Environmental Selection Using a Fuzzy Classifier for Multiobjective Evolutionary Algorithms

Jinyuan Zhang, Hisao Ishibuchi*, Ke Shang, Linjun He, Lie Meng Pang, and Yiming Peng
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, hisao@sustech.edu.cn, kshang@foxmail.com, this.helj@gmail.com, panglm@sustech.edu.cn, 11930669@mail.sustech.edu.cn

ABSTRACT
The quality of solutions in multiobjective evolutionary algorithms (MOEAs) is usually evaluated by objective functions. However, function evaluations (FEs) are usually time-consuming in real-world problems. A large number of FEs limit the application of MOEAs. In this paper, we propose a fuzzy classifier-based selection strategy to reduce the number of FEs of MOEAs. First, all evaluated solutions in previous generations are used to build a fuzzy classifier. Second, the built fuzzy classifier is used to predict each unevaluated solution’s label and its membership degree. The reproduction procedure is repeated to generate enough offspring solutions (classified as positive by the classifier). Next, unevaluated solutions are sorted based on their membership degrees in descending order. The same number of solutions as the population size is selected from the top of the sorted unevaluated solutions. Then, the best half of the chosen solutions are selected and stored in the new population. The other half solutions are evaluated. Finally, the evaluated solutions are used together with evaluated current solutions for environmental selection to form another half of the new population. The proposed strategy is integrated into two MOEAs. Our experimental results demonstrate the effectiveness of the proposed strategy on reducing FEs.

CCS CONCEPTS
• Theory of computation → Evolutionary algorithms; Bio-inspired optimization;

KEYWORDS
Multiobjective evolutionary optimization, fuzzy classifier, environmental selection, surrogate models

ACM Reference Format:

1 INTRODUCTION
Multiobjective optimization problems (MOPs) widely exist in real-world applications. Generally, an MOP can be defined as follows:

\[
\text{Minimize } \mathbf{f}(x) = (f_1(x), \ldots, f_m(x))^T, \\
\text{subject to } x \in \Omega \subseteq \mathbb{R}^n, 
\]

where \(x\) is an \(n\)-dimensional decision variable, \(f_i(x), i = 1, \ldots, m\) is the \(i\)-th objective function, and \(\Omega\) defines the decision space.

Generally, objectives in Eq. (1) conflict with each other. There does not exist a single optimal solution that can optimize all objectives simultaneously. However, MOPs have a set of solutions that can be treated with the same optimality. These solutions are named as Pareto optimal solutions. The set of Pareto optimal solutions are called Pareto set (PS). The image of PS in the objective space is called Pareto front (PF). Let \(x\) and \(y\) be two solutions of Eq. (1). If \(f_i(x) \leq f_i(y)\) for all \(i = 1, \ldots, m\) and \(\exists j \in \{1, \ldots, m\}\) such that \(f_j(x) < f_j(y)\), then \(x\) is said to dominate \(y\). If there does not exist any solution that dominates \(x\), then \(x\) is called a Pareto optimal solution.

A number of multiobjective evolutionary algorithms (MOEAs) have been proposed to solve MOPs [31]. Most MOEAs can be classified into three categories: (a) dominance-based MOEAs [10, 34], (b) indicator-based MOEAs [3, 33], and (c) decomposition-based MOEAs [19, 29]. Generally, MOEAs have the following three main components: (a) initialization, to initialize a set of solutions; (b) reproduction, to generate a set of offspring solutions; (c) environmental selection, to select a set of solutions for the new population.

Typical environmental selection strategies in MOEAs can be classified into three classes: objective function value-based [29], dominance relation-based [10], or fitness value-based [33]. In the above strategies, it is necessary to evaluate the objective function values of each solution. However, many evaluations are time-consuming for those expensive problems, which limits the application of MOEAs in solving real-world problems.

Based on the above considerations, many methods for reducing the number of function evaluations (FEs) are proposed. The most efficient methods on reducing FEs are surrogate model-based MOEAs (SAEAs) [6, 9, 13]. The SAEAs build computationally cheap surrogate models to approximate the original objective functions or fitness functions. Then, these approximated models are used to
replace the original functions for evaluation. In this manner, the FEs can be hopefully reduced.

