Mathematical Biosciences and Engineering

MBE, 20(11): 19839–19857.
DOI: 10.3934/mbe.2023878
Received: 24 June 2023
Revised: 19 October 2023
Accepted: 24 October 2023
Published: 31 October 2023

http://www.aimspress.com/journal/MBE

Research article

An active learning Gaussian modeling based multi-objective evolutionary algorithm using population guided weight vector evolution strategy

Xiaofang Guo*, Yuping Wang and Haonan Zhang

School of Sciences, Xi'an Technological University, Xi'an 710000, China

* **Correspondence:** Email: guoxiaofang@xatu.edu.cn.

Abstract: The inverse model based multi-objective evolutionary algorithm (IM-MOEA) generates offspring by establishing probabilistic models and sampling by the model, which is a new computing schema to replace crossover in MOEAs. In this paper, an active learning Gaussian modeling based multi-objective evolutionary algorithm using population guided weight vector evolution strategy (ALGM-MOEA) is proposed. To properly cope with multi-objective problems with different shapes of Pareto front (PF), a novel population guided weight vector evolution strategy is proposed to dynamically adjust search directions according to the distribution of generated PF. Moreover, in order to enhance the search efficiency and prediction accuracy, an active learning based training sample selection method is designed to build Gaussian process based inverse models, which chooses individuals with the maximum amount of information to effectively enhance the prediction accuracy of the inverse model. The experimental results demonstrate the competitiveness of the proposed ALGM-MOEA on benchmark problems with various shapes of Pareto front.

Keywords: active learning; multi-objective evolutionary algorithm; weight vector adjustment; Gaussian regression model; Pareto front

1. Introduction

Many practical problems in the fields of machine learning [1–4] and engineering optimization [5,6] often involve two or more optimization objectives, which are called multi-objective optimization problems (MOPs). These objective functions always conflict with each other. Therefore, it is impossible to have a unique optimal solution to optimize all objectives at the same time. For MOPs, the

ultimate goal is to find a set of compromise solutions which are non-dominated each other. One of the most widely studied MOPs is the box-constrained continuous MOP which is defined as follows.

$$\begin{aligned} &\min\ F(x) = (f_1(x), f_2(x), \cdots, f_m(x))^T \\ &s.t.\ \ x = (x_1, \cdots x_n)^T \in \Omega \end{aligned} \tag{1}$$

where x and Ω represent decision variable vector and search space, respectively. $F(x)$ denotes the objective function vector consisting of m objective functions.

As a kind of random search algorithms, evolutionary algorithm is very suitable for MOPs because of its characteristics of population-based search. Currently, multi-objective evolutionary algorithms (MOEAs) have used to solve difficult MOPs with complex shapes of Pareto front (e.g., discontinuous, degenerated or multi-modal) or with many objectives. Many branches, such as dominance modification-based approaches, indicator-based methods and decomposition-based strategies have been developed in recent years. The ultimate goal of MOEAs is to obtain a set of widely distributed Pareto optimal solutions to approach true Pareto front (PF). In particular, decomposition-based algorithm is a kind of effective algorithms due to easy execution. To design an effective decomposition-based algorithm, two factors are crucial to the quality of solutions. One is how to generate and adjust the search directions in the objective space to guide the population towards a true PF [7,8], and another is how to generate new high-quality solutions to ensure convergence and diversity [9,10].

One interest of this study focuses on dynamically generating search directions for adapting to cope with MOPs with various shapes of PF. In general, the search vectors in objective space can be represented by a number of weight vectors, which divide the MOP into several sub-problems of single-objective optimization. The proper setting of the weight vectors helps to generate the high quality non-dominated solution set [11]. If the PF shape is regular, uniformly distributed weight vectors have the potential to obtain uniformly distributed solutions. However, if the PF shape is irregular, it becomes very difficult to generate appropriate weight vectors to ensure the diversity of solutions. So far, lots of weight vector generation and adjustment strategies have been developed to adjust the weight vectors for different shapes of PF, such as the schemes of random or predefined adjustment, fitting-based adjustment and adaptive adjustment. Serafini [12] used randomly modified reference vectors to guide the search. Jin et al. [13] proposed predefined weight vectors which can be adjusted periodically in a fixed way to guide the population evolution. Gu et al. [14] adopted fitting or interpolation strategy to approximate the shape of PF, and re-sampled the uniformly distributed weight vectors on the estimated Gu et al. [15] designed reference points generation approach by training self-organizing mapping (SOM) network. Wang et al. [16] co-evolved the population and the weight vectors. Furthermore, some authors propose the weight vector adjustment strategies based on the information of local population or local archive. They identify the validity of the weight vectors with the help of the relation between the current weight vector and the solution, and dynamically adjust the weight vectors by a deleting-and-adding operator. Li et al. [8] used an archive and added new weight vectors in unexplored promising regions after deleting poorly invalid weight vectors. Yi et al. [17] adopted maximum-vector-angle-first principle to sequentially choose individual as weight vectors. Ge et al. [18] regenerated new reference vectors in the neighborhood of valid reference vectors combined with reference point incremental learning. These methods constantly update the weight vector with the help of local information of population, which have shown their effectiveness for many problems with certain kinds of irregular PF problems. We plan to adopt a learning strategy to dynamically adjust the distribution

