An Improved Fuzzy Classifier-based Evolutionary Algorithm for Expensive Multiobjective Optimization Problems with Complicated Pareto Sets

Jinyuan Zhang, Linjun He, and Hisao Ishibuchi*

Guangdong Provincial Key Laboratory of Brain-inspired Intelligent Computation, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China.
zhangjy@sustech.edu.cn, this.helj@gmail.com, hisao@sustech.edu.cn

Abstract. Various surrogate-based multiobjective evolutionary algorithms (MOEAs) have been proposed to solve expensive multiobjective optimization problems (MOPs). However, these algorithms are usually examined on test suites with unrealistically simple Pareto sets (e.g., ZDT and DTLZ test suites). Real-world MOPs usually have complicated Pareto sets, such as a vehicle dynamic design problem and a power plant design optimization problem. Such MOPs are challenging to construct reliable surrogates for surrogate-based MOEAs. Constructed surrogates with low accuracy are likely to make incorrect predictions and even mislead the search direction. In this paper, we propose an improved fuzzy classifier-based MOEA by leveraging the accuracy information of the classifier. The proposed algorithm is compared with five state-of-the-art algorithms on two well-known test suites with complicated Pareto sets and four real-world problems. Experimental results demonstrate the effectiveness of the proposed algorithm in solving realistic MOPs with complicated Pareto sets when only a limited number of function evaluations are available.

Keywords: Expensive multiobjective optimization · Evolutionary algorithms · Fuzzy classifier · Surrogate models · Complicated Pareto set.

1 Introduction

Engineering optimization problems usually have two or more conflicting objectives, known as multiobjective optimization problems (MOPs) [16, 5] that need to be optimized simultaneously. A number of multiobjective evolutionary algorithms (MOEAs) have been proposed to solve MOPs [45]. Typically, MOEAs can be classified into three categories: dominance-based MOEAs [9, 47], indicator-based MOEAs [2, 46, 33], and decomposition-based MOEAs [27, 42]. These MOEAs usually evaluate the quality of solutions based on the evaluated

* Corresponding author.

objective function values and require a large number of function evaluations (FEs) [23, 33]. However, FEs are usually computationally expensive in engineering MOPs where the evaluation of a solution requires physical simulations that consume a large amount of time or resources [5]. The available number of FEs is usually limited for solving these expensive MOPs.

Several methods have been proposed for solving expensive MOPs. One of the most efficient methods is surrogate-based MOEAs [17, 4, 8]. Generally, surrogate-based MOEAs use computationally cheap surrogate models to replace the original objective functions or fitness functions to evaluate the quality of solutions. These surrogate-based MOEAs can be classified into two categories depending on the types of surrogate models: regression-based MOEAs [17, 8, 24, 30] and classification-based MOEAs [26, 41, 29].

- Regression-based MOEAs use regression models to approximate the original objective functions or fitness functions of MOPs. The constructed models are used to evaluate the quality of solutions. Generally, the number of constructed models is the same as the number of objective functions, with one model for each objective function [21, 43, 3]. Therefore, the time consumption of model construction is high, and this consumption will increase with the increase in the number of objectives. Some algorithms have been proposed to reduce the number of constructed regression models [8, 12].
- Classification-based MOEAs use classifiers to model the relation among solutions, e.g., the Pareto dominance relation among solutions. These classifiers are used to select promising solutions for subsequent optimization procedures. Since classification-based MOEAs usually build one classifier to model the relation among solutions, the number of constructed models is smaller than that in the regression-based methods.

However, these surrogate-based algorithms have usually been examined on test suites with simple Pareto sets. In Table 1, we summarize some typical surrogate-based MOEAs and the test suites used in their experimental studies. We can see that ZDT [7], DTLZ [11] and WFG [14] test suites are commonly used to examine the performance of surrogate-based MOEAs. However, the Pareto sets (PSs) of most of these test problems are linear and parallel to coordinate axes, which are simple and unrealistic [22, 23]. Real-world MOPs, such as a vehicle dynamic design problem [19] and a power plant design optimization problem [13], usually have complicated PSs [23, 28, 15, 10] due to the linkages between variables [28, 10] and the nonlinear shape of PSs [23]. It is worth noting that real-world MOPs with complicated PSs are challenging to construct reliable surrogates for surrogate-based MOEAs. Constructed surrogates with low accuracy are likely to make incorrect predictions and even mislead the search direction. Although the accuracy of the surrogates can be measured during model construction, it is rarely used as an indicator to guide the search.

In this paper, we improve our previous work [39] and propose an improved fuzzy classifier-based multiobjective evolutionary algorithm (IFCS-MOEA) by leveraging the accuracy information of the classifier. A novel sorting mechanism

Table 1. Typical surrogate-based MOEAs and the test suites used in their experimental studies.

	Algorithm	Year	Test suites
Regression-based MOEAs	ParEGO [21]	2006	KNO1 [21], OKA [28], VLMOP [35], ZDT [7], DTLZ [11]
	MOEA/D-EGO [3]	2010	KNO1 [21], VLMOP2 [35], ZDT [7], LZ [23], DTLZ [11]
	K-RVEA [3]	2018	DTLZ [11], WFG [14]
	KTA2 [30]	2021	DTLZ [11], WFG [14]
	EDN-ARMOEA [12]	2022	DTLZ [11], WFG [14]
Classification-based MOEAs	CSEA [29]	2019	DTLZ [11], WFG [14]
	θ-DEA-DP [36]	2022	DTLZ [11], WFG [14]
	MCEA/D [31]	2022	DTLZ [11], WFG [14]

is proposed to consider the membership degree of each solution and the accuracy of the classifier simultaneously. The proposed algorithm is compared with five state-of-the-art surrogate-based algorithms on two well-known test suites with complicated PSs and four real-world optimization problems to show its superiority in dealing with realistic expensive MOPs.

The rest of this paper is organized as follows. Section 2 presents related work to this paper. Section 3 presents the proposed IFCS-MOEA framework in detail. Section 4 examines the effectiveness of the proposed framework and compares it with five state-of-the-art algorithms. Section 5 concludes this paper.

