Evolutionary Algorithm with Ensemble Classifier Surrogate Model for Expensive Multiobjective Optimization

LAN Tian[*]

College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R.China

(Received 14 June 2020; revised 6 September 2020; accepted 10 September 2020)

Abstract: For many real-world multiobjective optimization problems, the evaluations of the objective functions are computationally expensive. Such problems are usually called expensive multiobjective optimization problems (EMOPs). One type of feasible approaches for EMOPs is to introduce the computationally efficient surrogates for reducing the number of function evaluations. Inspired from ensemble learning, this paper proposes a multiobjective evolutionary algorithm with an ensemble classifier (MOEA-EC) for EMOPs. More specifically, multiple decision tree models are used as an ensemble classifier for the pre-selection, which is be more helpful for further reducing the function evaluations of the solutions than using single inaccurate model. The extensive experimental studies have been conducted to verify the efficiency of MOEA-EC by comparing it with several advanced multiobjective expensive optimization algorithms. The experimental results show that MOEA-EC outperforms the compared algorithms.

Key words: multiobjective evolutionary algorithm; expensive multiobjective optimization; ensemble classifier; surrogate model

CLC number: TP18 **Document code:** A **Article ID:** 1005-1120(2020)S-0076-12

0 Introduction

A real-world optimization problem usually contains multiple conflicting objectives to be optimized simultaneously. Such problems are known as the multiobjective optimization problems (MOPs) [1]. Different from single-objective optimization problem that has a single optimal solution, an MOP has a set of trade-of Pareto-optimal solutions, called the Pareto set (PS). And its projection on the objective space is called Pareto front (PF)[2].

Over the recent decades, multiobjective evolutionary algorithms (MOEAs) have been recognized as a major methodology for approximating PF. Based on the selection methods, MOEAs can be roughly classified into three categories.

(1) Dominance-based MOEAs: This approach uses the Pareto domination or its variants and some other strategies to differentiate and order solutions, such as NSGA-II[3] and SPEA2[4].

(2) Decomposition-based MOEAs: This approach decomposes an MOP into a set of sub-problems and tackles these sub-problems simultaneously. The offspring reproduction and environmental selection are based on the sub-problems, such as MOEA/D[5] and CDG-MOEA[6].

(3) Indicator-based MOEAs: This approach is a set based selection strategy. The performance indicators are utilized to measure the quality of some sets of solutions, and set with the best quality value will be selected, such as hypervolume[7] and IGD[8].

Due to the population-based nature, a great number of function evaluations (FEs) is usually required for MOEAs to well-approximate PFs. Even worse, for many real-world optimization problems, e.g., airfoil design[9], drug design[10] and trauma system design[11], the explicit formulations of the objective functions or the constraint functions are not

*Corresponding author, E-mail address: lantian@nuaa.edu.cn.

How to cite this article: LAN Tian. Evolutionary algorithm with ensemble classifier surrogate model for expensive multiobjective optimization[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(S): 76-87.

http://dx.doi.org/10.16356/j.1005-1120.2020.S.010

known or the function evaluation could be expensive financially or computationally. It is very desirable to reduce the number of FEs without deteriorating the algorithm performance significantly. Such MOPs are usually referred to as computationally expensive multiobjective optimization problems (EMOPs).

One type of feasible approaches for EMOPs is to introduce the computationally efficient surrogates for reducing the number of FEs. Over the recent years, various types of surrogates have been adopted in EMOPs, including radial basis function[12], artificial neural networks[13] and Gaussian process model, which is also known as Kriging model[14], or sometimes as efficient global optimization (EGO). A variety of multiobjective surrogate-assisted evolutionary algorithms (SAEAs) has been proposed to handle EMOPs over the past decades, which has been surveyed in Ref. [15]. Recently, more and more SAEAs focus on solving different types of EMOPs, such as constraints handling methods[16-17], many-objective expensive optimization problems[18-20], and large-scale expensive optimization problems[21-22]. Moreover, many advanced machine learning technologies have been introduced into SAEAs to improve algorithm performance. Wang et al.[17] proposed using random forests and radial basis function networks as surrogates to approximate both objective and constraint functions. Min et al.[23] proposed an adaptive knowledge reuse framework based on the novel idea of multi-problem surrogates. Lyu et al.[24] proposed an ensemble-based model management strategy for surrogate-assisted EA.

Different from the previous work, the surrogate in this paper is composed of multiple surrogate models rather than one single model, and we also consider the uncertainty of model in a novel way. Based on the above idea, we propose an MOEA with the ensemble classifiers (MOEA-EC) for EMOPs. More specifically, multiple decision tree models are used as the ensemble classifier to choose promising solutions from the candidate offspring for real FEs. In this way, the number of real FEs can be greatly reduced for EMOPs.

1 Background Introduction

This section, we describes some relevant background with regard to multiobjective optimization and the classification model used in this work.

1.1 Multi-objective problem

This paper only considers continuous multiobjective optimization problems (MOPs), which can be defined as follows

$$\begin{aligned}&\text{Minimize } \boldsymbol{F}(x)=(f_1(x),\cdots,f_m(x))^{\mathrm{T}}\\&\text{Subject to } x\in\Omega\end{aligned}\tag{1}$$

where Ω is the decision space and $\boldsymbol{F}:\Omega\rightarrow\mathbf{R}^m$ consists of m real-valued objective functions. $\{\boldsymbol{F}(x)|x\in\Omega\}$ is the attainable objective set.

Usually, the objective functions $(f_1(x),f_2(x),\cdots,f_m(x))$ cannot be minimized simultaneously by a single solution in Ω. For these cases, we use Pareto optimal solutions to represent a set of best trade-off solutions among all objective functions.

Let $u,v\in\mathbf{R}^m$, and u is asked to dominate v, denoted by $u\prec v$, if and only if $u_j\leqslant v_j$ for every $j\in\{1,\cdots,m\}$ and $u_k<v_k$ for at least one index $k\in\{1,\cdots,m\}$. Given a set S in $\mathbf{R}^m$, a solution $x\in S$ is called nondominated if no other solution in S dominates it. A solution $x^*\in\Omega$ is Pareto-optimal if $\boldsymbol{F}(x^*)$ is nondominated in the attainable objective set. $\boldsymbol{F}(x^*)$ is then called a Pareto-optimal (objective) vector. In other words, any improvement in one objective of a Pareto optimal solution is bound to deteriorate at least another objective.

