IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 June 2022, accepted 30 June 2022, date of publication 4 July 2022, date of current version 19 July 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3188248

RESEARCH ARTICLE

Expensive Multiobjective Optimization Algorithm Based on Equivariate Component Analysis

LI WENBIN, CHAI ZI'AN, AND LIU GANG
School of Information Science and Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China

Corresponding author: Liu Gang (4350594@qq.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 52076072, in part by the Youth Project of Scientific Research of Hunan Provincial Education Department under Grant 19B231, and in part by the Natural Science Foundation of Hunan Province under Grant 2020JJ4339.

ABSTRACT In order to reduce the cost of candidate solution evaluation in the process of solving expensive optimization problems, an expensive multi-objective optimization algorithm based on equivalence component analysis was proposed to study the influence of decision space equivalence components on the prediction accuracy of agent models. Based on the analysis of the equivalence of decision space attributes, a limit learning network based on the equivalence components was constructed for Pareto dominance prediction among candidate solutions. A multi-objective test problem with equivalent components was selected in Pareto dominance prediction experiments, the results of which showed that the algorithm can effectively improve the accuracy of Pareto dominance prediction among candidate solutions. Successively the candidate solutions were scored with multiple ELM (Extreme Learning Machine) models, selected for evaluation and updated, and integrated into the Pareto-based multi-objective evolutionary algorithm. Through comparative experiments on the test problem, the method could achieve a better Pareto approximation solution under the limitation of a limited number of evaluations, and the goal of reducing the cost of expensive multi-objective optimization calculations.

INDEX TERMS Expensive multi-objective optimization, equivalent components, Pareto dominance, surrogate model.

I. INTRODUCTION

Expensive multi-objective optimization problems (EMOPs) [1] are prevalent in many fields such as industrial scheduling [2], complex structure design [3], [4], and robotics experiments [5], and are a hot research topic in engineering applications. When evolutionary algorithms are evaluation of a large number of candidate solutions is required, which leads to a serious computational cost disaster. Therefore, how to reduce the number of costly evaluations of candidate solutions in multi-objective evolutionary algorithms [6] has attracted a lot of attention from scholars.

Surrogate-assisted evolutionary algorithms (SAEAs) [7], [8] is one of the main approaches to solve EMOPs, which can effectively reduce the number of expensive evaluations of candidate solutions. Ji *et al.* [9] proposed a two-population cooperative particle swarm optimizer to explore/exploit multiple modalities simultaneously. Subsequently, a modality-guided two-level cooperative surrogate model was constructed, capable of reducing individual evaluation costs. Liu *et al.* [10] used the R2 metric of the utility function to select candidate solutions for the evaluation, taking into account the diversity of populations, convergence, prediction expectation and prediction mean square error, which enhances the ability of populations to detect the space and can obtain solution sets with excellent convergence and distribution. Gu *et al.* [11] used a Gaussian model to approximate the objective function, and then projected the individuals in the objective space to the radial space and selected the appropriate solution for realistic evaluation based on the crowding level of individuals in the radial space, which was able to obtain a better solution. Li *et al.* proposed the surrogates assisted particle swarm optimization (EAPSO) [12] algorithm, which uses multiple trial positions for each particle in the population. And the optimal solutions of the polynomial regression model and the radial basis function model are evaluated in the convergence state of the particles. Meanwhile the superiority of integration and uncertainty are

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Li.

used to select promising positions to improve the convergence speed of the particles. Lucas and Pavelski *et al.* combined decomposition-based MOEA (Multi-objective Evolutionary Algorithm) and ELM regression models to propose the ELMOEA/D (Extreme Learning Surrogate Models in Multi-objective Optimization based on Decomposition) algorithm [13], [14], which contains only one ELM per agent model instead of a set of radial basis function networks and is able to reduce the computational resources consumed by the model. However, this algorithm can only select one solution for each evaluation, and the complexity of the algorithm is high. Cai *et al.* proposed an adaptive multitask population collaborative search algorithm (AMMCS) [15], which uses similarity metrics to detect evaluated individuals in real time to obtain the relationship between subproblems so that it adaptively divides tasks, which can improve prediction accuracy and sampling efficiency. However, the algorithm uses a fixed direction vector, which is generally effective for multi-objective problems with irregular Pareto fronts. For expensive multi-objective optimization problems, scholars have carried out fruitful researches on SAEAs algorithms, but since the design of SAEAs is an integrated process, there is still a lack of proven general models for agent objective and model selection, candidate solution selection evaluation and model management in the iterative evolution process.

In terms of pattern-based classification studies, Guo *et al.* [16] used pattern recognition methods to directly predict the Pareto dominance relationship among candidate solutions, clarify the agent objectives and reduce the number of agent model constructions in expensive multi-objectives, and proposed Pareto dominance nearest neighbor prediction based on the weighted sum of binary bit strings in decision space and Pareto dominance nearest neighbor prediction by generalized regression neural networks. Subsequently, to improve the accuracy of Pareto dominance prediction nearest neighbor classification methods, Guo Guanqi *et al.* proposed Pareto dominance prediction based on cross-similarity of equivalent components [17] and Pareto dominance prediction based on decision space transformation nearest neighbor method [18]. However, the Pareto dominant nearest neighbor prediction method suffers from the difficulty of convergence of the optimization algorithm when dealing with EMOPs with large comprehension spaces, therefore Li *et al.* proposed the Pareto dominant prediction method that combines the sequential relational predictive agent model with the decision space transformation method [19]. Since the nearest neighbor method is difficult to truly measure and distinguish the distance between candidate solution samples when facing problems such as high-dimensional decision space or nonlinear characteristics among decision space attributes, it is easy to have the problem of large individual differences between nearest neighbors in decision space and target space, which affects the prediction accuracy. Therefore, considering that the target space and decision space are potentially linked when performing Pareto dominance prediction, this paper introduces the decision space equivalence component similarity into Pareto dominance prediction, investigates the relationship between the existence of equivalence components in the decision space and the accuracy of the agent model, designs an extreme learning machine model based on equivalence components to predict Pareto dominance among candidate solution samples, and interacts with an expensive multi-objective optimization algorithm. The main innovations are as follows:

1) Propose an equivalence component analysis method and apply it to improve model prediction accuracy in agent model construction.
2) Propose a multi-model co-decision method for evaluating the selection of new candidate solutions to explore the unknown decision space.
3) A comparison was made of the performance of the proposed algorithm against state of the art frameworks on several benchmark test MOPs.

The remainder of the paper was as follows: Section II introduces some of the basic concepts used in the paper. Section III performs the Pareto dominance prediction. Section IV presents an expensive multi-objective optimization algorithm based on equivalent component analysis and gives comparative experimental results. Section V the conclusions of the full paper are drawn.

II. RELATED BACKGROUND

A. EQUIVALENT WEIGHT

For a multivariate function $f(x) = f(x_1, \cdots, x_i, \cdots, x_j, \cdots, x_n)$, if $x_i, x_j\ (i \sim= j)$ in the decision variables have the same scope and the exchange yields $f\left(X'\right) = f(x_1, \cdots, x_j, \cdots, x_i, \cdots, x_n)$, if $f(X) = f\left(X'\right)$, then x_i, x_j is said to be an equivalent component of the function $f(X)$.