2 Related Work

2.1 Multiobjective Optimization Problems

Typically, an MOP can be expressed as follows:

$$\begin{aligned} &\text{Minimize } F(x) = (f_1(x), \cdots, f_M(x))^{\mathrm{T}}, \\ &\text{subject to } x \in \Omega \subset R^n, \end{aligned} \tag{1}$$

where x is an n-dimensional decision vector, Ω is the decision space, $F(x)$ is an M-dimensional objective vector, and $f_i(x)$, $i = 1, \dots, M$ is the i-th objective function.

Since the objective functions in Eq. (1) are usually in conflict with each other, it is impossible to find a single optimal solution that can optimize all objective functions simultaneously. Therefore, Pareto optimal solutions are defined. Let u and v be two solutions to Eq. (1). u is said to dominate v, if $f_i(u) \leq f_i(v)$ for $i = 1, \dots, M$ and $f_j(u) < f_j(v)$ for at least one $j \in \{1, \dots, M\}$. Solution u is regarded as a Pareto optimal solution if there does not exist any solution that dominates u. The Pareto set (PS) is defined as the set of all Pareto optimal solutions. The Pareto front (PF) is defined as the image of the PS in the objective space.

2.2 Surrogate-based MOEAs

Regression models are widely used in surrogate-based MOEAs to approximated the objective functions of MOPs. Knowles [21] proposed to use an efficient global optimization (EGO) algorithm [18] to solve expensive MOPs. The proposed algorithm constructed a Gaussian process model to mimic the landscape of MOPs. Zhang et al. [43] combined the EGO algorithm with MOEA/D to solve expensive MOPs. The proposed algorithm constructed a Gaussian model to mimic the landscape of each decomposed subproblem of an MOP. Chugh et al. [3] combined the Kriging model with a reference vector guided evolutionary algorithm to solve expensive MOPs. The proposed algorithm constructed each Kriging model to mimic each objective function of an MOP. Song et al. [30] combined the Kriging model with a two-archive evolutionary algorithm. The proposed method constructed each Kriging model to approximate each objective function of an MOP.

Generally, solutions in a population in MOEAs can be divided into two categories: non-dominated solutions and dominated solutions, based on the Pareto dominance relation among them. Therefore, classifiers can be built to mimic this relation among solutions and can be used to select promising solutions. Loshchilov et al. [26] combined a classifier with a regression model to predict the dominance relation between a new solution and the existing non-dominated solutions. Bandaru et al. [1] applied multi-class classifiers to mimic the dominance relation between each pair of solutions. Zhang et al. [41, 40] employed classifiers to model the dominance relation among solutions and to pre-select promising offspring solutions. Lin et al. [25] used a classifier to pre-select promising offspring solutions, thereby reducing the required number of FEs of MOEA/D. Pan et al. [29] applied a classifier to predict the dominance relation between a new solution and the reference solutions. Zhang et al. [38, 39] employed a fuzzy classifier to assist environmental selection of MOEAs. Class labels and membership degrees were used to select promising offspring solutions for function evaluations. Yuan et al. [36] proposed to use two feedforward neural network models for solving expensive MOPs. One model was used to predict the Pareto dominance relation between solutions, and another model was built to predict the θ-dominance relation among solutions. Sonoda et al. [31] proposed to use multiple classifiers for solving high-dimensional expensive MOPs. Each classifier was constructed for each subproblem in the MOEA/D-DE algorithm. Zhang et al. [37] proposed a dual fuzzy-classifier-based surrogate model. One fuzzy classifier was constructed to learn the Pareto dominance relation among solutions, and another fuzzy classifier was constructed to learn the crowdedness of solutions.

3 Our Proposed Algorithm

This section presents the details of our improved fuzzy classifier-based MOEA (IFCS-MOEA) framework. IFCS-MOEA is proposed by using an improved fuzzy classifier-based surrogate model (IFCS). The IFCS model is constructed for sorting unevaluated solutions. First, Section 3.1 presents the general framework of IFCS-MOEA. Then, Section 3.2 describes IFCS-based sorting strategy in detail.

Algorithm 1: Framework of IFCS-MOEA

1 Initialize the population $P = \{x^1, x^2, \cdots, x^N\}$, and evaluate the solutions in P;
2 Set $Arc = P$;
3 **while** *termination condition is not satisfied* **do**
4   Set $A_+ =$ `Non-dominated_Selection`(Arc) and $A_- = Arc \backslash A_+$;
5   Construct a classifier $[l, md_+] =$ `fuzzy_classifier_construction`(x) by using A_+ and A_-;
6   Validate the accuracy of the classifier $Accuracy =$ `k-fold`(Arc);
7   Set $Q_p = \emptyset$;
8     $Mating\_P = P$;
9   **while** $w < w_{max}$ **do**
10   Generate $2N$ offspring solutions $Q = \{y^1, \cdots, y^{2N}\}$ by using $Mating\_P$;
11   Sort the offspring solutions $Q =$ `IFCS_Sorting`$(Q, Accuracy)$;
12   Select the top N solutions Q_{top} from Q;
13         $Q_p = Q_p \cup Q_{top}$;
14         $Mating\_P = Q_{top}$;
15         $w = w + 1$;
16   **end**
17   Sort all solutions in Q_p by $Q_p =$ `IFCS_Sorting`$(Q_p, Accuracy)$;
18   Select the top η solutions Q_{eval} from Q_p and evaluate them;
19     $Arc = Arc \cup Q_{eval}$;
20     $P =$ `Environmental_Selection`(Arc, N);
21 **end**

3.1 Algorithm Framework

The framework of the proposed IFCS-MOEA is presented in Algorithm 1. It is composed of four main procedures as follows.

- Initialization: N solutions are initialized and evaluated in Line 1. All the evaluated solutions are collected in Arc in Line 2.
- Fuzzy classifier construction: All the solutions in the archive are used as training data to construct a fuzzy classifier. The Pareto dominance relation is used to define two classes of the training data in Line 4. The non-dominated solutions are positive, and the dominated solutions are negative. A fuzzy classifier is constructed in Line 5. This paper uses a Fuzzy-KNN classifier [20] to construct the IFCS model. The fuzzy-KNN uses fuzzy similarity to predict the class of each solution. When a fuzzy classifier is used to predict the quality of a new solution, the class label l of the new solution and the membership degree to each class are obtained. A membership degree indicates the degree to the class which a new solution belongs to. A new solution's membership degree is calculated based on its K nearest neighbor's membership degrees. In this paper, we use the classifier to deal with the two-class problem. The membership degree md_+ in Line 5 is only for the positive class while the membership degree for the negative class is $1 - md_+$. $md_+ = 0.5$ is used as the classification boundary. If $md_+ \geq 0.5$, the solution is labeled as positive, otherwise it is negative. The k-fold cross-validation method is applied to validate the effectiveness of the classifier in Line 6. The accuracy of the classifier is obtained.