Typically, environmental selection in MOEAs can be regarded as a classification problem. The selected solutions are promising class, the discarded solutions are unpromising class. In terms of this consideration, classifiers have been applied to MOEAs [17, 20, 27]. Classifier-based methods build global models to predict the relation between solutions [2], where SAEAs usually build each model to approximate each objective or fitness function.

In this paper, we propose a general fuzzy classifier-based selection (FCS) mechanism to reduce FEs of MOEAs. First, we use all previously evaluated solutions as the training dataset for fuzzy classifier building. Since there exists a dominance relation between evaluated solutions in MOEAs, i.e., non-dominated and dominated solutions, we use evaluated solutions in these two classes as the training dataset. Next, the built fuzzy classifier is used to predict the label and the membership degree of each unevaluated solution to the positive class. Then, these unevaluated solutions are sorted according to their membership degrees in descending order. The same number of solutions as the population size are selected from the sorted unevaluated solutions based on their membership degrees. Finally, the best half of the selected solutions are stored in the new population without evaluations. The worst half of the selected solutions are evaluated and used together with the evaluated current solutions for environmental selection to generate another half of solutions for the new population. As a result, a half of solutions in the next population are unevaluated solutions. Therefore, the number of FEs is reduced.

The remainder of this paper is organized as follows. In Section 2, we provide the preliminary knowledge of classifier-based MOEAs and fuzzy classifier. In Section 3, we propose the framework of fuzzy classifier-based MOEAs. We also present the training dataset definition strategy and the proposed fuzzy classifier-based selection strategy in Section 3. In Section 4, we integrate the proposed fuzzy classifier-based selection strategy into two MOEAs. The experimental results of the proposed strategy on test problems and a real-world problem are provided. In Section 5, we conclude this paper with some future research topics.

2 PRELIMINARY KNOWLEDGE

In this section, first, we review some related studies on classifier-based MOEAs in Section 2.1. Then, we present the fuzzy classifier used in this paper in Section 2.2.

2.1 Classifier-based MOEAs

As a special case of surrogate-based MOEAs, using a classifier to improve the efficiency of MOEAs has been proposed in the last decades. Loshchilov et al. [17] was the first one to use a classifier in MOEAs. In their approach, the classifier was combined with a regression model to build a surrogate model. Then, the built model was used to predict the dominance relation between a new solution and the current non-dominated solution set. Bandaru et al. [2] used a multi-class classifier to learn the Pareto dominance relation among solutions. Bhattacharjee et al. [4] used a support vector machine (SVM) to assist NSGA-II in reducing FEs. Zhang et al. [26–28] built a classifier to pre-select promising offspring solutions from a set of candidates for MOEAs. Lin et al. [16] used SVM to pre-select offspring solutions for FEs. Pan et al. [20] proposed a classifier-based surrogate-assisted evolutionary algorithm (CSEA) for expensive many-objective optimization problems. They used a classifier to predict the dominance relation between the offspring solution and some selected reference solutions. These classification-based algorithms have shown their efficiency in improving the performance of MOEAs.

However, previous classifier-based algorithms only use the predicted labels of solutions for selection. It is necessary to utilize more information from the classifier to ensure high accuracy of prediction. Besides a class label, fuzzy classifiers also assign a membership degree of a solution to the class. Our major purpose in this paper is to propose a general fuzzy classifier-based selection (FCS) strategy. Any fuzzy classifier can be integrated into the FCS strategy, and FCS can be applied to any kind of MOEAs.

2.2 Fuzzy Classifier

Fuzzy set theory (or fuzzy theory) [22, 24] has been proposed for using fuzzy concepts in a quantitative manner. Fuzzy set theory has been used to solve problems with imprecise information. In general, a membership function [23] is used to specify a membership degree of an object (e.g., an input vector, a new pattern) to a fuzzy concept.