In the multivariate function $f(X)$, the following properties exist for the equivalent components:

1) Self-referential: if x_i is a set of equivalent components of $f(X)$, $f(X) = f\left(x_1, \cdots, x_i, \cdots, x_j, \cdots, x_n\right) = f(x_1, \ldots, x_i, \cdots, x_j, \cdots, x_1)$.
2) Symmetry: if x_i, x_j is a set of equivalent components of $f(X)$, $f(x) = f\left(x_1, \cdots, x_i, \cdots, x_j, \cdots, x_n\right) = f(x_n, \ldots, x_j, \ldots, x_i, \ldots, x_1)$.
3) Transferability: if x_i, x_j is a set of equivalent components of $f(X)$, x_j, x_k is a set of equivalent components of $f\left(X'\right)$, according to symmetry, there is another set of equivalent components of $f(x_1, \cdots, x_i, \cdots, x_j, \cdots, x_k, \cdots, x_n) = f(x_1, \cdots, x_j, \cdots, x_i, \cdots, x_k, \cdots, x_n)$, $f(x_1, \cdots, x_i, \cdots, x_j, \cdots, x_k, \cdots, x_n) = f(x_1, \cdots, x_i, \cdots, x_k, \cdots, x_j, \cdots, x_n)$ and $f(x_1, \cdots, x_i, \cdots, x_k, \cdots, x_j, \cdots, x_n) = f(x_1, \cdots, x_j, \cdots, x_i, \cdots, x_k, \cdots, x_n)$, due to x_j, x_k is a set of equivalent components, so $f(x_1, \cdots, x_i, \cdots x_k, \cdots, x_j, \cdots, x_n) = f(x_1, \cdots, x_j, \cdots, x_i, \cdots, x_k, \cdots, x_n) = f(x_1, \cdots, x_k, \cdots, x_i, \cdots, x_j, \cdots, x_n)$. Therefore,according to the definition of equivalence, x_i, x_k equivalent, i.e. x_i, x_j, x_k is equivalent to each other.

B. EQUIVALENT COMPONENT ANALYSIS

In expensive optimization problems, it is difficult to directly exchange decision component evaluation to identify the existence of equivalent components due to the high cost of objective function evaluation, so a method is designed for this type of problems to identify equivalent components in expensive optimization problems at the least possible cost. The method is as follows:

Step 1: Analyze the scopes of decision variables, find decision components with the same scope, and put them into the set S.

Step 2: Equation (1) is used to calculate the correlation coefficient between each decision component (x_i) in S and the objective function $y = f(X)$,and the decision components with the same or similar correlation coefficients to the objective function $f(X)$ are put into the set S1.

Step 3: Swap the decision components $x_i, x_j (i \sim= j)$ in the set S1, compare the objective function values before and after the swap, and identify the equivalent components.

$$r_{x_i y} = \frac{C_{x_i y}}{S_{x_i} S_y} = \frac{\sum (x_i - \overline{x})(y - \overline{y})}{N S_{x_i} S_y} = \frac{N \sum x_i y - \sum x_i \sum y}{\sqrt{\sum x_i^2 - \frac{(\sum x_i)^2}{N}} \sqrt{\sum y_i^2 - \frac{(\sum y_i)^2}{N}}} \tag{1}$$

where $C_{x_i y}$ is the covariance of x_i and y, and $S_{x_i} S_y$ is the product of the standard deviation of the two. If x_i, yare not correlated, then $r_{x_i y} = 0$,It is usually assumed that there is no linear relationship between x and y, but it is not excluded that other relationships may exist between x and y.

In a classification or regression problem, if a sample set M of size N is given, assume that the corresponding function on the sample set is $f(X)$, and there exists $x_i, x_j (i \sim= j)$ that is an equivalent component on its decision space. Exchanging x_i, x_j, since x_i, x_j is equivalent, the corresponding $f(X)$ remains unchanged, which is equivalent to the increased known sample set M (as in Figure 1). The blue circle in Figure 1 indicates the original sample M, and the red triangle indicates the sample set M' resulting from the exchange of equivalent components, that is, multiplying the number of known samples. Similarly, it can be seen that if more equivalence components exist, the corresponding number of known samples multiplies more, such as 3 components equivalence in the decision space, the exchange is equivalent to expanding the always sample to 5 times of the original. The formula for calculating the number of sample increase is as in equation (2), where Q is the sample expansion multiplier and A_k^k denotes the full arrangement of k equivalent components.

$$Q = A_k^k - 1 \tag{2}$$

If there are equivalent components in the decision space of the expensive optimization problem, the number of known samples can be increased by exchanging equivalent components, so if the application of equivalent components is considered in the process of constructing the machine learning agent model, the spatial features can be more fully learned,

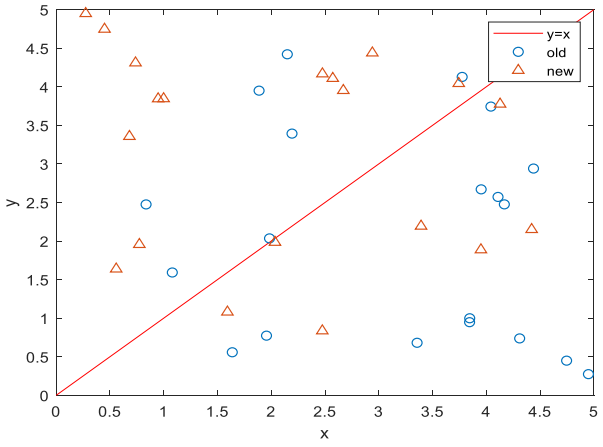

FIGURE 1. Schematic diagram of the equivalence component sample multiplication relationship.

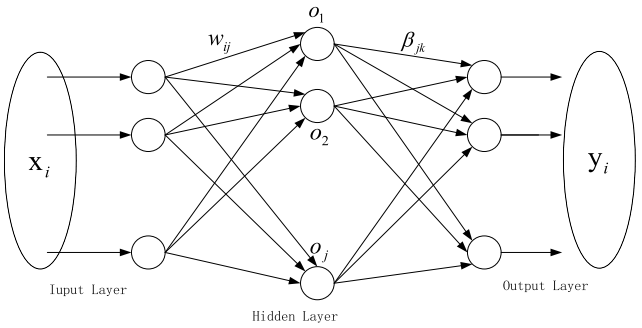

FIGURE 2. ELM network structure diagram.

which is beneficial to improve the model prediction accuracy. Therefore, this paper study how to improve the accuracy of the prediction of the Pareto dominant agent model of the multi-objective optimization problem using the property of equivalence components in the extreme learning machine network model.

C. ELM MODEL BASED ON EQUIVALENCE COMPONENTS

For the limit learning machine $f(\boldsymbol{x}; \boldsymbol{w})$, where the input is an l-dimensional vector $\boldsymbol{x}(\boldsymbol{x} = (x_1 \ldots x_l))$ and the limit learning machine weight parameter is $\boldsymbol{w}$. We find that for an equivalent dimensional subset of the learning task $x_{i_1}, \ldots, x_{i_k}\}_{1 \le i_1 < \ldots < i_k \le l}$, if the weight parameters $\boldsymbol{w_{i_j}}$ associated with x_{i_j} are equal in the range $1 \le j \le k$, then the limit learning machine $f(\boldsymbol{x}; \boldsymbol{w})$ must have $f(\boldsymbol{x}; \boldsymbol{w}) = f(\boldsymbol{x'}; \boldsymbol{w})$ hold for any new l-dimensional input vector $\boldsymbol{x'}$ obtained by disrupting an equivalent dimensional subset. Therefore, we introduce the equivalence dimension information in the training process of the limit learning machine by setting the weight parameters corresponding to any two equivalence dimensions to be equal in order to improve the learning generalization.