1.2 Classification and regression tree (CART) for SAEA

Decision tree is a reverse tree-like software model used for machine learning classification and numeric prediction. A prominent advantage of decision trees over other classification methods is that the tree-like structure allows for ease of data interpretation and analysis. Classification and regression tree (CART) is a nonparametric decision tree learning method, which can be used to do both classification and regression by using a decision tree[25]. In the case of classification, each interior node of the decision tree is an element of the feature vector, and each leaf node denotes a category label. A path from the root node to a leaf node represents a decision rule. The Gini[26] impurity is usually used to train the decision tree.

According to the idea of ensemble learning[27], this work investigates how to incorporate the CART model into the framework of NSGA-II, and propose a multiobjective evolutionary algorithm with an ensemble classifier (MOEA-EC) for expensive surrogate-assisted optimization problems. The main contributions of this work can be summarized as follow.

MOEA-EC divides the dominated solutions, which are much larger than nondominated solutions, into sets of equal size to nondominated solutions. This method solves the problem of imbalance data in a simple and effective way and does not waste the information contained in the solutions which evaluated in an expensive process in past generations.

MOEA-EC trains multiple models to predict the dominance relationship rather than single model. Because of the few true evaluation times, model would not be very accurate. To solve this problem, we combine multiple models into one ensemble classifier, and obtain good performance. In addition, the number of models in MOEA-EC is self-adaptive, which is a very valuable advantage in engineering.

Very few algorithms consider uncertainty of model, but the uncertainty plays an important role in evolution process. MOEA-EC treats the accuracy of models as the weight of models, then combines the weights of predicted value of multiple models into the final result.

2 MOEA-EC Algorithm Design

In this section, the main framework of MOEA-EC and the detail of algorithm will be presented.

2.1 Main framework

The pseudo code of MOEA-EC is presented in Algorithm 1, which includes eight steps as follows.

Algorithm 1 MOEA-EC

Input: an MOP;
a stopping criterion;
N: the population size of P;

Output: A solution set P;

1 $P \leftarrow$ Initialize the population with $11d-1$ solutions using Latin hypercube sampling method;
2 $Arc = P$;
3 **while** Not Stopping Criterion **do**
4 $[R, S] =$ DataHandle(Arc);
5 [models, acc] $=$ ModelBuild(R, S);
6 Generate an offspring solution set Q from P;
7 $Q' =$ ModelSelect (models, acc, Q);
8 Evaluate solutions in Q';
9 $Arc = Arc \cup Q'$;
10 Choose P from P and Q' using fast nondominated sorting and crowded distance;
11 **end**
12 **return** P

Initialization (Lines 1 to 2) An initial population P with $11d-1$ solutions is generated using Latin hypercube sampling, where d is the number of decision variables. In the initialization, the number of solutions to be evaluated using the expensive objective function equals $11d-1$, and these solutions are copied to archive Arc.

Data handling (Line 4) Solutions in archive Arc are divided according to dominance relationship into two categories. Then we divide the dominated solutions into multiple sets approximately equal size to nondominated solution to solve data imbalance problem.

Model building and training (Line 5) According to the number of training data sets, we first build several CART models. Then train the models with the corresponding data and obtain the accuracy of each model.

Reproduction (Line 6) For each solution $x \in P$, another two solutions are selected from P randomly. Then a new offspring solution is generated from the thee solutions by DE operators and added to Q.

Model predicting and selection (Line 7)

Predict the solution from Q with multiple models and choose promising solutions from Q according to prediction result. Then add these promising solutions to Q'.

True evaluation (Lines 8 to 9) Evaluate the population Q' using real evaluation functions, then add the population to archive Arc.

Environmental selection (Line 10) Environmental selection is performed to select N solutions from the population P and Q' to be the parent individuals of the next generation.

Repeat Steps 2—7 until meeting the stopping criterion, that is, the maximum number of FEs.

In MOEA-EC, the data handling and ensemble classifier are core parts. We will present the detail in the following subsections.

2.2 Data handling

In MOEA-EC, the data handling process described in Algorithm 2 is adopted to deal with the data imbalance problem. To implement MOEA-EC, we introduce an archive to store all solutions evaluated by the expensive fitness function. In this step, we do fast nondominated sorting on all solutions in to get the nondominated solution set and the dominated solution set, which described as lines 1—2 in Algorithm 2.

Algorithm 2 DataHandle(Arc)
Input: Arc: the archive set;
Output: R: A solution set P;
S ': the dominated solution sets;
1 R = nondominatedSort (Arc);
2 $S = Arc \backslash R$;
3 $K = \lceil |S|/|R| \rceil$;
4 $S' \leftarrow$ divide S int K equal parts $[S_1, \cdots, S_k]$;
5 **return** R, S'

Throughout the evolutionary process, the dominated solutions are much more than nondominated solutions, so the offspring solutions choosen by many classification based surrogate models are dominated and valuable function evaluations was wasted. In MOEA-EC, we first calculate the multiple relationship between dominated solutions and nondominated solutions, then we divide the dominated solutions into K sets S' which are approximately equal size to nondominated solution set R. Then we can use S and S' to build and train our models.

2.3 Building ensemble classifier

Choosing a suitable classification model is also very important. A good classification model should be able to catch the characteristic of complicated Pareto sets with different shapes in the decision space. We use CART to construct surrogate models for each objective in Eq.(1). CART is a nonparametric model and has a good performance in classification problem with small sample, so the model settings are easily to set.

Firstly, for each subset of S, we combine the nondominated solution set R and the subset into trainData. Then, we assign label "1" to each solution in R and label "0" to each solution in the subset. After combining the labels into trainLabel, we build a CART model and train it with the trainData and trainLabel. The accuracies of each model are calculated using all the solutions in R and S to approximate the global representations of each model. These processes are presented in Algorithm 3.

Algorithm 3 ModelBuild(R, S)
Input: R: A solution set P;
S: The dominated solution sets;
Output: models: The CART models set;
acc: A set of accuracies of each model;
$K = |S|$;
for $j = 1 \rightarrow K$ **do**
  trainData = $R \cup S_j$;
  for each $x \in R$ **do**
    add label 1 into trainLabel;
  end
  for each $x \in S_j$ **do**
    add label 0 into trainLabel;
  end
  models$_j$ = train(trainData, trainLabel);
  acc$_j$ = validation(models$_j$, $R \cup S$);
end
return models, acc

At the end of this step, we obtain K CART models as an ensemble classifier and the accuracies of these models. Then we will choose some promising solutions with the help of the ensemble classifier.