Fuzzy classifiers are one of the most successful applications of fuzzy set theory [12]. Different from standard non-fuzzy classifiers which usually provide only the prediction about the class label of each new pattern, fuzzy classifiers also provide the membership degree of each new pattern to each class. In the past decades, many fuzzy classifiers have been proposed [12, 14].

In this paper, we consider using a Fuzzy-KNN (FKNN) classifier [14]. Different from traditional K-nearest neighbor (KNN) [8] which uses the Euclidean distance to partition the pattern space, FKNN uses fuzzy similarity in the classification inference process. The main steps of FKNN are as follows. First, the fuzzy similarity between a new pattern and each training pattern is calculated. Next, K nearest neighbors of the new pattern are selected from the training patterns based on the calculated similarity. Then, the membership degree of the new pattern is calculated based on its K nearest neighbors’ membership degrees. Finally, the class label and membership degree of the new pattern are obtained.

Fuzzy set theory has been applied to improve the performance of evolutionary algorithms (EAs) for single-objective optimization problems. Akbarzadeh-Totonchi et al. [1] proposed an adaptive fuzzy fitness granulation to reduce the number of FEs. The proposed method maintained a dataset to store all evaluated solutions (fuzzy granules). If the new solution was similar to a fuzzy granule, it inherited the fitness value of the similar granule. Only the solutions that were not similar to any granules were evaluated. However, this fitness granulation method leads to the following problem: relatively ‘bad’ fitness values tend to be maintained and assigned to new solutions. Zhou et al. [32] integrated a fuzzy classifier to preselect promising offspring solutions from a set of candidates. Zhang et al. [25] used a fuzzy classifier to improve the performance of EAs by filtering unpromising offspring solutions.
3 THE PROPOSED FUZZY CLASSIFIER-BASED MOEA

In this section, we propose a fuzzy classifier-based selection (FCS) strategy for MOEAs. First, the general framework of fuzzy classifier-based MOEAs is presented in Section 3.1. Next, the training dataset definition strategy is provided in Section 3.2. Then, the proposed fuzzy classifier-based selection (FCS) strategy is presented in Section 3.3.

3.1 Algorithm Framework

Our proposed fuzzy classifier-based MOEA has the following main components. First, we use all the evaluated solutions in previous generations as the training dataset for classifier building. These solutions are divided into two classes according to their dominance relation. The non-dominated solutions are the positive training patterns, and the dominated solutions are the negative training patterns. Next, we execute the reproduction procedure to generate offspring solutions and use the built classifier to predict the quality of each offspring solution. The reproduction procedure is repeated until enough positive offspring solutions are generated. Finally, a fuzzy classifier-based selection strategy is applied to solutions in the current population and the generated offspring solutions to form the new population for the next generation. The framework of the proposed fuzzy classifier-based MOEA is presented in Algorithm 1.

Algorithm 1: Framework of Fuzzy Classifier-based MOEA

1. Initialize the population $P = \{x^1, x^2, \ldots, x^N\}$, evaluate solutions in $P$;
2. Set $P_+ = \text{Non-dominated}_\text{Selection}(P)$ and $P_-= P \setminus P_+$;
3. while termination condition is not satisfied do
   4. Train a classifier $[l, m] = \text{fuzzy}\_\text{classifier}\_\text{training}(x)$ by using $P_+$ and $P_-$;
   5. Set $Q_p = \emptyset$;
   6. while $|Q_p| < N$ do
      7. Generate an offspring population $Q = \{y^1, \ldots, y^N\}$;
      8. for $i = 1: N$ do
         9. Predict the label and the membership degree of $y^i$ by $[l_{y^i}, m_{y^i}] = \text{fuzzy}\_\text{classifier}\_\text{prediction}(y^i)$;
         10. if $l_{y^i} = 1$ then
              11. $Q_p = Q_p \cup \{y^i\}$;
         12. end
      13. end
      14. end
   15. end
   16. $P_+, P_-, P = \text{FCS}(P, Q_p)$.

3.2 Training Dataset Definition

Since all the evaluated solutions in previous generations contain search information of MOEAs, these solutions are the best choice for model building. The dominance relation among these solutions can be used to define the two classes of the training dataset. The non-dominated solutions are the positive class, and the dominated solutions are the negative class.