There are many expensive multi-objective optimization algorithms based on agent model, and neural network is a common agent model. ELM [14], [20], [21] has the characteristics of fast training speed and good generalization ability.

TABLE 1. Correlation coefficient between decision attributes of the test problem and the objective function.

Test Problem		$f_1(x)$	$f_2(x)$	$f_3(x)$
ZDT1	n=10	(1,0,0, ...)	(−0.67,0.18,0.18, ...)	
ZDT2	n=10	(1,0,0, ...)	(−0.81,0.09,0.09, ...)	
ZDT3	n=10	(1,0,0...)	(-0.34,0.58,0.58, ...)	
ZDT6	n=10	(0.54,0,0, ...)	(0.08,0.61,0.61, ...)	
UF1	n=10	(0.35,0,0, ...)	(0.15,0.74,0.74, ...)	
UF2	n=10	(0.51, 0,0, ...)	(0.09, 0.64,0.64, ...)	
DTLZ1	n=10	(-0.59,-0.59,0.01,-0.01,...)	(-0.59,-0.59,0.01,-0.01,⋯)	(0.87,-0.03,0.03,0,⋯)
DTLZ2	n=10	(-0.59,-0.59,0.01,-0.01,...)	(-0.59,0.59,0.01,0.01,⋯)	(0.87,,0.03,0,⋯)

TABLE 2. Equivalent components of the test problem at a given point.

Test Problem		$f_1(x)$	$f_2(x)$	$f_3(x)$
ZDT1	n=10	(x_1), $(x_2, x_3, ...)$	(x_1), $(x_2, x_3, ...)$	
ZDT2	n=10	(x_1), $(x_2, x_3, ...)$	(x_1), $(x_2, x_3, ...)$	
ZDT3	n=10	(x_1), $(x_2, x_3, ...)$	(x_1), $(x_2, x_3, ...)$	
ZDT6	n=10	(x_1), $(x_2, x_3, ...)$	(x_1), $(x_2, x_3, ...)$	
UF1	n=10	(x_1), $(x_2, x_3, ...)$	(x_1), $(x_2, x_3, ...)$	
UF2	n=10	(x_1), $(x_2, x_3, ...)$	(x_1), $(x_2, x_3, ...)$	
DTLZ1	n=10	(x_1, x_2), $(x_3, ...)$	(x_1, x_2), $(x_3, ...)$	(x_1), (x_2, x_3), $(x_4, ...)$
DTLZ2	n=10	(x_1, x_2), $(x_3, ...)$	(x_1, x_2), $(x_3, ...)$	(x_1), (x_2), $(x_3, ...)$

Therefore, elm is used as an agent model to reduce the time of real function evaluation in this paper. Extreme learning machine has three layers: input layer, hidden layer and output layer. Compared with other neural networks, the weights of hidden layer nodes are given randomly and do not need to be updated. Only the output weights are calculated. Its network structure is shown in Figure 2.

There are any N samples (X_i, t_i), where $X_i = [x_{i1}, x_{i2}, \cdots, x_{in}] \in R^n, t_i = [t_{i1}, t_{i2}, \cdots, t_{im}]^T \in R^n$. For a single-hidden layer neural network with L hidden layer nodes [22] can be expressed as:

$$\sum_{i=1}^{L} \beta_i g(W_i \cdot X_j + b_j) = o_j, j = 1, 2 \ldots, N \qquad (3)$$

where g(x) is the activation function, $W_i = [w_{i1}, w_{i2}, \cdots, w_{in}]^T$ is the input weight, β_i is the output weight, b_i is the bias of the i-th hidden layer unit.

For the components present in the decision space, their relationships are linearly correlated, equivalently and nonlinearly correlated. In the ELM network, they should be consistent with the other components, so the equivalent components are set in the implied layer weights as: $w_{i1} = w_{i2} = \cdots = w_{in}$, the randomness of the ELM is changed to improve the accuracy of the extreme learning network.

III. PARETO DOMINANT PREDICTION EXPERIMENT

This section describes the effect of equivalent component analysis Pareto dominance prediction, and the selected multi-objective optimization test problems are ZDT1, ZDT2, ZDT3, ZDT6, UF1, UF2, DTLZ1 and DTLZ2, and n is the number of decision space dimensions. The correlation coefficients of the decision components and the objective function are calculated according to equation (1), and the decision components with approximate correlation coefficients are selected to exchange the objective function, and the equivalent components are identified by the exchanged objective function values. The correlation coefficients of the decision attributes and the objective function for each test problem are shown in Table 1. By exchanging the objective functions of the components with similar correlation coefficients in Table 1, the equivalent components of each test problem can be derived, as shown in Table 2.

Three different algorithms are adopted: Nearest neighbor classification based on Euclidian distance measurement, ENNC [23], Nearest neighbor classification based on equivalent similarity, ESNNC and Gaussian process, GP [7] to compare the prediction accuracy of the algorithms. Each algorithm is run 20 times independently.

TABLE 3. Average prediction accuracy of ECA, ENNC, ESNNC and GP.

Test Problem	n	Algorithm	Pareto Dominance(%)			
			≺	≻	~	Accuracy
ZDT1	10	ECA	89.66	98.62	90.77	94.76
		ENNC	64.82	74.88	40.59	64.79
		ESNNC	65.51	87.21	70.12	83.23
		GP	65.45	90.11	65.45	88.76
ZDT2	10	ECA	74.62	63.07	68.46	79.54
		ENNC	52.14	46.91	38.26	40.76
		ESNNC	60.22	52.64	72.96	71.92
		GP	49.82	56.29	53.78	65.39
ZDT3	10	ECA	70.87	75.12	68.40	71.36
		ENNC	38.26	34.62	36.27	58.86
		ESNNC	67.21	60.57	72.10	76.85
		GP	53.28	78.23	54.29	71.10
ZDT6	10	ECA	46.21	39.73	50.46	45.68
		ENNC	46.87	46.71	37. 11	44.12
		ESNNC	61.12	65.27	65.02	64.85
		GP	46.92	51.13	47.22	48.68
UF1	10	ECA	72.16	78.35	59.36	63.34
		ENNC	50.23	65.76	54.86	60.08
		ESNNC	70.69	71.36	63.12	60.13
		GP	60.23	51.29	61.37	57.72
UF2	10	ECA	76.47	69.59	72.56	67.92
		ENNC	52.87	49.28	64.92	55.25
		ESNNC	58.61	64.38	70.04	56.93
		GP	37.09	64.72	52.55	47.42
DTLZ1	10	ECA	48.89	60.61	39.77	67.16
		ENNC	25.85	61.54	30.97	59.46
		ESNNC	48.68	90.11	35.81	70.12
		GP	12.31	90.14	13.91	89.67
DTLZ2	10	ECA	41.46	78.55	37.56	56.12
		ENNC	17.46	58.74	17.88	58.77
		ESNNC	28.74	95.14	22.47	95.10
		GP	22.67	89.45	23.63	82.66

From table 3 that the prediction accuracy of ECA algorithm is higher than that of the other three algorithms on problems ZDT 1, ZDT2 and ZDT3. On zdt6, UF1 and uF2, the prediction accuracy of ECA algorithm is similar to that of comparison algorithm. On the test questions DTLZ1 and DTLZ2, the prediction accuracy of ECA algorithm is higher than ENNC and GP algorithm, which is similar to ESNNC algorithm.