2.4 Prediction and selection

Once suitable classification model is built, we can use it to classify each new generated solution in Q during the evolutionary process, and discard all unpromising solutions without expensive FE.

The MOEA-EC model selection method is shown in Algorithm 4. For each solution in offspring population Q, we use each model in ensemble classifier to predict it once. Because of the limited number of true FEs in past generations, the models are not very accurate. To solve this problem, we consider the accuracies of each model as the weight of them. Therefore, the predicted label of each model is multiplied by the weight of the corresponding model and the predicted label of the corresponding model. Then we add up all the predicted labels to get the final predicted result *pre* of each solution. After rounding the predicted result *pre* up, we choose solutions with the largest label become promising solutions, and add them to Q'.

Algorithm 4 ModelSelect(R, S)

Input: models: the CART models set;

acc: a set of accuracies of each model;

Q: offspring solutions;

Output: Q': the solutions selected by ensemble classifier

$Q' \leftarrow \varnothing$;

for each $x \in Q$ **do**

  $Pre = 0$;

  for $j = 1 \rightarrow |\text{models}|$ **do**

    $\text{pre} = \text{pre} + \text{acc}_j \times$ predict(models_j, x);

  end

  /* calculate the integrated label for solution x */

  $\text{label}_x = \lceil pre \rceil$;

end

$Q' \leftarrow$ select solutions with max label from Q;

return Q'

3 Experimental Studies and Discussion

In this section, we first compare the performance of the original NSGA-II and two expensive MOEAs (i.e. MOEA/D-EGO[28] and CSEA[19]) on DTLZ[28] problems and ZDT[3] problems. Then, the influence of the ensemble classifier on the performance of MOEA-EC is investigated. All compared algorithm are implemented in PlatEMO[29].

3.1 Parameters settings

To make a fair comparison, the following parameters of all the compared algorithms are set in our experimental studies.

Dimension of objective space: $M = 3$ for DTLZ test instances and $M = 2$ for DTLZ test instances;

Dimension of decision space: $d = 10$ for all test instances;

Population size N is 50 for all test instance;

Maximal number of FEs is 300, including the initial evaluations;

For MOEA/D-EGO, the number of surrogate assisted fitness evaluations before updating the models is set to $2*(11d-1)$, and the other parameter settings can be found in Ref.[30];

For CSEA, number of solutions evaluated by surrogate model is $g_{\max}=3\ 000$ and the number of reference solutions is $k = 6$; Detailed parameter setting for ParEGO can be found in Ref.[19];

In MOEA-EC, the number of classifiers is dynamic and adaptive instead of a fixed value;

For MOEA-EC, we use nonparametric CART learning model, so the default settings in Ref.[31] are set. The other parameter settings are set as NSGA-II in Ref.[3].

3.2 Performance metrics

Inverted generational distance (IGD)[32]: It measures the average distance from a set of reference points P^* in the PF to the approximation set P. It can be formulated as follows

$$\text{IGD}(P, P^*) = \frac{1}{|P^*|} \sum_{v \in P^*} \text{dist}(v, P) \tag{2}$$

where dist(v, P) is the Euclidean distance between the solution v and its nearest point in P.

IGD is believed to be able to account for both convergence and diversity of the nondominated solutions, and a smaller IGD value indicates better performance of the MOEA. Since IGD requires a reference set, which should be evenly distributed on the Pareto optimal front of test problems. For all test problems, the closest integer to 10 000 is used as the number of reference points for IGD calculation.