- Offspring generation: $2N$ offspring solutions are generated by using the mating population in Line 10. Next, the IFCS model is applied to sort the $2N$ offspring solutions in Line 11. The top N promising offspring solutions are selected and stored in Line 12. Then, these selected solutions are used as mating solutions to generate new offspring solutions. This offspring generation process is repeated w_{max} times.
- New population generation: The IFCS model is used to sort all selected $w_{max} \times N$ offspring solutions in Line 17. The top η solutions are selected and evaluated by the objective functions in Line 18. The archive is updated by using the newly evaluated solutions in Line 19. Finally, the environmental selection mechanism of an MOEA is used to select N solutions from Arc to form the new population for the next generation in Line 20.

3.2 IFCS-based Sorting

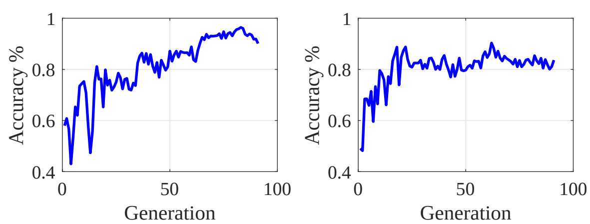
After the fuzzy classifier is constructed, the k-fold cross-validation method is used to measure the reliability of the classifier. The mean accuracy ($Accuracy$) of the classifier is obtained after the validation. In our algorithm framework, we use $k = 10$ for experiments.

As mentioned in Section 3.1, for a solution, if its membership degree with respect to the positive class is $md_+ \geq 0.5$, the solution is classified as a positive solution by the classifier with small uncertainty. When the $0 \leq md_+ < 0.5$, the solution is classified as a negative solution with small uncertainty. When the md_+ value is close to 0.5, the classification result has a large uncertainty in the class prediction.

Based on the above considerations, we propose an IFCS-based sorting strategy to sort solutions based on the model accuracy and membership degrees. The details of the proposed IFCS-based sorting strategy are presented in Algorithm 2. The constructed fuzzy classifier is used to predict the label l and the membership degree md_+ (with respect to the positive class) of each solution in Q (Line 1). These solutions are ranked in different manners according to the accuracy of the classifier and the membership degree to the positive class.

Fig. 1 plots the accuracy of the fuzzy classifier at each generation through the execution of IFCS-MOEA/D-DE on UF8 and LZ5 test problems with the median IGD values over 21 runs. The two figures show that the accuracy of the model is low at the beginning of optimization. The accuracy will increase along with the increase of generations. The reason is that since the size of the training data set is small in early generations, the model constructed by using these data cannot approach the true relation among solutions and is hard to make correct predictions. After several generations, the size of the training data set increases, the model can approach the true relation among solutions and the accuracy of the prediction increases. For this reason, we consider the following three cases according to the model accuracy. We specify the threshold values as 30% and 70% since the accuracy of the model is usually larger than 70% in our experiments as shown in Fig. 1.

```
Algorithm 2: Q = IFCS_Sorting(Q, Accuracy)
Predict the label and membership degree of each solution in Q by
  [l, md+] = fuzzy_classifier_prediction(y);
if Accuracy ≥ 70% then
    Sort solutions in Q with respect to their membership degrees in
      descending order;
else if 30% ≤ Accuracy < 70% then
    Q_p = {y ∈ Q | l = 1};
    Q_n = {y ∈ Q | l ≠ 1};
    Sort solutions in Q_p with respect to their membership degrees in
      ascending order;
    Sort solutions in Q_n with respect to their membership degrees in
      descending order;
    Q = Q_p ∪ Q_n, the solutions in Q_p are ranked before the solutions in Q_n;
else
    Sort solutions in Q with respect to their membership degrees in ascending
      order;
end
```



(a) UF8 (b) LZ5

Fig. 1. The accuracy versus generation obtained by IFCS-MOEA/D-DE on UF8 and LZ5 with the median IGD values over 21 runs.

- $Accuracy \geq 70\%$: The unevaluated solutions are ranked in descending order with respect to md_+ values. This is because the model accuracy is high and we can trust the predictions of the classifier.
- $30\% \leq Accuracy < 70\%$: First, positive solutions are ranked in ascending order with respect to md_+ values. This is because the model is more uncertain for the prediction of a solution with a smaller md_+ value than that with a larger md_+ value for the positive class. Evaluating uncertain solutions can improve the model accuracy (after evaluation, these solutions will be added to training data). Next, negative solutions are ranked in descending order with respect to md_+ values. This is because the model is more uncertain for the prediction of the solution with a larger md_+ value than that with a smaller md_+ value for the negative class. Then, the positive solutions are ranked before the negative solutions.
- $Accuracy < 30\%$: The unevaluated solutions are ranked in ascending order with respect to md_+ values. This is because the model accuracy is too small and we cannot trust the predictions of the classifier.

Table 2. Example of four unevaluated solutions.

	s_1	s_2	s_3	s_4
Label predicted by the classifier	0	1	1	0
Membership degree with respect to the positive class (md_+)	0.4	0.9	0.6	0.1

For example, suppose we have four solutions in Q as shown in Table 2. Each solution has a label and a membership degree with respect to the positive class predicted by the classifier. When the accuracy of the classifier is larger than or equal to 70%, these solutions are ranked in descending order with respect to their membership degrees (i.e., $s_2 > s_3 > s_1 > s_4$). When the accuracy of the classifier is larger than or equal to 30% and smaller than 70%, the positive solutions are ranked in ascending order with respect to their membership degrees (i.e., $s_3 > s_2$). Next, the negative solutions are ranked in descending order with respect to their membership degrees (i.e., $s_1 > s_4$). Then, the positive solutions are ranked before the negative solutions (i.e., $s_3 > s_2 > s_1 > s_4$). When the accuracy of the classifier is smaller than 30%, these solutions are ranked in ascending order with respect to their membership degrees (i.e., $s_4 > s_1 > s_3 > s_2$).

