MOEAs for a given MOP. The proposed method first collects the performance of a set of candidate MOEAs on multiple benchmark MOPs as training data, then establishes a predictor for suggesting the best MOEA for a given MOP, where the inputs are the explicit and implicit features of the given MOP, and the

output is the index of the suggested MOEA among all the candidate MOEAs. The explicit features of an MOP refer to the number of objectives and the number of decision variables, and the implicit features are represented by the objective values of a number of solutions sampled by Latin hypercube sampling. The effectiveness of both the explicit features and implicit features have been verified in experimental studies. The experimental results have also indicated that the proposed method can generally select the best or the second best MOEA for a given MOP.

This work has revealed the possibility of predicting the best MOEA from a set of candidate MOEAs for real-world MOPs, and further investigation on this topic is still desirable. Firstly, the characteristics of the landscape and Pareto front can be used as the implicit features, which can be learnt based on a large number of offline samples of benchmark MOPs by using deep learning models. Secondly, the automatic component-wise design can be considered to solve real-world MOPs by multiple MOEAs simultaneously [56]. Finally, some other techniques such as transfer optimization [57] and evolutionary multitasking [58] can be employed to decrease the difficulty of real-world MOPs with the help of benchmark MOPs.

On the other hand, since most academic benchmark functions are cheap in terms of required computational budget, one could consider to scale this approach up to many more MOEAs and benchmarks, possibly even adding different hyper-parameter configurations for each algorithm. The resulting large-scale database could be tackled with data mining methods to look for so far hidden correlations, sweet spots, or weaknesses. Empirical findings from this database could be compared to theoretical work and possibly lead to new research directions. One could even look for dominated algorithms, that is MOEAs that are always outperformed by other algorithms, paving the way for a procedure to actually reduce the number of competitor algorithms researchers and practitioners have to consider.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China under Grant 61672033, 61822301, 61876123, 61906001, and U1804262, in part by the State Key Laboratory of Synthetical Automation for Process Industries under Grant PAL-N201805, and in part by the Anhui Provincial Natural Science Foundation under Grant 1808085J06 and 1908085QF271. Ye Tian acknowledges the financial support from Honda Research Institute Europe.

REFERENCES

[1] J. Handl, D. B. Kell, and J. Knowles, "Multiobjective optimization in bioinformatics and computational biology," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 4, no. 2, pp. 279–292, 2007.

[2] J. G. Herrero, A. Berlanga, and J. M. M. López, "Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 151–168, 2009.

[3] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 3, pp. 321–344, 2013.

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

[5] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer Science & Business Media, 2007.

[6] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Computing*, 2017, in press.

[7] J. D. Schaffer, "Multiple objective optimization with vector evaluated genetic algorithms," in *Proceedings of the 1st International Conference on Genetic Algorithms*, 1985, pp. 93–100.

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

[9] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evolutionary Computation*, vol. 2, no. 3, pp. 221–248, 1995.

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

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

[12] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Proceedings of the Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, 2001, pp. 95–100.

[13] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multi-objective optimization," in *Proceedings of the 2001 Conference on Genetic and Evolutionary Computation*, 2001, pp. 283–290.

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

[15] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[16] C. A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization," in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, vol. 2, 2002, pp. 1051–1056.

[17] S. Kukkonen and J. Lampinen, "GDE3: The third evolution step of generalized differential evolution," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 1, 2005, pp. 443–450.

[18] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[19] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*, 2004, pp. 832–842.

[20] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[21] K. Li, R. Wang, T. Zhang, and H. Ishibuchi, "Evolutionary many-objective optimization: A comparative study of the state-of-the-art," *IEEE Access*, vol. 6, pp. 26 194–26 214, 2018.

[22] T. Rodemann, "Industrial portfolio management for many-objective optimization algorithms," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation*, 2018.

[23] Y. Tian, R. Cheng, X. Zhang, Y. Su, and Y. Jin, "A strengthened dominance relation considering convergence and diversity for evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2018, in press.

[24] 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.

[25] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, "A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2018, in press.

[26] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 6, pp. 939–952, 2016.