IV. EMOEAS ALGORITHM BASED ON EQUIVALENCE COMPONENT

A. EXPENSIVE MULTI-OBJECTIVE OPTIMIZATION ALGORITHM DESIGN

The agent model based on equivalence components can effectively improve the Pareto dominance prediction accuracy among samples, so an expensive multi-objective optimization algorithm (ECA-EMO) based on equivalence component analysis is designed. Since the known sample space is increasing in the evolutionary iteration process, how to select individuals for evaluation is the key to model management and another key point for the success of the algorithm, so this paper studies the equivalence between decision attributes to obtain the mapping relationship from decision space to target space to ensure the distributivity of the population in the target space, then the classification is carried out by the distance from the population to the weight, and selects the Pareto non-dominated individuals for evaluation. In order to improve the model quality, the evaluated liberation is updated into the model after each iteration.

Algorithm 1 ECA-EMO Algorithm.

Step1: Initialize the weight vector W, the number of nodes N, maximum evolutionary algebra Emax;
Step2: Initial population Pop of size NN was generated using Latin hypercube sampling and evaluated using the true function;
Step3: Using ELM model to evaluate populations to generate progeny solutions and merge parent-child populations;
Step4: The merged parent and child populations were selected to get the best individuals into Pop;
Step5: Selection of non-dominated individuals for assessment using multi-model shared decision making and distributed distance assessment;
Step6: Update Pop and Evaluating populations using test functions;
Step7: If Size $<$ Q, the loop continues; otherwise, the algorithm end;
Step8: Update model;
Step9: If the evolutionary algebra gen $>$ Emax, the algorithm ends; otherwise, turn to Step 5;
Step10: Output the Population Pop.

In Algorithm 1, the initial population of size M is first generated in the initialization phase by first generating the weight vector W, the number of nodes N, and the evolutionary

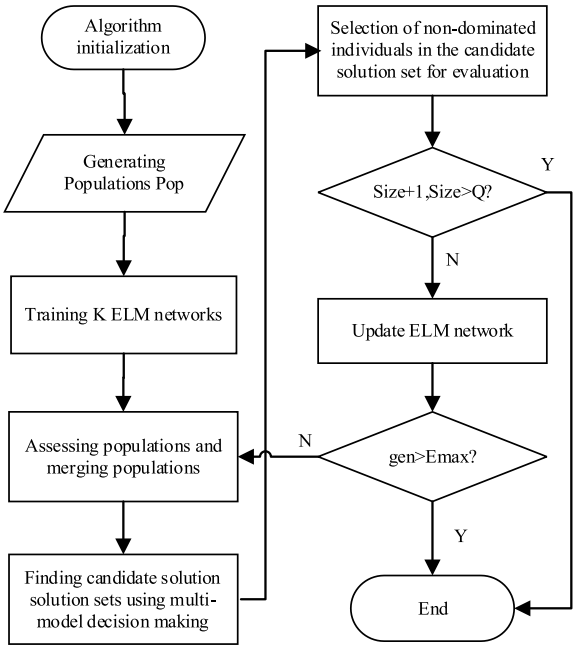

FIGURE 3. Flow chart of expensive many-objective evolutionary algorithm based on Equivariate Component Analysis.

algebra EMax, and using Latin hypercube sampling [24]. The population is then evaluated using the true function. In the main loop, firstly, the ELM model is used to evaluate the subpopulation and then merge the parent and child populations, and select some outstanding individuals in the merged population into Pop. By building multiple ELM models and finding the set of candidate solutions to be evaluated using the multi-model joint decision making method, and then combining the distributional distance evaluation to select non-dominated individuals in the set of candidate solutions to be evaluated to update the population Pop. Judge whether the maximum evolutionary generation is reached, and if not, return to continue the operation of evaluating the subpopulation; otherwise, stop the evolution. The algorithm flow chart is shown in Figure 3.

1) CANDIDATE SOLUTION SELECTION

In order to improve the distribution and convergence speed of ECA-EMO, an evaluation strategy and model management method of new candidate solutions based on multi-model common decision support were designed based on the high accuracy of ECA algorithm in Pareto dominance prediction. Using k trained models of ELM and the distributed distance method to decide the new candidate solution by voting. The classifier model can predict Pareto non-dominant individuals more accurately, so as to evaluate new candidate solutions more accurately, which can reduce the evaluation times of expensive objective function in the process of evolution, and save expensive evaluation resources. The candidate solution selection method can search the decision space of the optimization problem to a greater extent, and