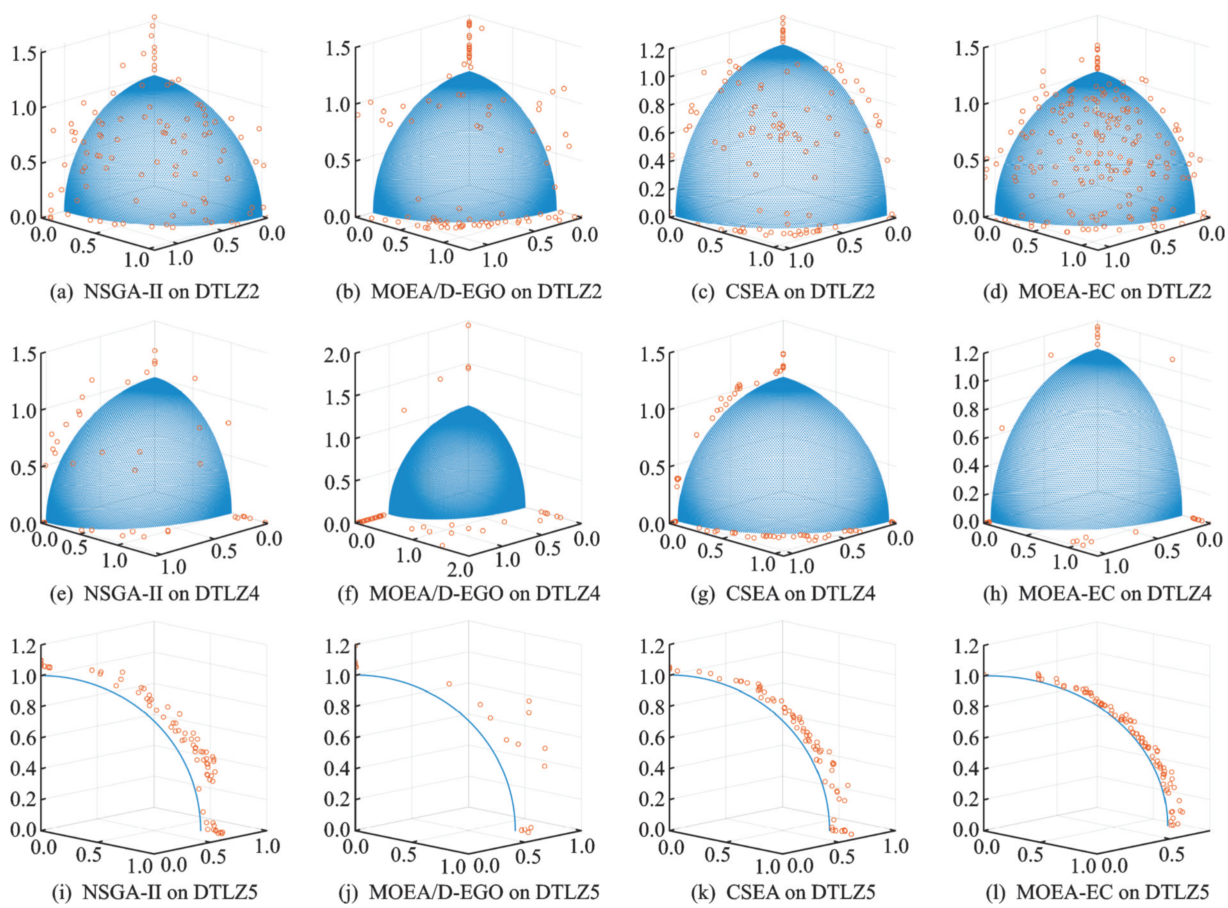
3.3 Experiment on DTLZ problems

The IGD values achieved by the four compared algorithms over ten independent runs on DTLZ problems are summarized in Table 1, where the best results are highlighted. It can be observed that MOEA-EC has achieved the best results on all the benchmark problems except for DTLZ7. In DTLZ7, MOEA-EC achieves the second-best performance, where MOEA/D-EGO has the best performance. Fig. 1 plots the final nondominated solution set obtained by the four compared algorithms in all ten runs on DTLZ2 and DTLZ4-7.

Table 1 Mean and std IGD values obtained of compared algorithms over 10 runs on DTLZ

Problem	M	NSGA-II	MOEA/D-EGO	CSEA	MOEA-EC
DTLZ1	3	9.166 0e+1 (1.66e+1) −	7.981 4e+1 (1.33e+1) −	7.468 9e+1 (7.86e+0) −	6.702 8e+1 (8.59e+0)
DTLZ2	3	2.760 6e−1 (3.30e−2) −	3.254 1e−1 (3.08e−2) −	2.427 9e−1 (1.61e−2) −	2.026 6e−1 (3.61e−2)
DTLZ3	3	2.254 4e+2 (4.67e+1) −	2.011 4e+2 (1.07e+1) −	1.877 2e+2 (9.59e+0) =	1.735 9e+2 (4.20e+1)
DTLZ4	3	7.610 5e−1 (1.92e−1) −	6.489 9e−1 (6.44e−2) −	6.188 0e−1 (4.77e−2) −	5.389 3e−1 (1.04e−1)
DTLZ5	3	1.760 1e−1 (1.90e−2) −	2.636 5e−1 (3.56e−2) −	1.111 1e−1 (2.22e−2) −	8.246 7e−2 (3.60e−2)
DTLZ6	3	5.904 3e+0 (3.04e−1) −	2.333 3e+0 (7.94e−1) −	5.460 2e+0 (5.50e−1) −	2.088 6e+0 (8.79e−1)
DTLZ7	3	3.975 2e+0 (9.54e−1) −	2.414 8e−1 (1.34e−1) +	2.044 6e+0 (2.73e−1) −	1.580 9e+0 (6.63e−1)
+/−/=		0/7/0	1/6/0	0/6/1	

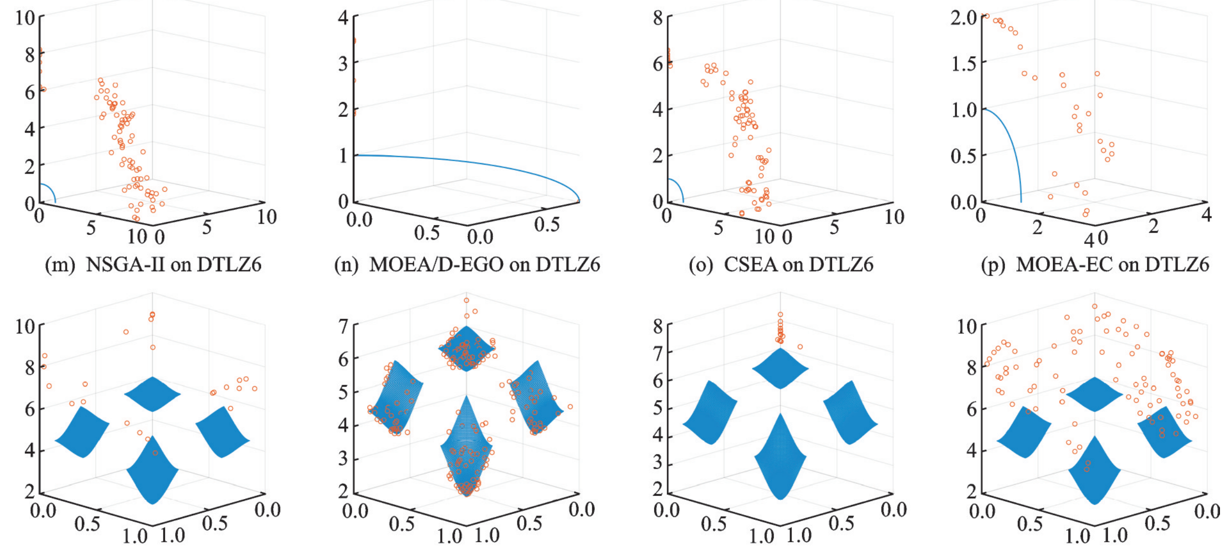
Wilcoxon's rank sum test at a 0.05 significance level is performed to IGD values on DTLZ problem. "+" means the IGD value of the algorithm on this problem is significantly better than that of MOEA-EC. "−" means the IGD value of the algorithm on this problem is significantly worse than that of MOEA-EC. "=" means there is no significant difference between the compared results.