[27] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multi-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, 2019, in press.

[28] F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Sequential model-based optimization for general algorithm configuration," in *Proceedings of the 2012 International Conference on Learning and Intelligent Optimization*, 2012, pp. 507–523.

[29] A. Blot, H. H. Hoos, L. Jourdan, M. E. Kessaci-Marmion, and H. Trautmann, "MO-ParamILS: A multi-objective automatic algorithm configuration framework," in *Proceedings of the 2016 International Conference on Learning and Intelligent Optimization*, 2016, pp. 32–47.

[30] T. Rodemann, "A comparison of different many-objective optimization algorithms for energy system optimization," in *Proceedings of the 2019 International Conference on the Applications of Evolutionary Computation*, 2019.

[31] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[32] Y. Tian, X. Xiang, X. Zhang, R. Cheng, and Y. Jin, "Sampling reference points on the Pareto fronts of benchmark multi-objective optimization problems," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation*, 2018, in press.

[33] Y. Jin, H. Wang, T. Chugh, and D. Guo, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Transactions on Evolutionary Computation*, 2018, in press.

[34] G. Chen, H. Xu, G. Liu, J. Chao, and Z. Zhao, "An efficient multi-objective optimization method for black-box functions using sequential approximate technique," *Applied Soft Computing*, vol. 12, no. 1, pp. 14–27, 2012.

[35] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[36] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[37] X. Ma, F. Liu, Y. Qi, X. Wang, L. Li, L. Jiao, M. Yin, and M. Gong, "A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 2, pp. 275–298, 2016.

[38] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2018.

[39] X. Zhang, X. Zheng, R. Cheng, J. Qiu, and Y. Jin, "A competitive mechanism based multi-objective particle swarm optimizer with fast convergence," *Information Sciences*, vol. 427, pp. 63–76, 2018.

[40] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000.

[41] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and Their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

[42] Y. Tian, R. Cheng, X. Zhang, F. Cheng, and Y. Jin, "An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility," *IEEE Transactions on Evolutionary Computation*, 2017, in press.

[43] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, 2015.

[44] Y. Tian, X. Zhang, R. Cheng, C. He, and Y. Jin, "Guiding evolutionary multi-objective optimization with generic front modeling," *IEEE Transactions on Cybernetics*, 2018, in press.

[45] K. Li, K. Deb, Q. Zhang, and S. Kwong, "Combining dominance and decomposition in evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[46] Q. Lin, S. Liu, Q. Zhu, C. Tang, R. Song, J. Chen, C. A. C. Coello, K. C. Wong, and J. Zhang, "Particle swarm optimization with a balanceable fitness estimation for many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 32–46, 2018.

[47] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[48] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[49] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[50] Y. Tian, X. Zhang, R. Cheng, and Y. Jin, "A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation*, 2016, pp. 5222–5229.

[51] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to non-dominated sorting for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 201–213, 2015.

[52] R. Cheng, T. Rodemann, M. Fischer, M. Olhofer, and Y. Jin, "Evolutionary many-objective optimization of hybrid electric vehicle control: From general optimization to preference articulation," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 1, no. 2, pp. 97–111, 2017.

[53] T. Chugh, K. Sindhya, K. Miettinen, Y. Jin, T. Kratky, and P. Makkonen, "Surrogate-assisted evolutionary multiobjective shape optimization of an air intake ventilation system," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation*, 2017, pp. 1541–1548.

[54] L. B. S. Huband, P. Hingston and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[55] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[56] L. C. T. Bezerra, M. López-Ibáñez, and T. Stützle, "Automatic component-wise design of multi-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 403–417, 2016.

[57] A. Gupta, Y. S. Ong, and L. Feng, "Insights on transfer optimization: Because experience is the best teacher," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 51–64, 2018.

[58] L. Feng, L. Zhou, J. Zhong, A. Gupta, Y. S. Ong, K. C. Tan, and A. K. Qin, "Evolutionary multitasking via explicit autoencoding," *IEEE Transactions on Cybernetics*, 2018, in press.