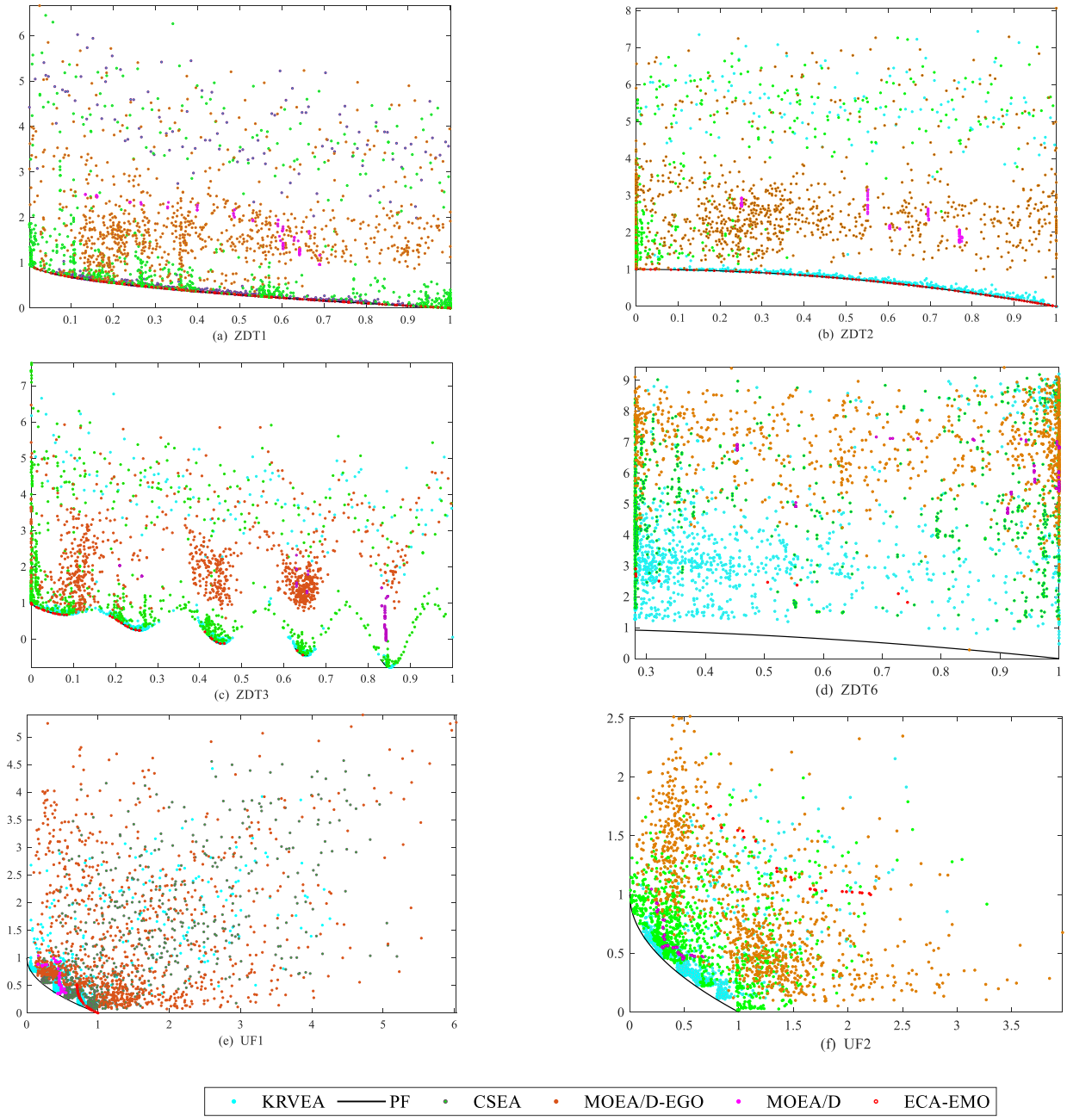

FIGURE 4. The approximate Pareto front obtained by ECA-EMO algorithm and four comparison algorithms on ZDT1,ZDT2,ZDT3,ZDT6,UF1 and UF2.

ensure that MOEAs drives the candidate solution population to approach the real Pareto optimal surface gradually. The process is: (1) The new sample is constructed by connecting the new candidate solution with the known sample; (2) K trained classification models were used to classify the new samples; (3) Statistical classification results, find the candidate solution to be evaluated, and calculate the distribution distance; (4) Evaluate the new candidate solution found in step 3.

This section analyzes the complexity of the ECA-EMO algorithm. The main computational cost of the algorithm comes from the creation of the agent model, the generation of the population and the selection of the evaluation solution. Let the population size be N and the target dimension be M. The time complexity of building the agent model is $O(MN^3)$. In the process of population generation selection and, if all individuals in the population are not dominated by each other, the time complexity of non-dominated sorting is $O(MN^2)$. In the selection of evaluation solutions, the time complexity of selecting individuals for evaluation is $O(kMN^2)$. Therefore, the time complexity of the ECA-EMO algorithm is $O(MN^3)$.

TABLE 4. ECA-EMO, KRVEA, MOEA/D, CSEA, MOEA/D-EGO algorithm average and standard deviation of IGD performance index.

Test Problem	M	Algorithm				
		ECA-EMO	KRVEA	MOEA/D	CSEA	MOEA/D-EGO
ZDT1	2	3.24e-2(2.71e-3)	2.87e-3(2.75e-4)	1.81e+0(1.33e+0)	3.41e-2(1.05e-2)	9.73e-2(7.45e-2)
ZDT2	2	5.82e-3(4.34e-4)	1.68e-2(4.05e-3)	1.64e+0(1.51e+0)	3.78e-1(3.20e-1)	1.49e-1(9.45e-2)
ZDT3	2	2.88e-2(1.61e-3)	7.59e-3(7.84e-4)	1.29e+0(9.15e-1)	7.81e-2(3.34e-2)	3.20e-1(1.01e-1)
ZDT6	2	9.06e-1(5.96e-5)	3.01e-1(1.95e-1)	7.95e+0(1.17e+0)	1.82e+0(6.85e-1)	1.62e+0(1.19e+0)
UF1	2	7.01e-1(6.20e-3)	1.05e-1(2.97e-2)	3.55e-1(8.77e-2)	1.10e-1(2.06e-2)	1.80e-1(4.32e-2)
UF2	2	7.31e-1(6.89e-3)	5.59e-2(1.75e-2)	2.04e-1(3.61e-2)	9.03e-2(1.37e-2)	1.47e-1(2.24e-2)
DTLZ2	3	5.57e-2(5.61e-2)	6.10e-2(8.86e-3)	1.20e-1(2.34e-2)	1.06e-1(7.46e-3)	3.02e-1(2.56e-2)
+/-/=			2/5/0	5/2/0	5/1/1	4/2/1

TABLE 5. ECA-EMO, KRVEA, MOEA/D, CSEA, MOEA/D-EGO algorithm average and standard deviation of HV performance index.

Test Problem	M	Algorithm				
		ECA-EMO	KRVEA	MOEA/D	CSEA	MOEA/D-EGO
ZDT1	2	5.08e-1(4.77e-1)	7.21e-1(4.46e-4)	2.19e-2(6.48e-2)	6.74e-1(1.65e-2)	6.09e-1(6.84e-2)
ZDT2	2	1.73e-1(4.61e-1)	4.33e-1(3.53e-3)	7.99e-3(2.93e-2)	2.19e-1(1.83e-1)	2.73e-1(8.07e-2)
ZDT3	2	6.88e-1(4.46e-1)	5.97e-1(6.46e-1)	1.20e-1(1.50e-1)	5.22e-1(3.66e-2)	4.19e-1(9.45e-2)
ZDT6	2	4.33e-1(3.69e-1)	5.09e-1(4.93e-2)	0.00e+0(0.00e+0)	0.00e+0(0.00e+0)	0.00e+0(0.00e+0)
UF1	2	1.84e-1(3.79e-1)	5.78e-1(3.55e-2)	3.32e-1(1.23e-1)	5.61e-1(3.58e-2)	4.62e-1(7.23e-2)
UF2	2	2.16e-1(4.65e-1)	6.49e-1(1.71e-2)	5.71e-1(5.19e-2)	5.89e-1(2.21e-2)	5.05e-1(3.20e-2)
DTLZ2	3	3.05e-1(6.18e-2)	5.34e-1(1.05e-2)	3.93e-1(4.20e-2)	5.05e-1(1.07e-2)	1.68e-1(3.67e-2)
+/-/=			6/1/0	4/3/0	5/1/1	4/2/1

B. PERFORMANCE METRICS AND PARAMETER SETTING

To test the performance of the ECA-EMO algorithm, four multi-objective algorithms, KRVEA, MOEA/D, CSEA and MOEA/D-EGO, were selected for comparison. The advantages and disadvantages of the Pareto frontier are first compared, then the parameters and performance evaluation metrics of the test functions are given, and finally the performance of the ECA-EMO algorithm is analyzed by comparing the IGD and HV metrics. The algorithms in this paper were run on the Matlab platform. All the compared algorithms were run in the multi-objective evolutionary algorithm platform PlatEMO [25].