Let $T$ be a dataset (i.e., all the evaluated solutions), $P_+$ be the positive training dataset, $P_-$ be the negative training dataset, and $\text{Non-dominated}_\text{Selection}$ be a Pareto optimal solution filter. Then, $P = \text{Non-dominated}_\text{Selection}(T)$ indicates the non-dominated solution set, and the two training datasets are generated as follows:

$$P_+ = \text{Non-dominated}_\text{Selection}(T),$$
and
$$P_- = T \setminus P_+.$$

3.3 Fuzzy Classifier-based Selection

Our proposed fuzzy classifier-based selection (FCS) strategy is presented in Algorithm 2. First, we use the built classifier to predict the label and the membership degree of each unevaluated current solution (Line 1–5). Second, we merge the unevaluated solutions and the positive offspring solutions (i.e., $Q_p$ obtained by Algorithm 1) into one population $Q$ (Line 6). Third, we sort these unevaluated solutions according to their membership degrees in descending order (Line 7). The top $N$ solutions where $N$ is the population size are chosen (Line 8). Next, the best half (with respect to the membership degrees) of the chosen $N$ solutions are selected and stored in population $Q_{\text{uneval}}$ (Line 13). The worst half solutions of the chosen $N$ solutions are evaluated (Line 14). Then, the evaluated solutions are used together with the evaluated current solutions for environmental selection to form a half of the new population (i.e., $P_2$ in Line 15). Finally, the new population is created by combining all solutions in $Q_{\text{uneval}}$ and $P_2$ (Line 16). The training dataset is updated based on all the evaluated solutions (Line 18–20). It should be noted that all solutions in the merged population are evaluated every five generations (Line 9–11) to maintain the high quality of the current population. In Line 9, $\text{Gen}$ indicates the number of generations. Since half of the solutions in the population are not evaluated in this strategy, the number of FEs can be reduced.

4 EXPERIMENTS

In this section, we study the effectiveness of the proposed fuzzy classifier-based selection (FCS) strategy. First, the experimental settings are shown in Section 4.1. Second, the efficiency of our proposed FCS strategy with different numbers of $K$ in FKNN are provided in Section 4.2. Third, the performance comparisons between the selection strategy with FKNN and KNN are shown in Section 4.3. Then, the FCS-based MOEA and other MOEAs are compared in Section 4.4. Finally, the FCS-based MOEA and a surrogate-based MOEA are compared on a real-world problem in Section 4.5.

4.1 Experimental Settings

(1) Baseline algorithms: Two dominance-based MOEAs, NSGA-II [10] and SPEA2 [34] are selected as baseline MOEAs for experiments. The classification-based preselection (CPS) strategy proposed in [26] is also used for comparison.

(2) Classifier: According to the presentation in Section 2.2, FKNN is used as the classifier.
Algorithm 2: \{P_+, P_-, P\} = FCS(P, Q_p)

1. \(P_{\text{eval}} = \{x \in P | x \text{ is evaluated}\}\)
2. \(P_{\text{uneval}} = \{x \in P | x \text{ is unevaluated}\}\)
3. for \(i = 1 : |P_{\text{uneval}}|\) do
   4. Predict the label and membership degree of \(x^i\) by \([l_{x^i}, m_{x^i}] = \text{fuzzy_classifier_prediction}(x^i)\)
5. end
6. \(Q = P_{\text{uneval}} \cup Q_p\)
7. Sort solutions in \(Q\) according to their membership degrees in descending order;
8. \(Q = \text{Select}(Q, N)\);
9. if \((\text{Gen} \mod 5) == 0\) then
   10. Evaluate all solutions in \(Q\) and store them in \(Q_{\text{eval}}\);
   11. \(P = \text{Environmental_Selection}(Q \cup P_{\text{eval}}, N)\);
12. else
   13. Select best half of solutions in \(Q\) and store them in \(Q_{\text{uneval}}\);
   14. \(Q_{\text{eval}} = Q \cup Q_{\text{uneval}}\) and evaluate solutions in \(Q_{\text{eval}}\);
   15. \(P_2 = \text{Environmental_Selection}(Q_{\text{eval}} \cup P_{\text{eval}}, N/2)\);
   16. \(P = Q_{\text{uneval}} \cup P_2\);
17. end
18. \(T = Q_{\text{eval}} \cup P_e \cup P_+;\)
19. \(P_+ = \text{Non-dominated_Selection}(T)\);
20. \(P_- = T \setminus P_+;\)
21. \(\text{Gen} = \text{Gen} + 1;\)