(a) NSGA-II on DTLZ2 (b) MOEA/D-EGO on DTLZ2 (c) CSEA on DTLZ2 (d) MOEA-EC on DTLZ2

(e) NSGA-II on DTLZ4 (f) MOEA/D-EGO on DTLZ4 (g) CSEA on DTLZ4 (h) MOEA-EC on DTLZ4

(i) NSGA-II on DTLZ5 (j) MOEA/D-EGO on DTLZ5 (k) CSEA on DTLZ5 (l) MOEA-EC on DTLZ5



(m) NSGA-II on DTLZ6 (n) MOEA/D-EGO on DTLZ6 (o) CSEA on DTLZ6 (p) MOEA-EC on DTLZ6

(q) NSGA-II on DTLZ7 (r) MOEA/D-EGO on DTLZ7 (s) CSEA on DTLZ7 (t) MOEA-EC on DTLZ7

Fig.1 Final nondominated solution set obtained by four algorithms in all 10 runs on DTLZ2 and DTLZ4-7 with 10 variables

Both DTLZ1 and DTLZ3 have complex multi-modal landscapes, which makes them difficult for MOEAs to converge with a small number of FEs. After 300 FEs, the approximations obtained by the four compared algorithms on both DTLZ1 and DTLZ3 are far from the real PFs, thus they have not been included in Fig. 1. Nevertheless, it can be observed from Table 1 that MOEA-EC has the best performance in terms of IGD values, followed by CSEA, MOEA/D-EGO and NSGA-II.

For DTLZ2, as shown in Figs.1(a)—(d), MOEA-EC has the best performance in terms of both convergence and diversity, while the nondominated solution sets obtained by MOEA/D-EGO and CSEA are not diversely-distributed; and that obtained by NSGA-II is not well-converged. These observations are consistent with the results presented in Table 1.

For DTLZ4, as shown in Figs.1(e)—(h), MOEA-EC outperforms other compared algorithms, especially in convergence, although the solution sets delivered by all the compared algorithms do not have satisfactory diversity. It can be also observed in Table 1 that MOEA-EC achieves the best performance in terms of IGD values.

DTLZ5 and DTLZ6 contain the degenerated PFs. It can be observed from Figs.1(i)—(l) that MOEA-EC has the best convergence on DTLZ5 which might be attributed to the domination-based algorithm framework. For DTLZ6, the nondominated solution set obtained by MOEA-EC is not diversely-distributed but well-converged as shown in Figs.1(m)—(p). By contrast, the compared algorithms fail to achieve a set of well-converged solutions.

The real PF of DTLZ7 is discontinuous, and therefore diversity maintenance is challenging. It can be observed from Figs.1(q)—(t) and Table 1 that MOEA/D-EGO has the best performance, followed by MOEA-EC.

It can be seen from Table 1 that MOEA-EC is significantly better than NSGA-2 in all DTLZ test problems, and significantly better than MOEA/D-EGO in all test problems except for DTLZ7. Compared with CSEA, MOEA-EC is significantly better than CSEA in all test problems except for DTLZ3, but IGD value obtained by MOEA-EC on DTLZ3 is better than CSEA.

Fig.2 presents the evolutions of the median IGD values obtained by different algorithms on DTLZ test problems with 10 variables. It is clear that MOEA-EC converges faster than other algorithms in most test problems. In DTLZ1 and DTLZ6, MOEA-EC obtains the best final IGD values even if the convergence speed is slightly slower than MOEA/D-EGO. In DTLZ7, MOEA-EC has

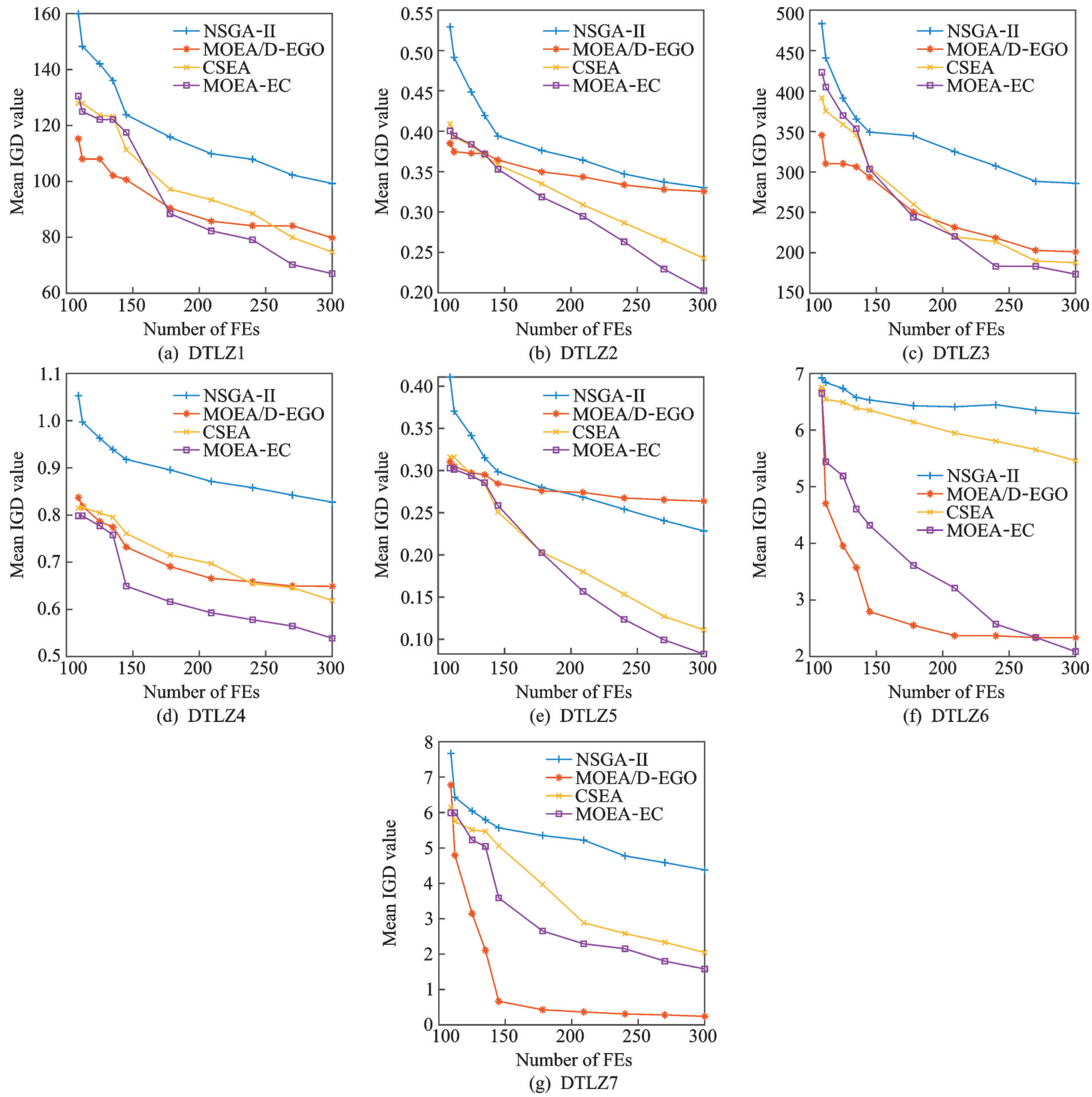


Fig.2 Evolutions of the mean IGD values obtained by different algorithms versus the number of function evaluations for DTLZ test instances with 10 variables

achieved second-ranked results in both convergence speed and IGD value.