1) TEST PROBLEM AND ALGORITHM PARAMETER SETTING

In this paper, the ZDT,UF and DTLZ series of test functions [26] are selected instead of the real functions for expensive multi-objective optimization problems. The number of objectives for the ZDT and UF series test function is 2, the number of decision variables is 10, and the population size is 100. The number of objectives for the DTLZ series test function is 3, the number of decision variables is 10, and the population size is 100. The symbols '+' in Tables 3 and 4 indicate that the ECA-EMO algorithm results are better than the comparison algorithm, the symbol '-' indicates that the ECA-EMO algorithm results are inferior to the comparison algorithm, and the symbol '=' indicates that the difference between the comparison algorithm results and the ECA-EMO algorithm is small.

In this paper, 2 metrics, IGD and HV, are chosen to measure the performance of the algorithm.

1) IGD Indicator

The IGD metric [27] is a composite metric to evaluate the convergence and diversity of the algorithm, and its

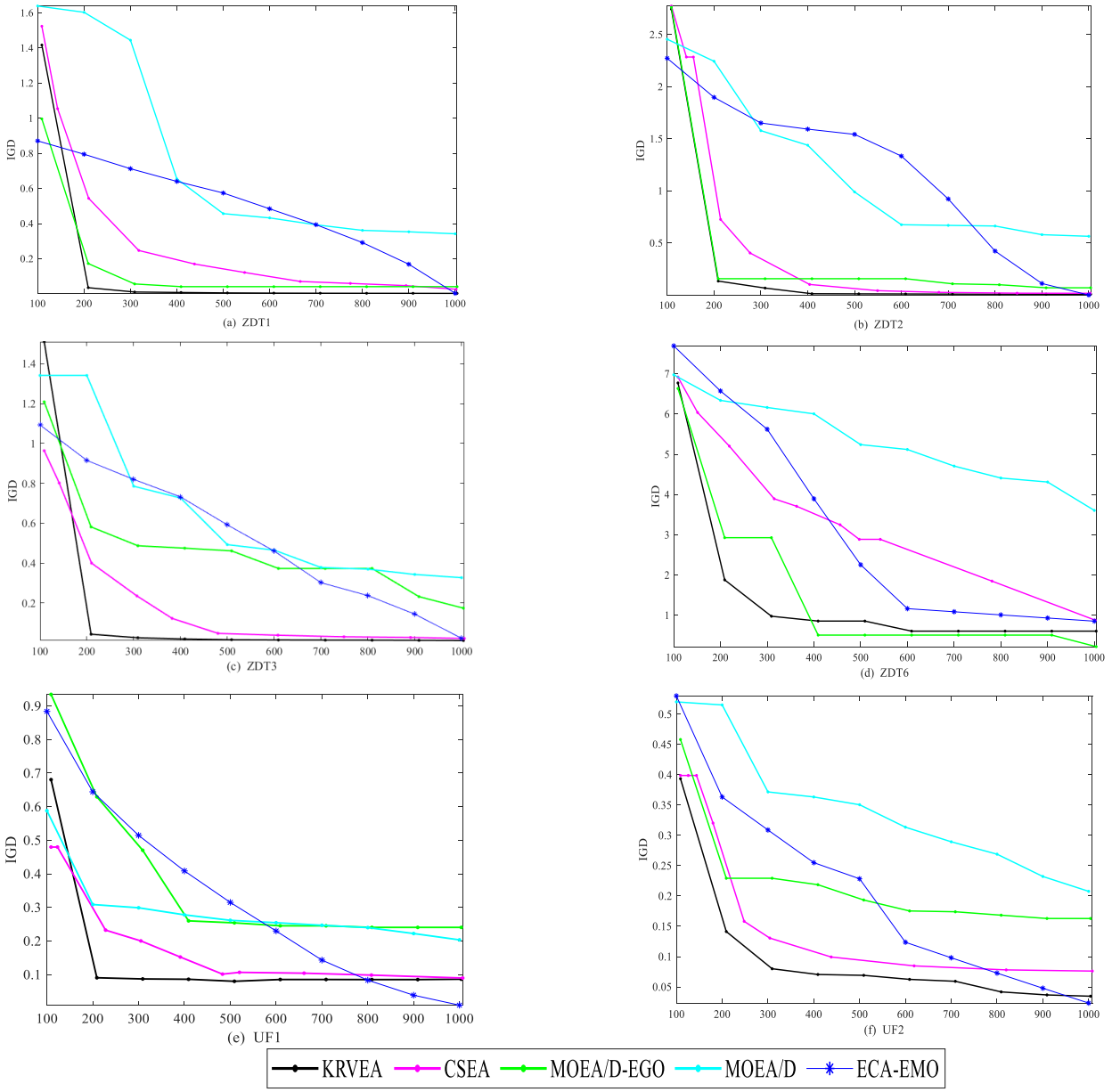

FIGURE 5. The change trend of IGD values between ECA-EMO algorithm and four comparative algorithms on ZDT1,ZDT2,ZDT3,ZDT6,UF1 and UF2.

function is defined as:

$$IGD = \frac{(\sum_{i=1}^{n} d_i)}{n} \tag{4}$$

where n denotes the number of individuals in PF_{true} and d_i denotes the Euclidean distance of each individual obtained in the target space from the nearest individual in PF_{true}. Smaller IGD values indicate better convergence and diversity of the algorithm.

2) HV Indicator

The HV metric [28] is a metric to evaluate the convergence and diversity of the algorithm, and its function is defined as:

$$HV = \delta(Y_{i=1}^{|S|} v_i) \tag{5}$$

where δ denotes the Lebesgue measure [29], $|S|$ denotes the number of non-dominated solution sets, and v_i denotes the hypervolume composed of the reference point and the i-th non-dominated solution. A larger HV value indicates a better overall performance of the algorithm.

C. SIMULATION EXPERIMENT RESULTS AND ANALYSIS

The comprehensive performance of the ECA-EMO algorithm is verified by comparing with the simulation experiments of KRVEA [30], MOEA/D [31], CSEA [32] and MOEA/D-EGO [7] algorithms. The comprehensive performance of the algorithm is evaluated by first comparing the IGD and HV values of each algorithm on solving the ZDT,