(3) Test instances: We choose the UF1–UF10, LZ1–LZ9 test problems [15, 30] for experiments. The number of objectives is \(M = 2\) for UF1–UF7, LZ1–LZ5, and LZ9, \(M = 3\) for UF8–UF10, and LZ6.

(4) Parameter settings: For each test problem, the number of decision variables is \(n = 30\) for UF1–UF7, LZ1–LZ5, and LZ9, \(n = 10\) for LZ6–LZ8; each algorithm executes for 21 times on each test problem independently. The population size is \(N = 50\). The maximum number of FEs is 500 since the problems are assumed to be computationally expensive. The other algorithm parameters are the same as in [10, 34]. All of the experiments are performed on the PlatEMO [22] platform.

(5) Performance metric: The inverted generational distance (IGD) [7] metric is used to evaluate the performance of the algorithms in the experiments.

4.2 Comparison of different values of K

In this section, we investigate the performance of our proposed FCS strategy with different values of \(K\) of the FKNN model. The \(K = 1, 3, 5, 7\) are used for experiments. The basic algorithm for experiments is NSGA-II. The four values of \(K\) are applied to FCS-NSGA-II and studied on UF1–UF10 and LZ1–LZ9 test problems.

Table 1 presents the mean \(\text{IGD}_{\text{avg}}\) IGD values obtained by FCS-NSGA-II with \(K = 1, 3, 5, 7\) after 500 FEs. The table provides the rank of each algorithm on each problem. The mean rank value of each algorithm on all test problems is also calculated. The results in Table 1 suggest that each algorithm can obtain the best IGD value. On most test problems, these four algorithms get similar IGD values. In terms of the mean rank value of each algorithm, \(K = 1\) is the best.

Therefore, we can conclude that FCS-NSGA-II with the above four values of \(K\) performs similarly on the above test problems. This situation means that the FCS strategy is not really sensitive to the value of \(K\). Based on the experimental results in Table 1, we use \(K = 1\) in our following experiments.

4.3 Comparison with KNN classifier

In this section, we investigate the efficiency of using FKNN by comparing it with the original KNN classifier. Since the KNN algorithm can only predict the label of a solution, in this experiment, the procedures of Line 7-8 in Algorithm 2 is designed as a random procedure. \(N\) solutions are randomly selected from \(Q\). Then, we apply the FKNN and KNN to NSGA-II for experiments, the value of \(K\) is 1. The two compared algorithms are denoted as FKNN-NSGA-II and KNN-NSGA-II.

Figure 1 presents the mean IGD values versus FEs of FKNN-NSGA-II, KNN-NSGA-II, and NSGA-II on UF1, UF9, LZ3, and LZ5. The figure suggests that FKNN-NSGA-II converges faster than KNN-NSGA-II and NSGA-II on these four test problems. KNN-NSGA-II is the second best. Figure 2 shows the solutions obtained by FCS-NSGA-II, KNN-NSGA-II, and NSGA-II on UF6, UF7, LZ4, and LZ9 in the execution with the median IGD value. The figures suggest solutions obtained by FCS-NSGA-II are closer to the Pareto front than KNN-NSGA-II and NSGA-II.

The above results clearly suggest that the fuzzy KNN-based strategy outperforms the KNN-based strategy. We can also conclude that with the assistance of the membership function, the performance of the fuzzy classifier is better than only using labels.