4 Experiments

This section examines the effectiveness of the proposed IFCS-MOEA framework. First, Section 4.1 presents the experimental settings. Second, Section 4.2 examines the effect of IFCS on MOEA/D-DE. Then, Section 4.3 compares the performance of IFCS-MOEA/D-DE with five state-of-the-art MOEAs on 19 test problems. Finally, Section 4.4 compares the performance of IFCS-MOEA/D-DE with five state-of-the-art MOEAs on four real-world application problems.

4.1 Experimental Settings

MOEA/D-DE [23] is integrated with the proposed framework for experiments, and the resulting algorithm is named IFCS-MOEA/D-DE. Five surrogate-based MOEAs, i.e., FCS-MOEA/D-DE [39], CPS-MOEA [41], CSEA [29], MOEA/D-EGO [43] and EDN-ARM-OEA [12] are used for comparison. UF1–10, LZ1–9 test problems [44, 23] with complicated PSs are used for experiments. Among them, UF1–7, LZ1–5, and LZ7–9 have 2 objectives, UF8–10, and LZ6 have 3 objectives. UF1–10, LZ1–5, and LZ9 are with 30 decision variables, and LZ6–8 are with 10 decision variables. The population size N is set to 45 for all compared algorithms. The maximum number of FEs is set as 500 since the problems are viewed as expensive MOPs [39]. For each test problem, each algorithm is executed 21 times independently. For IFCS-MOEA/D-DE, w_{max} is set to 30 and η is set to 5. For the other algorithms, we use the settings suggested in their papers. The IGD [6] metric is used to evaluate the performance of each algorithm. All algorithms are examined on PlatEMO [34] platform.

Table 3. The $mean_{std}$ IGD values of IFCS-MOEA/D-DE and MOEA/D-DE on UF1–10 and LZ1–9.

	IFCS-MOEA/D-DE	MOEA/D-DE
UF1	$8.87e\text{-}01_{1.03e-01}$	$1.04e{+}00_{1.48e-01}(-)$
UF2	$1.88e\text{-}01_{2.87e-02}$	$2.17e\text{-}01_{1.90e-02}(-)$
UF3	$6.04e\text{-}01_{2.57e-02}$	$6.51e\text{-}01_{4.11e-02}(-)$
UF4	$1.32e\text{-}01_{6.34e-03}$	$1.37e\text{-}01_{7.67e-03}(\sim)$
UF5	$4.20e{+}00_{3.67e-01}$	$4.49e{+}00_{3.60e-01}(-)$
UF6	$3.78e{+}00_{5.72e-01}$	$4.52e{+}00_{4.10e-01}(-)$
UF7	$9.50e\text{-}01_{1.11e-01}$	$1.11e{+}00_{1.21e-01}(-)$
UF8	$7.24e\text{-}01_{9.32e-02}$	$7.93e\text{-}01_{1.34e-01}(\sim)$
UF9	$7.57e\text{-}01_{8.23e-02}$	$8.99e\text{-}01_{9.73e-02}(-)$
UF10	$4.65e{+}00_{4.34e-01}$	$5.10e{+}00_{6.81e-01}(-)$
LZ1	$1.55e\text{-}01_{8.70e-03}$	$1.71e\text{-}01_{1.75e-02}(-)$
LZ2	$8.99e\text{-}01_{1.69e-01}$	$1.07e{+}00_{1.54e-01}(-)$
LZ3	$2.22e\text{-}01_{1.45e-02}$	$2.61e\text{-}01_{2.37e-02}(-)$
LZ4	$2.15e\text{-}01_{2.08e-02}$	$2.61e\text{-}01_{2.59e-02}(-)$
LZ5	$2.00e\text{-}01_{2.53e-02}$	$2.21e\text{-}01_{1.97e-02}(-)$
LZ6	$4.92e\text{-}01_{4.56e-02}$	$5.74e\text{-}01_{1.17e-01}(-)$
LZ7	$1.02e{+}00_{3.52e-01}$	$1.30e{+}00_{2.39e-01}(-)$
LZ8	$8.33e\text{-}01_{1.23e-01}$	$8.94e\text{-}01_{1.14e-01}(\sim)$
LZ9	$9.52e\text{-}01_{1.43e-01}$	$1.08e{+}00_{1.22e-01}(-)$
$+/-/\sim$		0/16/3

4.2 Effect of IFCS on MOEA/D-DE

This section examines the effectiveness of IFCS-MOEA framework on MOEA/D-DE. IFCS-MOEA is embedded with MOEA/D-DE (IFCS-MOEA/D-DE) and compared with the original MOEA/D-DE on UF1–10 and LZ1–9 test problems.

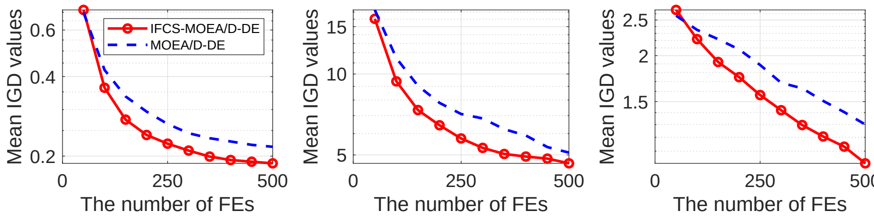
Table 3 shows the mean IGD values obtained by IFCS-MOEA/D-DE and MOEA/D-DE after 500 FEs on the 19 test problems. The Wilcoxon rank-sum test at the 5% significance level is used to evaluate the statistical difference between IFCS-MOEA/D-DE and MOEA/D-DE. In this table, "$+, -, \sim$" denote that the results obtained by MOEA/D-DE are better than, worse than, or similar to those obtained by IFCS-MOEA/D-DE, respectively. Table 3 shows that IFCS-MOEA/D-DE outperforms MOEA/D-DE on 16 test problems. On UF4, UF8, and LZ8, the two algorithms obtain similar results.