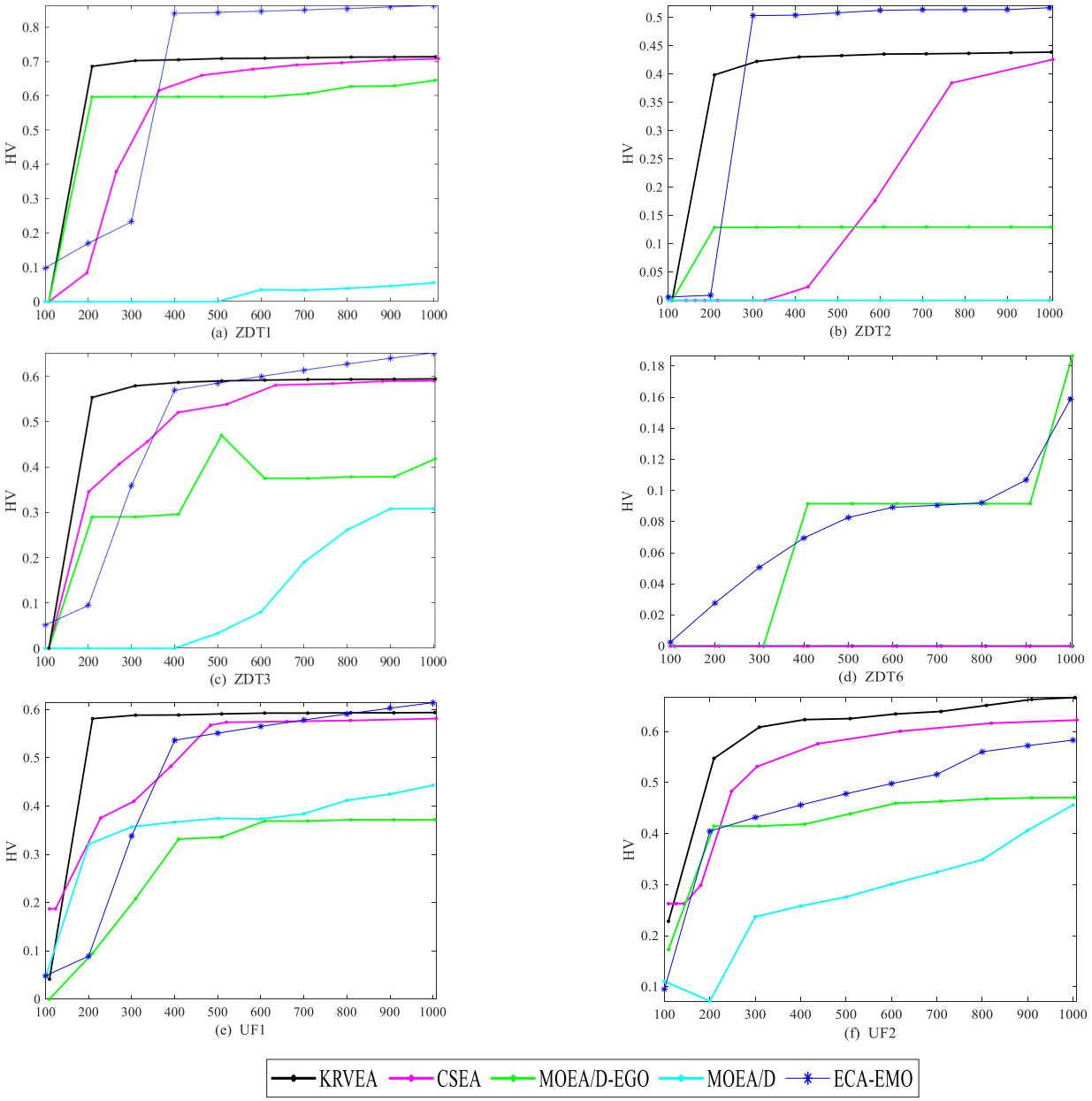

FIGURE 6. The change trend of HV values between ECA-EMO algorithm and four comparative algorithms on ZDT1,ZDT2,ZDT3,ZDT6,UF1 and UF2.

UF and DTLZ test functions. The convergence performance of the algorithm is then evaluated by comparing the effect of each algorithm in approximating the real frontier of Pareto. For each test function, the number of evaluations is set to 100, and each algorithm is run 20 times.

1) PARETO FRONTIER COMPARATIVE EXPERIMENTS AND ANALYSIS

The approximation effects of the five algorithms on the real Pareto front for solving ZDT1, ZDT2, ZDT3, ZDT6, UF1 and UF2 problems are shown in Figure 4. The solutions obtained by the other three comparison algorithms are mostly far from the true Pareto front; on problem UF1, the solutions of the ECA-EMO algorithm do not fully converge to the true Pareto front, but the other four comparison algorithms are farther from the true Pareto front. It can be concluded that the ECA-EMO algorithm has the best approximation of the true frontier, which indicates that the convergence of the ECA-EMO algorithm is better than the other comparison algorithms.

2) IGD AND HV VALUE COMPARISON EXPERIMENT AND ANALYSIS

Tables 4 and 5 record the results of IGD values and HV values obtained by ECA-EMO and the four comparison algorithms under the ZDT, UF and DTLZ test functions, respectively, containing the mean and standard deviation of IGD and HV values. From Table 4, we can get that the IGD value of

the ECA-EMO algorithm are all better than the other algorithms under the test problem ZDT, which indicates that the convergence and diversity of the solution set of the ECA-EMO algorithm are better. It is only slightly worse than the other algorithms for the UF problem. From Table 5, it can be concluded that the ECA-EMO algorithm has better HV value on UF and ZDT6 problems, slightly inferior to the MOEA/D algorithm on ZDT1 and ZDT2 problems, but slightly worse than other comparative algorithms on test problems ZDT1 and ZDT3, indicating that the algorithm has better overall performance. Equivalent components can learn spatial features more fully and ELM has better generalization ability, so the ECA-EMO algorithm proposed in this paper can achieve better results.

The trends of the IGD performance metrics of the ECA-EMO algorithm and the other comparison algorithms for the ZDT1, ZDT2, ZDT3, ZDT6, UF1, and UF2 problems are shown in Figure 5. At the beginning of the ZDT1 and ZDT2 problems, the IGD metrics of the ECA-EMO algorithm are lower than all the other comparison algorithms, and the IGD values of the ECA-EMO algorithm keep decreasing at a uniform rate. At the end of the evaluation for all problems except ZDT6, the IGD values of the ECA-EMO algorithm were lower than those of the comparison algorithms. On problems ZDT3, ZDT6 and UF1, the IGD values of CSEA, KRVEA, MOEA/D-EGO and MOEA/D algorithms fluctuated greatly and were unstable under each test problem; and four algorithms, CSEA, KRVEA, MOEA/D-EGO and MOEA/D, fell into local optimum in the middle and late stages of evaluation. It indicates that the ECA-EMO algorithm has a better comprehensive performance.

Figure 6 reflects the variation of HV metrics between the ECA-EMO algorithm and the other compared algorithms on the ZDT1, ZDT2, ZDT3, ZDT6, UF1, and UF2 problems. On the ZDT1, ZDT2, ZDT3 and UF1 problems, it can be seen that the trend of HV values of the ECA-EMO algorithm is smoother and the HV values are higher than the other compared algorithms at the later stage of evaluation. On the ZDT6 problem, only the HV values of the MOEA/D-EGO algorithm and the ECA-EMO algorithm varied, and the trend of the HV values of the ECA-EMO algorithm was smoother. This indicates the excellent performance of the ECA-EMO algorithm.

V. CONCLUSION

In this paper, an expensive multi-objective optimization algorithm (ECA-EMO) based on equivalence component analysis is proposed to improve the prediction accuracy by introducing decision space equivalence components in the Pareto dominance prediction process, which is combined with the fast running speed and generalization capability of ELM to ensure the fast and efficient algorithm. Rotational equivalence decision components are used in ELM to generate calculation of implied layer thresholds and weights to ensure the distributivity of populations. The candidate solutions to be evaluated predicted by the ELM model are combined with the weight class distributivity maintenance strategy to select some of the candidate solutions for evaluation so that the model is updated under the premise of a limited number of evaluations. From the experimental results, it is shown that the equivalent component analysis method outperforms the Pareto dominance prediction results of existing nearest neighbor classification algorithms and typical agent models, and the convergence and distributivity of the solution set obtained by the ECA-EMO algorithm is better than that of the solution set of the comparison algorithm. The next work is to use the ECA-EMO algorithm proposed in this paper to try to solve more expensive multi-objective problems.