4.4 Comparison with other classifier-based MOEAs

This section dedicates to investigate the effectiveness of the proposed FCS strategy. The proposed FCS strategy is integrated into SPEA2 and NSGA-II for experiments. To validate the efficiency of the FCS, the CPS strategy is chosen and applied to SPEA2, NSGA-II for comparison. The comparison results of FCS-SPEA2, CPS-SPEA2, SPEA2, FCS-NSGA-II, CPS-NSGA-II, and NSGA-II are provided in Table 2.

Table 2 shows the mean, and std (standard deviation) IGD values obtained by the above mentioned six algorithms after 500 FEs on 19 test problems. The Wilcoxon rank-sum test is employed for results comparison. ‘+’, ‘-’ denotes the algorithm is better than, worse than, or similar to the FCS-MOEAs at the 95% significance level. The results suggest that FCS-SPEA2 gets better IGD values than CPS-SPEA2 on 11 test problems. The two algorithms perform similarly on UF4, UF8–UF10, LZ1, LZ2, LZ5, and LZ6 test problems. FCS-SPEA2 outperforms SPEA2 on 17 test problems. The two algorithms perform similarly on UF2 and UF4. FCS-NSGA-II outperforms FCS-SPEA2 on LZ8, and CPS-NSGA-II outperforms SPEA2 on UF4. On the other 17 test problems, the two algorithms obtain similar results. FCS-NSGA-II obtains better results than NSGA-II.
on 14 test problems. NSGA-II performs better on UF4. On the other 4 test problems, the two algorithms get similar IGD values.

Figure 3 presents the statistical results of mean IGD values versus FEs of FCS-SPEA2, CPS-SPEA2, and SPEA2 on UF2, UF8, LZ2, and LZ6. The figures suggest FCS-SPEA2 converges faster than CPS-SPEA2 and SPEA2 on UF2, UF8, and LZ2. On LZ6, FCS-SPEA2 and CPS-SPEA2 perform similarly. Figure 4 presents the statistical

### Table 1: Statistical mean [rank] IGD values obtained by FCS-NSGA-II with four values of $K$ on UF1–UF10, and LZ1–LZ9 after 500 FEs over 21 runs.

<table>
<thead>
<tr>
<th>K</th>
<th>Mean IGD Value</th>
<th>Mean IGD Value</th>
<th>Mean IGD Value</th>
<th>Mean IGD Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2.92e-01, 1.34e-00 [1]</td>
<td>2.94e-01, 1.54e-02 [2]</td>
<td>2.97e-01, 1.54e-02 [3]</td>
<td>2.94e-01, 1.37e-02 [4]</td>
</tr>
</tbody>
</table>

![Figure 1](image1.png)  
![Figure 2](image2.png)
Table 2: Statistical mean$_{std}$ IGD values of FCS-SPEA2, CPS-SPEA2 and SPEA2, FCS-NSGA-II, CPS-NSGAl and NSGA-II on UF1–UF10, LZ1–LZ9 after 500 FEs over 21 runs.