Figure 2 plots the mean IGD versus the number of FEs obtained by IFCS-MOEA/D-DE and MOEA/D-DE on the UF2, UF10, and LZ7 test problems. Figure 2 shows that IFCS-MOEA/D-DE converges faster and obtains better IGD values than MOEA/D-DE on these three test problems.

Based on the above results, we can conclude that IFCS-MOEA/D-DE is more efficient than MOEA/D-DE in solving these 19 MOPs with complicated PSs under a limited number of FEs.

4.3 Performance Comparison with the State-of-the-art MOEAs

This section compares the performance of IFCS-MOEA/D-DE with five state-of-the-art surrogate-based MOEAs: FCS-MOEA/D-DE, CPS-MOEA, CSEA, MOEA/D-EGO, and EDN-ARMOEA. These algorithms are compared on the UF1–10 and LZ1–9 test problems.



(a) UF2 (b) UF10 (c) LZ7

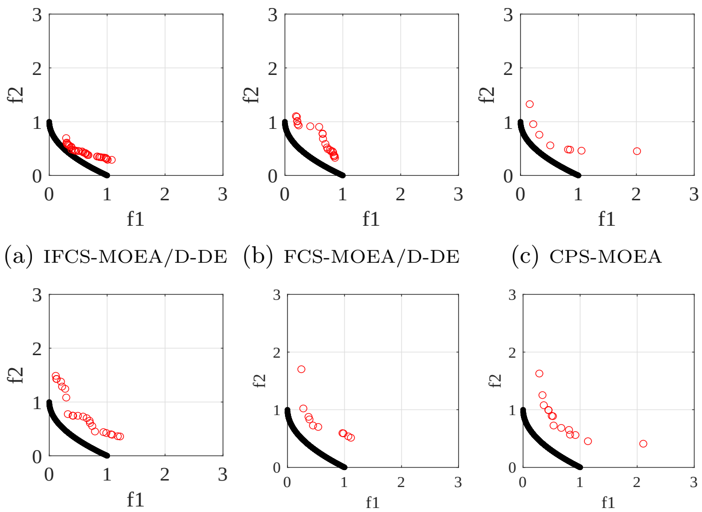
Fig. 2. The mean IGD values versus the number of FEs obtained by IFCS-MOEA/D-DE and MOEA/D-DE on UF2, UF10, and LZ7.

Table 4 presents the mean IGD values obtained by the six algorithms after 500 FEs on the 19 test problems. The Wilcoxon rank-sum test is used for statistical test. The best result on each test problem is shaded. At the bottom of the table, we summarize the number of problems on which the performance of the compared algorithm is better than, worse than, and similar to that of IFCS-MOEA/D-DE, respectively. In Table 4, IFCS-MOEA/D-DE outperforms FCS-MOEA/D-DE, CPS-MOEA, CSEA, MOEA/D-EGO, and EDN-ARMOEA on 9, 16, 11, 15, and 12 test problems, respectively. IFCS-MOEA/D-DE performs worse than FCS-MOEA/D-DE, CPS-MOEA, CSEA, MOEA/D-EGO, and EDN-ARMOEA on 6, 0, 6, 0, and 0 test problems, respectively.

Table 4. The *mean* IGD values of IFCS-MOEA/D-DE, FCS-MOEA/D-DE, CPS-MOEA, CSEA, MOEA/D-EGO, and EDN-ARMOEA on UF1–10 and LZ1–9.