of weight vectors based on population information.

Another interest of this study is the reproduction strategy. Two kinds of approaches are adopted to generate offspring, namely a genetic-based operator and estimation distribution-based algorithms (EDAs). In genetic based MOEAs, simulated binary crossover (SBX), particle swarm algorithm (PSO), differential evolution (DE) and hybrid operators are often used to generate new individuals, which are commonly adopted in most MOEAs. Besides, EDAs are a specific framework, which initially collect relevant statistical information of selected solutions and establish probability models to generate new solutions by sampling. Different from genetic based operators, EDA-based methods [19–21] can better estimate the distribution of population via statistical information, and thus can be easy to create offspring in promising locations. Laumanns et al. [22] made use of the relationship of decision variables and developed a binary decision tree-based Bayesian optimization model. Karshenas et al. [23] established a joint probabilistic model of Bayesian network employing the information of objectives and variables. Zhang et al. [24] took advantage of the regularity property of MOPs, and constructed a probability distribution of candidate solutions by using local principal component analysis (PCA). Cheng et al. [25] utilized the statistical information of population and established Gaussian process-based inverse models (IM-MOEA) from objective space to decision space to generate offspring. In the procedure of reproduction, once the new test points in objective function space are given as input, the established Gaussian regression model will generate new points in decision space as offspring. Recently, many scholars have made several variants to IM-MOEA. Farias et al. [26] adopted decomposition framework for evolution. A two-stage environmental selection [27] was incorporated with IM-MOEA to improve the both convergence and diversity of population. Zhang et al. [28] proposed a non-random grouping strategy to enhance the accuracy of models.

Although IM-MOEA and their variants perform well on MOPs with regular PFs, there exist some aspects for improvement. First, a number of evenly distributed weight vectors are adopted to partition the population into sub-problems in IM-MOEA. However, if the shape of the true Pareto front is not regular (e.g., discontinuous, degenerate, inverted PF or PF with long tail and peak), the evenly distributed weight vectors cannot guarantee the uniform partition of the objective space, thereby affecting the uniformity and diversity of solutions. Second, in the process of establishing Gaussian process based inverse model, the training samples are selected randomly, and this will affect the accuracy of the prediction model to some extent.

To address the above issues, the proposed study aims to develop a learning-to-adjustment scheme that dynamically adjusts the reference vectors in objective space, and establish more accurate Gaussian regression models to estimate the location of true Pareto optimal set. An active learning Gaussian modeling based multi-objective evolutionary algorithm using population-guided weight vector evolution strategy (ALGM-MOEA) is proposed. The main contribution focuses on the following two aspects.

1) To properly cope with multi-objective problems with different shapes of Pareto front (PF), a new population-guided weight vector evolution strategy is developed, which dynamically adjusts the weight vector according to the distribution of generated PF.

2) To effectively enhance the prediction accuracy of the inverse model, an active learning-based training sample selection method is designed to build Gaussian process based inverse models, which selects individuals in the population with the maximum amount of information, and this helps to improve the prediction accuracy of established models and generate satisfactory offspring.

The rest of the paper is organized as follows. The basic framework of IM-MOEA is described in

Section 2. In Section 3, we elaborate the proposed algorithm. Section 4 presents experimental comparisons of the proposed algorithm with the state-of-the-art algorithms. Finally, the conclusion and future work are put in Section 5.

2. Related works

Estimation of distribution algorithm (EDA) utilizes the statistical information from current individuals and establishes the probability distribution model reflecting the geometric shape of Pareto front or Pareto set. Subsequently, sampling is executed according to the established model to generate new offspring.