<table>
<thead>
<tr>
<th></th>
<th>FCS-SPEA2</th>
<th>CPS-SPEA2</th>
<th>SPEA2</th>
<th>FCS-NSGA-II</th>
<th>CPS-NSGAl</th>
<th>NSGA-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>UF1</td>
<td>5.32e-01</td>
<td>6.31e-01</td>
<td>8.32e-01</td>
<td>5.60e-01</td>
<td>7.29e-01</td>
<td></td>
</tr>
<tr>
<td>UF2</td>
<td>2.69e-01</td>
<td>3.10e-01</td>
<td>2.83e-01</td>
<td>2.92e-01</td>
<td>3.99e-01</td>
<td></td>
</tr>
<tr>
<td>UF3</td>
<td>6.70e-01</td>
<td>7.36e-01</td>
<td>7.85e-01</td>
<td>7.01e-01</td>
<td>7.68e-01</td>
<td></td>
</tr>
<tr>
<td>UF4</td>
<td>1.68e-01</td>
<td>1.66e-01</td>
<td>1.74e-01</td>
<td>1.66e-01</td>
<td>1.68e-01</td>
<td></td>
</tr>
<tr>
<td>UF5</td>
<td>3.12e-01</td>
<td>3.47e-01</td>
<td>3.25e-01</td>
<td>3.51e-01</td>
<td>3.70e-01</td>
<td></td>
</tr>
<tr>
<td>UF6</td>
<td>2.19e-01</td>
<td>2.77e-01</td>
<td>2.48e-01</td>
<td>2.58e-01</td>
<td>3.07e-01</td>
<td></td>
</tr>
<tr>
<td>UF7</td>
<td>5.84e-01</td>
<td>7.51e-01</td>
<td>7.92e-01</td>
<td>6.40e-01</td>
<td>8.10e-01</td>
<td></td>
</tr>
<tr>
<td>UF8</td>
<td>1.35e-01</td>
<td>1.52e-01</td>
<td>1.71e-01</td>
<td>1.52e-01</td>
<td>1.77e-01</td>
<td></td>
</tr>
<tr>
<td>UF9</td>
<td>1.43e-01</td>
<td>1.53e-01</td>
<td>1.67e-01</td>
<td>1.62e-01</td>
<td>1.75e-01</td>
<td></td>
</tr>
<tr>
<td>UF10</td>
<td>7.70e-01</td>
<td>8.29e-01</td>
<td>9.17e-01</td>
<td>8.59e-01</td>
<td>9.59e-01</td>
<td></td>
</tr>
<tr>
<td>LZ1</td>
<td>1.50e-01</td>
<td>1.49e-01</td>
<td>1.53e-01</td>
<td>1.49e-01</td>
<td>1.63e-01</td>
<td></td>
</tr>
<tr>
<td>LZ2</td>
<td>5.24e-01</td>
<td>6.04e-01</td>
<td>5.80e-01</td>
<td>6.45e-01</td>
<td>7.03e-01</td>
<td></td>
</tr>
<tr>
<td>LZ3</td>
<td>2.95e-01</td>
<td>3.73e-01</td>
<td>3.08e-01</td>
<td>3.19e-01</td>
<td>3.64e-01</td>
<td></td>
</tr>
<tr>
<td>LZ4</td>
<td>2.83e-01</td>
<td>3.35e-01</td>
<td>3.09e-01</td>
<td>3.21e-01</td>
<td>3.65e-01</td>
<td></td>
</tr>
<tr>
<td>LZ5</td>
<td>2.89e-01</td>
<td>3.10e-01</td>
<td>3.01e-01</td>
<td>2.89e-01</td>
<td>3.54e-01</td>
<td></td>
</tr>
<tr>
<td>LZ6</td>
<td>6.08e-01</td>
<td>5.99e-01</td>
<td>7.05e-01</td>
<td>6.45e-01</td>
<td>7.13e-01</td>
<td></td>
</tr>
<tr>
<td>LZ7</td>
<td>7.13e-01</td>
<td>8.92e-01</td>
<td>9.02e-01</td>
<td>9.04e-01</td>
<td>1.10e-00</td>
<td></td>
</tr>
<tr>
<td>LZ8</td>
<td>7.48e-01</td>
<td>9.21e-01</td>
<td>7.50e-01</td>
<td>8.52e-01</td>
<td>1.01e-00</td>
<td></td>
</tr>
<tr>
<td>LZ9</td>
<td>5.10e-01</td>
<td>6.47e-01</td>
<td>5.83e-01</td>
<td>5.98e-01</td>
<td>7.16e-01</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: The mean IGD values versus FEs obtained by FCS-NSGA-II, CPS-NSGA-II, and NSGA-II on UF2, UF8, LZ2, and LZ6.