In conclusion, MOEA-EC performs well in solving multi-modal landscapes, non-convex and degenerate problems, and has acceptable performance in solving discontinuous problems.

3.4 Experiment on ZDT problems

The IGD values achieved by the four compared algorithms over ten independent runs on ZDT problems are summarized in Table 2, where the best results are highlighted. It can be observed that MOEA-EC has achieved the best results on all the benchmark problems. Fig.3 plots the final nondominated solution set obtained by the four compared algorithms in all ten runs on ZDT1—4 and ZDT6.

ZDT1 has a convex Pareto optimal front and ZDT2 has a non-convex Pareto optimal front. It can be observed from Figs.3(a)—(b) that MOEA-EC has the best convergence. Because of the dominant framework used in MOEA-EC, the diversity performance of MOEA-EC is slightly worse than that of MOEA/D-EGO based on decomposition framework, but the diversity performance of MOEA-EC in the middle space is better than that of MOEA/D-EGO.

Table 2 Mean and std IGD values obtained by compared algorithms over 10 runs ON ZDT

Problem	M	NSGA-II	MOEAD-EGO	CSEA	MOEA-EC
ZDT1	2	9.859 6e+0 (2.51e+0) −	1.789 6e−1 (1.36e−1) −	1.810 1e+0 (8.49e−1) −	3.297 5e−2 (1.24e−2)
ZDT2	2	1.199 4e+1 (2.57e+0) −	2.428 4e−1 (2.27e−1) −	3.667 8e+0 (2.07e+0) −	4.137 5e−2 (1.72e−2)
ZDT3	2	1.175 7e+1 (2.56e+0) −	2.883 8e−1 (7.89e−2) −	2.124 5e+0 (5.17e−1) −	5.037 3e−2 (2.33e−2)
ZDT4	2	5.694 4e+1 (9.53e+0) −	8.740 6e+1 (7.43e+0) −	3.880 9e+1 (7.73e+0) =	3.435 7e+1 (9.49e+0)
ZDT6	2	1.012 6e+1 (4.22e−1) −	2.734 8e+0 (1.58e+0) =	6.697 3e+0 (1.22e+0) −	1.610 1e+0 (2.15e−1)
+/−/=		0/5/0	0/4/1	0/4/1	

Wilcoxon's rank sum test at a 0.05 significance level is performed to IGD values on ZDT problem. "+" means the IGD value of the algorithm on this problem is significantly better than that of MOEA-EC. "−" means the IGD value of the algorithm on this problem is significantly worse than that of MOEA-EC. "=" means there is no significant difference between the compared results.

ZDT3 has several non-contiguous convex parts in Pareto optimal front, as shown in Fig.3(c), MOEA-EC outperforms other compared algorithms in terms of both convergence and diversity. Although MOEA/D-EGO can obtain the diversity at the boundary well, MOEA-EC achieves better diversity at the global space.

ZDT4 has multi-modal landscapes. ZDT6 contains two difficulties: Non-uniformity of search space and the low density of solutions near the Pareto optimal front. It can be observed from Figs.3(d)—(e) that all the comparison algorithms fail to converge to the reference PF, but MOEA-EC obtains the best convergence performance on both test problems and the best diversity performance on ZDT6.

It can be seen from Table 2 that MOEA-EC is significantly better than NSGA-2 in all ZDT test problems, significantly better than MOEA/D-EGO in all test problems except for ZDT6 and significantly better than CSEA in all test problems except for ZDT4. On ZDT4 and ZDT6, MOEA-EC is not significantly different from CSEA or MOEA/D-EGO, but the IGD value obtained by MOEA-EC are better than other algorithms.