	IFCS-MOEA/D-DE	FCS-MOEA/D-DE	CPS-MOEA	CSEA	MOEA/D-EGO	EDN-ARMOEA
UF1	8.87e-01$_{1.03e-01}$	5.73e-01$_{1.70e-01}$(+)	1.05e+00$_{1.56e-01}$(−)	5.98e-01$_{2.16e-01}$(+)	9.40e-01$_{1.68e-01}$(∼)	9.68e-01$_{1.69e-01}$(−)
UF2	1.88e-01$_{2.87e-02}$	3.04e-01$_{4.55e-02}$(−)	3.00e-01$_{3.73e-02}$(−)	3.26e-01$_{5.30e-02}$(−)	4.09e-01$_{5.39e-02}$(−)	4.12e-01$_{3.71e-02}$(−)
UF3	6.04e-01$_{2.57e-02}$	6.91e-01$_{5.89e-02}$(−)	7.30e-01$_{6.35e-02}$(−)	7.11e-01$_{7.26e-02}$(−)	7.63e-01$_{7.43e-02}$(−)	7.27e-01$_{5.08e-02}$(−)
UF4	1.32e-01$_{6.34e-03}$	1.70e-01$_{7.55e-03}$(−)	1.47e-01$_{6.80e-03}$(−)	1.59e-01$_{8.22e-03}$(−)	1.52e-01$_{8.43e-03}$(−)	1.72e-01$_{4.21e-03}$(−)
UF5	4.20e+00$_{3.67e-01}$	3.00e+00$_{4.19e-01}$(+)	4.49e+00$_{3.64e-01}$(−)	2.91e+00$_{5.98e-01}$(+)	4.90e+00$_{3.84e-01}$(−)	4.53e+00$_{3.80e-01}$(−)
UF6	3.78e+00$_{5.72e-01}$	2.48e+00$_{8.10e-01}$(+)	4.27e+00$_{6.09e-01}$(−)	1.79e+00$_{6.32e-01}$(+)	4.31e+00$_{8.82e-01}$(−)	4.03e+00$_{7.69e-01}$(∼)
UF7	9.50e-01$_{1.11e-01}$	6.03e-01$_{1.51e-01}$(+)	1.03e+00$_{1.75e-01}$(−)	4.21e-01$_{1.04e-01}$(+)	1.08e+00$_{1.88e-01}$(−)	1.06e+00$_{2.20e-01}$(∼)
UF8	7.24e-01$_{9.32e-02}$	1.35e+00$_{3.77e-01}$(−)	1.43e+00$_{2.03e-01}$(−)	1.40e+00$_{3.25e-01}$(−)	1.62e+00$_{3.32e-01}$(−)	1.93e+00$_{2.12e-01}$(−)
UF9	7.57e-01$_{8.23e-02}$	1.33e+00$_{2.04e-01}$(−)	1.38e+00$_{2.92e-01}$(−)	1.40e+00$_{2.99e-01}$(−)	1.87e+00$_{6.32e-01}$(−)	1.81e+00$_{2.13e-01}$(−)
UF10	4.65e+00$_{4.34e-01}$	7.13e+00$_{1.10e+00}$(−)	8.06e+00$_{1.20e+00}$(−)	7.93e+00$_{1.45e+00}$(−)	8.78e+00$_{1.40e+00}$(−)	9.72e+00$_{1.25e+00}$(−)
LZ1	1.55e-01$_{8.70e-03}$	1.59e-01$_{1.07e-02}$(∼)	1.52e-01$_{1.29e-02}$(∼)	1.62e-01$_{1.75e-02}$(∼)	1.75e-01$_{1.56e-02}$(−)	1.61e-01$_{1.41e-02}$(∼)
LZ2	8.99e-01$_{1.69e-01}$	5.54e-01$_{1.41e-01}$(+)	1.03e+00$_{1.31e-01}$(−)	4.92e-01$_{2.21e-01}$(+)	1.05e+00$_{1.90e-01}$(−)	9.98e-01$_{1.61e-01}$(∼)
LZ3	2.22e-01$_{1.45e-02}$	3.53e-01$_{4.84e-02}$(−)	3.49e-01$_{2.87e-02}$(−)	3.45e-01$_{6.22e-02}$(−)	4.46e-01$_{6.09e-02}$(−)	4.46e-01$_{3.44e-02}$(−)
LZ4	2.15e-01$_{2.08e-02}$	3.51e-01$_{5.51e-02}$(−)	3.40e-01$_{4.16e-02}$(−)	3.41e-01$_{4.85e-02}$(−)	4.24e-01$_{7.09e-02}$(−)	4.41e-01$_{4.02e-02}$(−)
LZ5	2.00e-01$_{2.53e-02}$	3.07e-01$_{4.22e-02}$(−)	3.10e-01$_{3.57e-02}$(−)	3.01e-01$_{4.34e-02}$(−)	4.20e-01$_{7.43e-02}$(−)	4.17e-01$_{4.52e-02}$(−)
LZ6	4.92e-01$_{4.56e-02}$	5.46e-01$_{1.50e-01}$(∼)	8.54e-01$_{2.24e-01}$(−)	6.32e-01$_{1.82e-01}$(−)	5.25e-01$_{8.00e-02}$(∼)	5.32e-01$_{1.07e-01}$(∼)
LZ7	1.02e+00$_{3.52e-01}$	8.12e-01$_{2.57e-01}$(∼)	1.52e+00$_{5.86e-01}$(−)	9.44e-01$_{2.66e-01}$(∼)	1.49e+00$_{5.47e-01}$(−)	1.50e+00$_{2.13e-01}$(−)
LZ8	8.33e-01$_{1.23e-01}$	7.54e-01$_{2.02e-01}$(∼)	8.82e-01$_{3.10e-01}$(∼)	9.71e-01$_{1.72e-01}$(−)	7.85e-01$_{2.73e-01}$(∼)	8.77e-01$_{8.86e-02}$(∼)
LZ9	9.52e-01$_{1.43e-01}$	5.78e-01$_{1.78e-01}$(+)	1.01e+00$_{1.37e-01}$(∼)	4.96e-01$_{1.57e-01}$(+)	9.46e-01$_{1.83e-01}$(∼)	9.50e-01$_{2.18e-01}$(∼)
+/ − / ∼		6/9/4	0/16/3	6/11/2	0/15/4	0/12/7

Figure 3 plots the non-dominated solutions obtained by IFCS-MOEA/D-DE, FCS-MOEA/D-DE, CPS-MOEA, CSEA, MOEA/D-EGO, and EDN-ARMOEA on UF2. For each algorithm, we choose a single run with the median IGD value over 21 runs. In this figure, the solutions obtained by each algorithm are shown by red circles and the PF is shown by a black curve. This figure shows that



(a) IFCS-MOEA/D-DE (b) FCS-MOEA/D-DE (c) CPS-MOEA

(d) CSEA (e) MOEA/D-EGO (f) EDN-ARMOEA

Fig. 3. The non-dominated solutions obtained by the five compared algorithms on UF2 with the median IGD value.

the solutions obtained by IFCS-MOEA/D-DE are closer to the PF than the solutions obtained by other five algorithms. The above results show that IFCS-MOEA/D-DE outperforms the five compared algorithms on most test problems. Therefore, we can conclude that IFCS-MOEA/D-DE is efficient in solving MOPs with complicated PSs.

4.4 Performance Comparison on Real-World Problems

This section compares the performance of IFCS-MOEA/D-DE and the five state-of-the-art MOEAs on four real-world MOPs [32]: reinforced concrete beam design problem (RCBD), pressure vessel design problem (PVD), coil compression spring design problem (CCSD), and gear train design problem (GTD). The first three MOPs have 2 objectives and the last one MOP have 3 objectives. Due to the page limit, readers can refer to [32] for the details of these real-world MOPs. In the experiments, the population size is $N = 45$. The maximal number of FEs is 500. Each algorithm executes 21 times on each test problem.

Table 5 shows the mean IGD values obtained by the five compared algorithms. The best result on each test problem is shaded. At the bottom of the table, we summarize the number of problems on which the performance of the compared algorithm is better than, worse than, and similar to that of IFCS-MOEA/D-DE, respectively. In Table 5, IFCS-MOEA/D-DE outperforms all the other algorithms on all test problems except for one case: there is no statistically significant difference between IFCS-MOEA/D-DE and EDN-ARMOEA on the GTD problem whereas the best average IGD value is obtained by IFCS-MOEA/D-DE. From the above results, we can conclude that the proposed IFCS-

Table 5. The *mean* IGD values of IFCS-MOEA/D-DE, FCS-MOEA/D-DE, CPS-MOEA, CSEA, MOEA/D-EGO, and EDN-ARMOEA on four real-world problems.