The results of mean IGD values versus FEs of FCS-NSGA-II, CPS-NSGA-II, and NSGA-II on UF4, UF5, LZ3, and LZ8. The figures show FCS-NSGA-II outperforms CPS-NSGA-II and NSGAII on UF5, LZ3, and LZ8. The figures show FCS-NSGA-II on UF4, CPS-NSGA-II, and NSGA-II on UF4, UF5, LZ3, and LZ8. The figures show FCS-NSGA-II outperforms CPS-NSGA-II and NSGA-II on UF4, UF5, LZ3, and LZ8. The figures show FCS-NSGA-II on UF4, UF5, LZ3, and LZ8. The figures show FCS-NSGA-II outperforms CPS-NSGA-II and NSGA-II on UF4, UF5, LZ3, and LZ8.
4.5 Results on the Real-world Problem

In this section, we investigate the performance of the proposed FCS strategy on a real-world gear train design problem [11, 21]. The gear train design problem contains three objectives. The first objective is to minimize the error between the realized gear ratio and the required gear ratio. The second objective is to minimize the maximum size of four gears. The third objective is for the constraint violation. The problem has four variables, each variable represents the number of teeth in each gear. The details of the gear train design problem can be found in [11]. The proposed FCS strategy is integrated into NSGA-II for experiments. The Kriging-assisted reference vector guided evolutionary algorithm (K-RVEA) [5] and the original NSGA-II are employed for comparison.

Figure 5: The mean IGD values versus FEs obtained by FCS-NSGA-II, K-RVEA, and NSGA-II on the gear train design problem.

Figure 5 presents the mean IGD values versus FEs of FCS-NSGA-II, K-RVEA, and NSGA-II on the gear train design problem. The three algorithms use the Latin hypercube sampling [18] method for initialization, following the initialization procedure of K-RVEA. The population size is N = 43. The number of decision variables is n = 4. The maximal FEs of each algorithm is 500. Each algorithm executes for 21 times on the problem independently. The experimental results suggest our proposed FCS strategy can improve the performance of the original NSGA-II algorithm on the gear train design problem with limited evaluations. The FCS-NSGA-II performs better than K-RVEA in solving the gear train design problem.

5 CONCLUSION

In this paper, we proposed a fuzzy classifier-based selection (FCS) strategy for multiobjective evolutionary algorithms (MOEAs) to reduce the number of function evaluations (FEs). First, the proposed strategy uses all evaluated solutions in previous generations to build a fuzzy classifier. The built fuzzy classifier is used to predict the label and the membership degree of each unevaluated solution. Second, the reproduction procedure is repeated to generate enough positive offspring solutions without evaluations. Next, the unevaluated solutions are sorted based on their membership degrees in descending order. The same number of solutions as the population size are selected from the top of the sorted unevaluated solutions. Then, the best half of the chosen solutions are selected and stored in the new population without evaluations. The worst half of the chosen solutions are evaluated. Finally, the evaluated solutions are used together with the evaluated solutions in the current population for environmental selection to form another half of solutions for the new population. Since the population always contains half unevaluated solutions, the number of FEs can be reduced.

The proposed strategy was integrated into two Pareto dominance-based algorithms: NSGA-II and SPEA2 for experiments. The FKNN was chosen as the classifier. First, FCS-NSGA-II with four values of neighborhood size K was examined. The results suggested that FCS-NSGA-II with four values of K performed similarly, among them K = 1 was the best. Next, FCS-NSGA-II was compared with the strategy using the original KNN classifier. The comparison results indicated that the FKNN outperformed KNN with the assistance of the membership degree. Then, we applied FCS to SPEA2 and NSGA-II for comparison. The FCS based algorithms were compared with other classifier-based algorithms and original algorithms. The experimental results suggested the efficiency of the proposed strategy on reducing FEs. Finally, the FCS-NSGA-II was compared with a surrogate-based MOEA (K-RVEA) and NSGA-II on a real-world gear train design problem. The experimental results suggested FCS-NSGA-II outperformed K-RVEA and NSGA-II on the gear train design problem with limited number of evaluations.

The efficiency of the proposed FCS strategy was validated in this paper, however, some research issues should be addressed in the future. The proposed strategy was only integrated into two Pareto dominance-based algorithms. It can be applied to other kinds of MOEAs. The proposed strategy should be tested on other kinds of test problems, especially, on real-world problems.

ACKNOWLEDGEMENT

This work was supported by National Natural Science Foundation of China (Grant No. 61876075), Guangdong Provincial Key Laboratory (Grant No. 2020B121201001), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), Shenzhen Science and Technology Program (Grant No. KQTD201611251435531).

REFERENCES