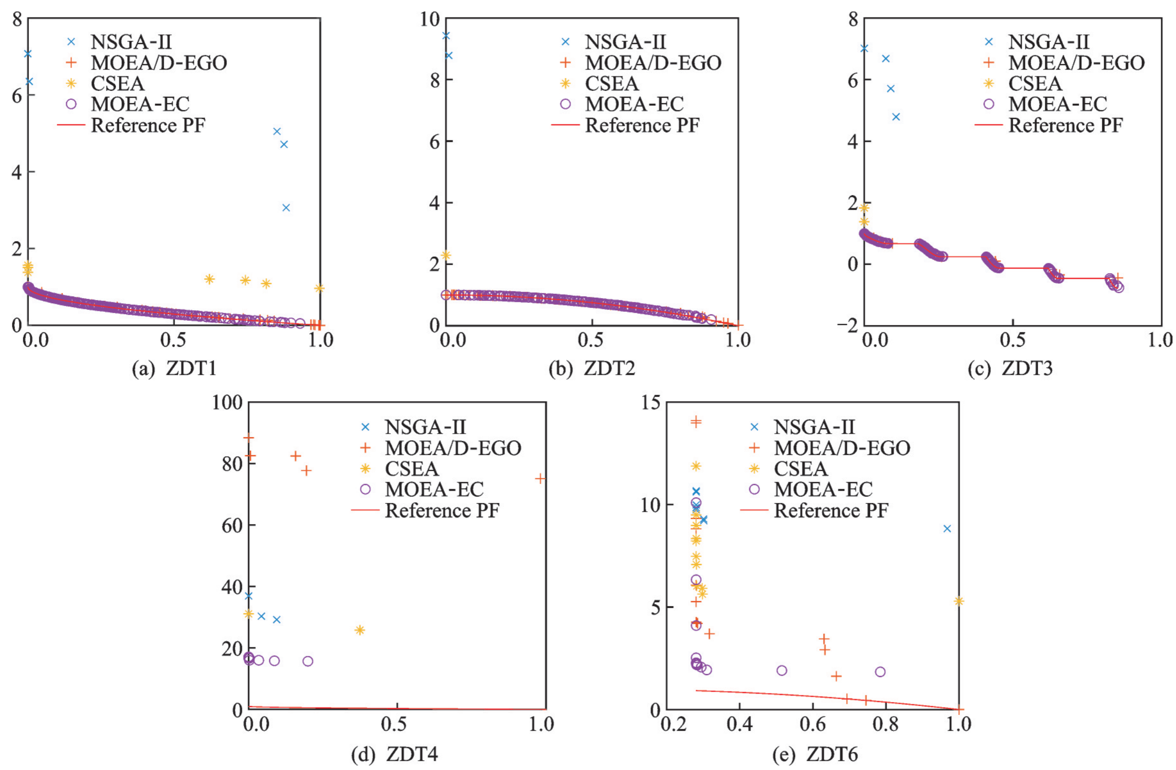
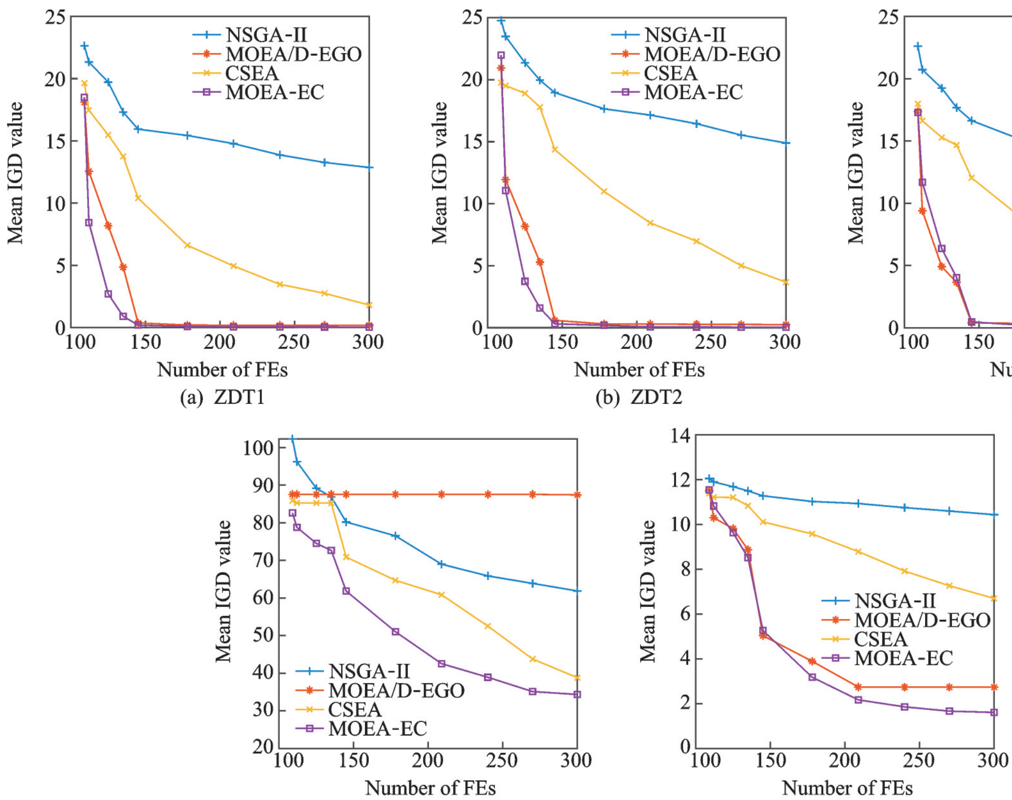


Fig.3 Final nondominated solution set obtained by four algorithms in all 10 runs on ZDT1—4 and ZDT6 with 10 variables

Fig.4 presents the evolutions of the median IGD values obtained by different algorithms on ZDT test problems with 10 variables. It is clear that MOEA-EC converges faster than other algorithms and achieve the best IGD value in all test problems.

The experimental results confirm once again that MOEA-EC performs well in solving multi-modal landscapes, non-convex and degenerate problems, and has good performance in solving two-objective discontinuous problems.



(a) ZDT1 (b) ZDT2 (c) ZDT3 (d) ZDT4 (e) ZDT6

Fig.4 Evolutions of the mean IGD values obtained by different algorithms versus the number of function evaluations for ZDT test instances with 10 variables

4 Conclusions

In this paper, we propose a multiobjective evolutionary algorithm with an ensemble classifier, called MOEA-EC, for the expensive multiobjective optimization problems. The ensemble classifier consists of multiple CART models, which is used to identify the promising solutions from candidate offspring for real FEs during search procedure. The effects of the ensemble classifier are empirically investigated and the experimental results confirmed that the proposed MOEA-EC significantly outperforms other compared algorithms.

The work reported in this paper is very preliminary. More advanced machine learning model can be further incorporated into MOEAs for enhancing their performance on EMOPs.

References

[1] KALYANMOY D. Multi objective optimization using evolutionary algorithms[M]. [S.l.]: John Wiley and Sons, 2001.

[2] MIETTINEN K. Nonlinear multiobjective optimization [J]. International Series in Operations Research and Management Science, 1999, 51(2): 246.

[3] KALYANMOY D, PRATAP A, AGARWAL S, et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(2): 182-197.

[4] ZITZLER E, LAUMANNS M, THIELE L. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization[C]//Proceedings of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problem. Ath-

ens, Grecle: [s.n.], 2001.

[5] ZHANG Q, HUI L. Moea/d: A multiobjective evolutionary algorithm based on decomposition[J]. IEEE Transactions on Evolutionary Computation, 2007, 11(6): 712-731.

[6] CAI X, MEI Z, FAN Z, et al. A constrained decomposition approach with grids for evolutionary multiobjective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 22(4): 564-577.

[7] ZITZLER Z, KÜNZLI S. Indicator-based selection in multiobjective search[C]//Proceedings of International Conference on Parallel Problem Solving from Nature. [S.l.]: Springer, 2004: 832-842.

[8] SUN Y, YEN G G, ZHANG Y. IGD indicator-based evolutionary algorithm for many-objective optimization problems[J]. IEEE Transactions on Evolutionary Computation, 2018, 99: 1.