	IFCS-MOEA/D-DE	FCS-MOEA/D-DE	CPS-MOEA	CSEA	MOEA/D-EGO	EDN-ARMOEA
RCBD	1.26e-02$_{7.25e-03}$	3.09e-02$_{2.07e-02}$(−)	1.68e-02$_{3.09e-03}$(−)	2.80e-02$_{7.70e-03}$(−)	7.83e-02$_{3.14e-02}$(−)	1.89e-02$_{6.17e-03}$(−)
PVD	2.61e-02$_{8.53e-03}$	6.95e-02$_{6.66e-02}$(−)	1.27e-01$_{1.34e-01}$(−)	9.62e-02$_{3.75e-02}$(−)	1.16e-01$_{5.48e-02}$(−)	6.96e-02$_{3.52e-02}$(−)
CCSD	5.11e-03$_{3.80e-03}$	9.25e-02$_{9.78e-02}$(−)	3.74e-02$_{3.60e-02}$(−)	1.23e-01$_{7.78e-02}$(−)	1.78e-01$_{1.17e-01}$(−)	8.13e-02$_{8.42e-02}$(−)
GTD	5.15e-02$_{1.28e-02}$	1.23e-01$_{1.15e-01}$(−)	8.34e-02$_{3.41e-02}$(−)	1.56e-01$_{6.79e-02}$(−)	1.50e-01$_{6.07e-02}$(−)	9.17e-02$_{7.25e-02}$(∼)
+/ − / ∼		0/4/0	0/4/0	0/4/0	0/4/0	0/3/1

MOEA/D-DE algorithms outperforms the five state-of-the-art MOEAs in solving these real-world application problems under a limited number of FEs.

5 Conclusion

This paper proposed an improved fuzzy classifier-based multiobjective evolutionary algorithm framework (IFCS-MOEA) for expensive MOPs. The IFCS-MOEA framework was developed based on an improved fuzzy classifier-based surrogate model. The IFCS model is used to sort unevaluated solutions based on the membership degrees and the model accuracy. Then, the promising offspring solutions are selected for function evaluations based on the sorting results. All the evaluated solutions are used for fuzzy classifier construction.

The proposed IFCS-MOEA framework was embedded with MOEA/D-DE for examination. The Fuzzy-KNN was used as the fuzzy classifier. The 10-fold cross-validation method was used to validate the quality of the classifier. IFCS-MOEA/D-DE was compared with the original MOEA/D-DE. The experimental results validated the effectiveness of IFCS in improving the performance of MOEA/D-DE on solving expensive MOPs under a limited number of FEs. Then, IFCS-MOEA/D-DE was compared with five state-of-the-art MOEAs on 19 test problems and four real-world application problems. The experimental results showed that IFCS-MOEA/D-DE outperformed the other five MOEAs in solving these problems under a limited number of FEs.

This paper validated the effectiveness of the IFCS-MOEA framework by embedding it with MOEA/D-DE. It is a future research topic to examine the effectiveness of IFCS-MOEA by embedding it with other MOEAs. It is also interesting to examine the performance of IFCS-MOEA/D-DE on other MOPs with complicated PSs. This paper used 30% and 70% as the accuracy threshold values in the proposed sorting strategy according to our preliminary results. It is necessary to further examine these threshold values on more test problems.

Acknowledgements. This work was supported by National Natural Science Foundation of China (Grant No. 62106099, 61876075), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Enterpreneurial Teams (Grant No. 2017ZT07X386), The Stable Support Plan Program of Shenzhen Natural Science Fund (Grant No. 20200925174447003), Shenzhen Science and Technology Program (Grant No. KQTD2016112514355531).

References

1. Bandaru, S., Ng, A.H., Deb, K.: On the performance of classification algorithms for learning Pareto-dominance relations. In: Proceedings of 2014 IEEE Congress on Evolutionary Computation (CEC 2014). pp. 1139–1146. Beijing, China (Jul 2014)
2. Beume, N., Naujoks, B., Emmerich, M.: SMS-EMOA: Multiobjective selection based on dominated hypervolume. European Journal of Operational Research **181**(3), 1653–1669 (Sep 2007)
3. Chugh, T., Jin, Y., Miettinen, K., Hakanen, J., Sindhya, K.: A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **22**(1), 129–142 (Feb 2018)
4. Chugh, T., Sindhya, K., Hakanen, J., Miettinen, K.: A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms. Soft Computing **23**(9), 3137–3166 (Dec 2019)
5. Chugh, T., Sindhya, K., Miettinen, K., Jin, Y., Kratky, T., Makkonen, P.: Surrogate-assisted evolutionary multiobjective shape optimization of anair intake ventilation system. In: Proceedings of the 2017 IEEE Congresson Evolutionary Computation (CEC 2017). pp. 1541–1548. Donostia, Spain (Jul 2017)
6. Coello, C.A.C., Sierra, M.R.: A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In: Proceedings of Mexican International Conference on Artificial Intelligence (MICAI 2004). pp. 688–697. Mexico City, Mexico (Apr 2004)
7. Deb, K.: Multi-objective genetic algorithms: problem difficulties and construction of test problems. Evolutionary Computation **7**(3), 205–230 (Sep 1999)
8. Deb, K., Hussein, R., Roy, P.C., Toscano-Pulido, G.: A taxonomy for metamodeling frameworks for evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation **23**(1), 104–116 (Feb 2019)
9. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (Apr 2002)
10. Deb, K., Sinha, A., Kukkonen, S.: Multi-objective test problems, linkages, and evolutionary methodologies. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2006). pp. 1141–1148. New York, NY, United States (Jul 2006)
11. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable test problems for evolutionary multiobjective optimization. In: Abraham, A., Jain, L., Goldberg, R. (eds.) Evolutionary Multiobjective Optimization. pp. 105–145. Springer-Verlag, London, UK (2005)
12. Guo, D., Wang, X., Gao, K., Jin, Y., Ding, J., Chai, T.: Evolutionary optimization of high-dimensional multiobjective and many-objective expensive problems assisted by a dropout neural network. IEEE Transactions on Systems, Man, and Cybernetics: Systems **52**(4), 2084–2097 (Apr 2022)
13. Hillermeier, C.: Nonlinear multiobjective optimization: A generalized homotopy approach. Basel, Switzerland: Birkhauser-Verlag (2000)
14. Huband, S., Barone, L., While, L., Hingston, P.: A scalable multi-objective test problem toolkit. In: Proceedings of International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005). pp. 280–295. Guanajuato, Mexico (March 2005)

15. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**(5), 477–506 (Oct 2006)
16. Jia, L., Wang, Y., Fan, L.: Multiobjective bilevel optimization forproduction-distribution planning problems using hybrid genetic algorithm. Integrated Computer-Aided Engineering **21**(1), 77–90 (Jan 2014)
17. Jin, Y.: Surrogate-assisted evolutionary computation: Recent advances and future challenges. Swarm and Evolutionary Computation **1**(2), 61–70 (Jun 2011)
18. Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. Journal of Global Optimization **13**(4), 455–492 (Dec 1998)
19. Kasprzak, E.M., Lewis, K.E.: An approach to facilitate decision tradeoffs in pareto solution sets. Journal of Engineering Valuation and Cost Analysis **3**(1), 173–187 (Jan 2000)
20. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy K-nearest neighbor algorithm. IEEE Transactions on Systems, Man, and Cybernetics **SMC-15**(4), 580–585 (July-Aug 1985)
21. Knowles, J.: ParEGO: A hybrid algorithm with on-line landscape approximationfor expensive multiobjective optimization problems. IEEE Transactions on Evolutionary Computation **10**(1), 50–66 (Feb 2006)
22. Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In: Proceedings of International Conference on Parallel Problem Solving from Nature (PPSN 2006). pp. 583–592. Reykjavik, Iceland (Sep 2006)
23. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation **13**(2), 284–302 (Apr 2009)
24. Lin, Q., Wu, X., Ma, L., Li, J., Gong, M., Coello, C.A.C.: An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization. IEEE Transactions on Evolutionary Computation **26**(4), 631–645 (Aug 2022)
25. Lin, X., Zhang, Q., Kwong, S.: A decomposition based multiobjective evolutionary algorithm with classification. In: Proceedings of 2016 IEEE Congress on Evolutionary Computation (CEC 2016). pp. 3292–3299. Vancouver, Canada (Jul 2016)
26. Loshchilov, I., Schoenauer, M., Sebag, M.: A mono surrogate for multiobjective optimization. In: Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2010). pp. 471–478. Portland, USA (Jul 2010)
27. Murata, T., Ishibuchi, H.: MOGA: multi-objective genetic algorithms. In: Proceedings of 1995 IEEE Congress on Evolutionary Computation (CEC 1995). pp. 289–294. Perth, Australia (Nov 1995)
28. Okabe, T., Jin, Y., Olhofer, M., Sendhoff, B.: On test functions for evolutionary multi-objective optimization. In: Proceedings of International Conference on Parallel Problem Solving from Nature (PPSN 2004). pp. 792–802. Birmingham, UK (Sep 2004)
29. Pan, L., Cheng, H., Tian, Y., Wang, H., Zhang, X., Jin, Y.: A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **23**(1), 74–88 (Feb 2019)
30. Song, Z., Wang, H., He, C., Jin, Y.: A Kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization. IEEE Transactions on Evolutionary Computation **25**(6), 1013–1027 (Dec 2021)
31. Sonoda, T., Nakata, M.: Multiple classifiers-assisted evolutionary algorithm based on decomposition for high-dimensional multi-objective problems. IEEE Transactions on Evolutionary Computation (2022 (Early Access))

32. Tanabe, R., Ishibuchi, H.: An easy-to-use real-world multi-objective optimization problem suite. Applied Soft Computing **89**, 106078 (Apr 2020)
33. Tian, Y., Cheng, R., Zhang, X., Cheng, F., Jin, Y.: An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility. IEEE Transactions on Evolutionary Computation **22**(4), 609–622 (Aug 2018)
34. Tian, Y., Cheng, R., Zhang, X., Jin, Y.: PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]. IEEE Computational Intelligence Magazine **12**(4), 73–87 (Nov 2017)
35. Veldhuizen, D.A.V., Lamont, G.B.: Multiobjective evolutionaryalgorithm test suites. In: Proceedings of the 1999 ACM Symposium on Applied computing (SAC 99). pp. 351–357. San Antonio, Texas USA (Feb 1999)
36. Yuan, Y., Banzhaf, W.: Expensive multi-objective evolutionary optimization assisted by dominance prediction. IEEE Transactions on Evolutionary Computation **26**(1), 159–173 (Feb 2022)
37. Zhang, J., He, L., Ishibuchi, H.: Dual fuzzy classifier-based evolutionary algorithm for expensive multiobjective optimization. IEEE Transactions on Evolutionary Computation (2022 (Early Access))
38. Zhang, J., Ishibuchi, H.: Multiobjective optimization with fuzzy classification-assisted environmental selection. In: Proceedings of the 11th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2021). pp. 580–592. Shenzhen, China (Mar 2021)
39. Zhang, J., Ishibuchi, H., Shang, K., He, L., Pang, L.M., Peng, Y.: Environmental selection using a fuzzy classifier for multiobjective evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2021). pp. 485–492. Lille, France (Jun 2021)
40. Zhang, J., Zhou, A., Tang, K., Zhang, G.: Preselection via classification: A case study on evolutionary multiobjective optimization. Information Sciences **465**, 388–403 (Oct 2018)
41. Zhang, J., Zhou, A., Zhang, G.: A classification and Pareto domination based multiobjective evolutionary algorithm. In: Proceedings of 2015 IEEE Congress on Evolutionary Computation (CEC 2015). pp. 2883–2890. Sendai, Japan (May 2015)
42. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (Dec 2007)
43. Zhang, Q., Liu, W., Tsang, E., Virginas, B.: Expensive multiobjective optimization by MOEA/D with Gaussian process model. IEEE Transactions on Evolutionary Computation **14**(3), 456–474 (Jun 2009)
44. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC 2009 special session and competition. Techreport CES-487, The School of Computer Science and Electronic Engineering, University of Essex (2009)
45. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthanb, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation **1**(1), 32–49 (Mar 2011)
46. Zitzler, E., Kunzli, S.: Indicator-based selection in multiobjective search. In: Proceedings of International Conference on Parallel Problem Solving from Nature (PPSN 2004). pp. 832–842. Birmingham, UK (Sep 2004)
47. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm. Tech. Rep. 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland (2001)