REFERENCES

[1] Z. Chen, Y. Zhou, and X. He, "Handling expensive multi-objective optimization problems with a cluster-based neighborhood regression model," *Appl. Soft Comput.*, vol. 80, pp. 211–225, Aug. 2019.

[2] B. Liu, S. Koziel, and Q. Zhang, "A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems," *J. Comput. Sci.*, vol. 12, pp. 28–37, Jan. 2016.

[3] S. Rashidi and P. Ranjitkar, "Bus dwell time modeling using gene expression programming," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 30, no. 6, pp. 478–489, Apr. 2015.

[4] H. Wang, Y. Jin, and J. O. Jansen, "Data-driven surrogate-assisted multiobjective evolutionary optimization of a trauma system," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 939–952, Dec. 2016.

[5] R. Ariizumi, M. Tesch, K. Kato, H. Choset, and F. Matsuno, "Multiobjective optimization based on expensive robotic experiments under heteroscedastic noise," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 468–483, Apr. 2017.

[6] C. A. C. Coello, S. G. Brambila, J. F. Gamboa, M. G. C. Tapia, and R. H. Gómez, "Evolutionary multiobjective optimization: Open research areas and some challenges lying ahead," *Complex Intell. Syst.*, vol. 6, no. 2, pp. 221–236, Jun. 2019.

[7] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by MOEA/D with Gaussian process model," *IEEE Trans. Evol. Comput.*, vol. 14, no. 3, pp. 456–474, Jun. 2010, doi: 10.1109/TEVC.2009.2033671.

[8] X. Lin, Q. Zhang, and S. Kwong, "An efficient batch expensive multi-objective evolutionary algorithm based on decomposition," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1343–1349, doi: 10.1109/CEC.2017.7969460.

[9] X. Ji, Y. Zhang, D. Gong, and X. Sun, "Dual-surrogate-assisted cooperative particle swarm optimization for expensive multimodal problems," *IEEE Trans. Evol. Comput.*, vol. 25, no. 4, pp. 794–808, Aug. 2021, doi: 10.1109/TEVC.2021.3064835.

[10] J. C. Liu, Y. J. Zhao, F. Li, and Y. X. Song, "An expensive multi-objective evolutionary algorithm based on R^2 metrics," *Control Decis. Making*, vol. 35, no. 4, pp. 823–832, Apr. 2020.

[11] G. Qing-Hua, Z. Yu-Feng, L. Xue-Xian, and R. Shun-Ling, "An expensive multi-objective evolutionary algorithm based on radial space partitioning," *J. Automat.*, vol. 47, pp. 1–21, Jan. 2021.

[12] F. Li, X. Cai, and L. Gao, "Ensemble of surrogates assisted particle swarm optimization of medium scale expensive problems," *Appl. Soft Comput.*, vol. 74, pp. 291–305, Jan. 2019.

[13] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, nos. 1–3, pp. 489–501, 2006.

[14] L. M. Pavelski, M. R. Delgado, C. P. Almeida, R. A. Gonçalves, and S. M. Venske, "Extreme learning surrogate models in multi-objective optimization based on decomposition," *Neurocomputing*, vol. 180, pp. 55–67, Mar. 2016.

[15] X. Y. Cai, C. Y. Ma, F. Zhang, N. Li, H. L. Cheng, Q. Sun, Y. S. Xiao, and S. P. Li, "Adaptive decomposition-based multi-task collaborative expensive multi-objective optimization algorithm," *J. Comput. Sci.*, vol. 44, no. 9, pp. 1934–1948, Sep. 2021.

[16] Y. Cheng, W. J. Zeng, and G. Q. Guo, "Predicting D-Pareto dominance in multi-objective optimization using nearest neighbor classification method," *Appl. Res. Comput.*, vol. 30, no. 12, pp. 3571–3575, Dec. 2013.

[17] G. Guanqi, Y. Cheng, Z. Wenjing, L. Wu, and Y. Taishan, "Pareto dominance prediction based on cross-similarity of equivalent components," *J. Automat.*, vol. 40, no. 1, pp. 33–40, Jan. 2014.
[18] W.-B. Li, J.-J. He, C.-Y. Feng, and G.-Q. Kuan, "Pareto dominance prediction based on decision space transform nearest neighbor method," *J. Automat.*, vol. 43, no. 2, pp. 294–301, Feb. 2017.
[19] W. B. Li, J. J. He, G. Q. Guo, C. Y. Feng, and L. Pan, "Multi-objective optimization of Pareto merit prediction based on correlation analysis," *J. Electron.*, vol. 45, no. 2, pp. 459–467, Feb. 2017.
[20] Y. Li and Z. Yang, "Application of EOS-ELM with binary Jaya-based feature selection to real-time transient stability assessment using PMU data," *IEEE Access*, vol. 5, pp. 23092–23101, 2017, doi: 10.1109/ACCESS.2017.2765626.
[21] Y. Li, J. Wang, D. Zhao, G. Li, and C. Chen, "A two-stage approach for combined heat and power economic emission dispatch: Combining multi-objective optimization with integrated decision making," *Energy*, vol. 162, pp. 237–254, Nov. 2018.
[22] N. Li, B. Liu, and W. Wang, "Optimization algorithm based on single hidden layer feedforward neural network," *Sci., Technol. Eng.*, vol. 19, no. 1, pp. 136–141, Jan.2019.
[23] G. Guo, C. Yin, T. Yan, and W. Li, "Nearest neighbor classification of Pareto dominance in multi-objective optimization," in *Proc. IEEE 5th Int. Conf. Adv. Comput. Intell. (ICACI)*, Oct. 2012, pp. 328–331, doi: 10.1109/ICACI.2012.6463179.
[24] H. Liu and Q. Liu, "An improved MOPSO algorithm based on Latin hypercube sampling and cluster analysis," *Foreign Electron. Meas. Technol.*, vol. 39, no. 3, pp. 15–19, 2020.
[25] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Nov. 2017, doi: 10.1109/MCI.2017.2742868.
[26] E. Yap, M. A. Muñoz, and K. Smith-Miles, "On the diversity and robustness of parameterised multi-objective test suites," *Appl. Soft Comput.*, vol. 110, Oct. 2021, Art. no. 107613.
[27] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 807–813, Oct. 2016, doi: 10.1109/TEVC.2015.2505784.
[28] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006, doi: 10.1109/TEVC.2005.861417.
[29] H. Wang, T. Cai, K. Li, and W. Pedrycz, "Constraint handling technique based on Lebesgue measure for constrained multiobjective particle swarm optimization algorithm," *Knowl.-Based Syst.*, vol. 227, Sep. 2021, Art. no. 107131.
[30] L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 74–88, Feb. 2019, doi: 10.1109/TEVC.2018.2802784.
[31] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007, doi: 10.1109/TEVC.2007.892759.
[32] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, Feb. 2018, doi: 10.1109/TEVC.2016.2622301.

LI WENBIN was born in 1981. He received the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 2020. His research interests include industrial process control, evolutionary computation, and multi-objective optimization.

CHAI ZI'AN was born in 1999. He is currently pursuing the master's degree with the Hunan Institute of Technology. His main research interest includes expensive multi-objective optimization.

LIU GANG was born in 1983. He received the Ph.D. degree from the National University of Defense Technology, China, in 2013. His research interests include intelligent information processing, optimization, and path planning.