[9] KUYA Y, TAKEDA K, ZHANG X, et al. Multifidelity surrogate modeling of experimental and computational aerodynamic data sets[J]. AIAA Journal, 2011, 49(2): 289-298.

[10] DOUGUET D. E-lea3D: A computational-aided drug design web server[J]. Nucleic Acids Research, 2010, 38(S2): 615-621.

[11] WANG H, JIN Y, JANSEN J O. Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system[J]. IEEE Trans Evolutionary Computation, 2016, 20(6): 939-952.

[12] BROOMHEAD D S, LOWE D. Radial basis functions, multi-variable functional interpolation and adaptive networks [R]. United Kingdom: Royal Signals and Radar Establishment Malvern, 1988.

[13] ZURADA J M. Introduction to artificial neural systems [M]. Avonmore: West Publishing Company, 1992.

[14] KRIGE D G. A statistical approach to some basic mine valuation problems on the witwatersrand[J]. Journal of the Southern African Institute of Mining and Metallurgy, 1951, 52(6): 119-139.

[15] ALLMENDINGER R, EMMERICH M T, HAKANEN J, et al. Surrogate-assisted multicriteria optimization: Complexities, prospective solutions, and business case [J]. Journal of Multi-criteria Decision Analysis, 2017, 24(1/2): 5-24.

[16] HABIB A, SINGH H, CHUGH T, et al. A multiple surrogate-assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(6): 1000-1014.

[17] WANG H, JIN Y. A random forest-assisted evolutionary algorithm for data-driven constrained multiobjective combinatorial optimization of trauma systems [J]. IEEE Transactions on Cybernetics, 2018, 50(2): 536-549.

[18] CHUGH T, JIN Y, MIETTINEN K, et al. A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018,22(1): 129-142.

[19] PAN L, HE C, TIAN Y, et al. A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization[J]. IEEE Transactions on Evolutionary Computation, 2018, 23(1): 74-88.

[20] BHATTACHARJEE K S, SINGH H K, RAY T. Multiple surrogate-assisted many-objective optimization for computationally expensive engineering design[J]. Journal of Mechanical Design, 2018, 140(5): 051403.

[21] HE C, CHENG R, ZHANG C, et al. Evolutionary large-scale multiobjective optimization for ratio error estimation of voltage transformers[J]. IEEE Transactions on Evolutionary Computation, 2020, 24(5): 868-881.

[22] TIAN Y, YANG S, ZHANG L, et al. A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining[J]. IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, 3(2): 106-116.

[23] MIN A T W, ONG Y S, GUPTA A, et al. Multi-problem surrogates: Transfer evolutionary multiobjective optimization of computationally expensive problems[J]. IEEE Transactions on Evolutionary Computation, 2019, 23(1): 15-28.

[24] LYU Z, WANG L, HAN Z, et al. Surrogate-assisted particle swarm optimization algorithm with pareto active learning for expensive multi-objective optimization[J]. IEEE/CAA Journal of Automatica Sinica, 2019, 6(3): 838-849.

[25] EVERITT B S. Classification and regression trees [M]. Hoboken: John Wiley & Sons, Ltd, 2005.

[26] FARNSWORTH D, ORR R. Gini means[J]. The American Mathematical Monthly, 1986, 93(8): 603-607.

[27] DIETTERICH T G. Ensemble methods in machine learning[C]//Proceedings of International Workshop on Multiple Classifier Systems. [S. l.]: Springer, 2000: 1-15.

[28] DEB K, THIELE L, LAUMANNS M, et al. Scalable multi-objective optimization test problems [C]//Proceedings of the 2002 Congress on Evolutionary Computation. Honolulu: IEEE, 2002: 825-830.

[29] YE T, RAN C, ZHANG X, et al. Platemo: A matlab platform for evolutionary multi-objective optimization[J]. IEEE Computational Intelligence Magazine, 2017, 12(4): 73-87.

[30] ZHANG Q, LIU W, TSANG E, et al. Expensive multiobjective optimization by moea/d with gaussian process model[J]. IEEE Transactions on Evolutionary Computation, 2010, 14(3): 456-474.

[31] PEDREGOSA F, VAROQUAUX G, GRAMFORT A, et al. Scikit-learn: Machine learning in python[J]. Journal of Machine Learning Research, 2013, 12(10): 2825-2830.

[32] ZHOU A, ZHANG Q, JIN Y, et al. A model-based evolutionary algorithm for bi-objective optimization [C]//Evolutionary Computation, The 2005 IEEE Congress on. [S.l.]: IEEE, 2005: 2568-2575.

Author Mr. LAN Tian received the B.E. degree from College of Software Engineering, Jinling Institute of Science and Technology, Nanjing, China, in 2016, and the M.S. degree from College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2020. His research interests include evolutionary computation, multiobjective optimization, and surrogate-assisted evolutionary optimization.

Author contributions Mr. LAN Tian designed the study, investigated the background, complied the models, conducted the analysis, interpreted the results and wrote the manuscript. The author commented on the manuscript and approved the submission.

Competing interests The author declare no competing interests.

(Production Editor: LIU Yandong)

基于集成分类器代理模型的昂贵多目标进化算法

蓝 天

(南京航空航天大学计算机科学与技术学院,南京 211106,中国)

摘要: 对于许多现实世界中的多目标优化问题,它们目标函数的评估在计算上通常成本高昂,这些问题称为昂贵多目标优化问题(Expensive multiobjective optimization problems, EMOP)。现有的解决昂贵多目标优化问题的一种可行方式是引入计算效率较高的代理模型来减少函数评估的次数。受集成学习启发,本文针对EMOP,提出了基于集成分类器代理模型的多目标进化算法(Multiobjective evolutionary algorithm with an ensemble classifier, MOEA-EC)。具体来说,多个决策树(Classification and regression tree, CART)模型组成一个集成分类器用于进行于选择操作。相比于使用单个不准确的模型参与计算,这种方法可以更有效地减少解的函数评估次数。为了验证MOEA-EC算法的有效性,本文进行了实验研究,并与几种先进的昂贵多目标优化算法进行了比较。实验结果表明,MOEA-EC算法的性能优于其他比较算法。

关键词: 多目标进化算法;昂贵多目标优化;集成分类器;代理模型