Algorithm 1. The framework of IM-MOEA

Initialization: Initialize population $P(0)$ at random, and divide the population into K sub-problems by reference vectors;
while termination condition is not satisfied **do**
  Divide the population $P(t)$ into K sub-populations, and use non-dominated sorting and environmental selection to generate K sub-parent populations $P^1(t), P^2(t), \cdots, P^K(t)$.
for $k = 1$ to K **do**
  for $j = 1$ to m **do**
    Inverse Modeling: use random grouping to assign L decision variables to f_j, and L Gaussian regression models are built for estimating $IM(x_i \mid f_j)$.
    Reproduction: Sample in the objective space and generate new offspring by inverse model, and perform mutation on the sampled candidate solutions;
endfor
  endfor
  Update the combined population
  $t = t + 1$
endwhile

IM-MOEA can be classified as a type of estimation of distribution algorithms, which aims to build Gaussian regression inverse models (GPR) to approximate the Pareto front. In order to simplify the modeling process, the population is partitioned into K sub-populations by a number of evenly distributed weight vectors after calculating the position information of each individual and its neighboring reference weight vector.

In each sub-population, the operators including environment selection, inverse modeling and reproduction are executed independently. Given a MOP with m objective and n decision variables, it is difficult to train a $m-input-and-n-output$ multivariate model to approximate the relation from objective vectors to decision vectors. Instead, a group of decomposed univariate probability distribution models are built for each objective. To be specific, IM-MOEA takes the $j-th$ objective values and the $i-th$ decision values of all solutions in one sub-population as the training data, denoted by $T_{j,i} = [f_j, x_i]$, ($f_j = (f_j^1, \cdots f_j^{Nt})^T, x_i = (x_i^1, \cdots x_i^{Nt})^T$, N_t represents the population size used to train the model), and then constructs the conditional probability distribution $P(x_i \mid f_j)$ to reflect the mapping relationship between them, which can be estimated by training a Gaussian process regression inverse model $IM(x_i \mid f_j)$.

In each model group for one objective f_j, random grouping strategy is adopted to assign a part of L ($L \ll n$) decision variables, that is, $(x_{j1},\cdots x_{jL})$, to establish L Gaussian process regression models $IM(x_{j1} \mid f_j),\cdots, IM(x_{jL} \mid f_j)$, while the rest of $(n-L)$ decision variables will remain unchanged. For example, if the decision variables x_1 and x_2 are assigned to objective f_1, then two models $IM(x_1 \mid f_1)$ and $IM(x_2 \mid f_1)$ are built. The framework is presented in Algorithm 1 [25].

In calculation, $IM(x_i \mid f_j)$ is estimated by a Gaussian distribution $N(\mu_{j,i},(\sigma_{j,i})^2)$, which formulated in (2). Using this trained Gaussian regression model, we can input the $j-th$ objective values $f_j = (f_j^1,\cdots f_j^{Nt})^T$ and the $i-th$ decision values $x_i = (x_i^1,\cdots x_i^{Nt})^T$ of all solutions in one sub-population as the training data, and a new point f_j^* sampling in the objective space as test data, and then predict the mean and variance of output x_i^* in the decision space as offspring. The detailed induction can be referred to [25].

$$\mu_{j,i} = C_*^T (C + (\sigma_n)^2 I)^{-1} x_i$$

$$(\sigma_{j,i})^2 = C_{**} - C_*^T (C + (\sigma_n)^2 I)^{-1} C_* \tag{2}$$

where I represents an identity matrix, and $C = (c_{pq})_{Nt \times Nt}$ is covariance matrix between any two-training sample f_j^p and f_j^q. Here, $c_{pq} = c(f_j^p, f_j^q)$ is the covariance parameters between any two-training sample f_j^p and f_j^q. $C_* = [c(f_j^*, f_j^1),\cdots, c(f_j^*, f_j^{Nt})]^T$ denotes a covariance parameters vector between the test input f_j^* and each element in training sample $f_j = (f_j^1,\cdots f_j^{Nt})^T$, and C_{**} is a covariance parameters of test data f_j^*.

3. Proposed algorithm

3.1. Motivation

In order to enhance the search efficiency of distribution estimation algorithm and improve the prediction accuracy of Gaussian model when dealing with complex multi-objective optimization problems with irregular shape of Pareto front, and an active learning Gaussian modeling based multi-objective evolutionary algorithm using population-guided weight vector evolution strategy (ALGM-MOEA) is proposed. We integrate a learning-to-adjustment scheme into the procedure of update weight vector and modification of Gaussian modeling. First, in order to cope with problems with complex shape of PF, we dynamically adjust the weight vectors by learning the distribution of current population, thereby ensuring that the search occurs in the potential region where the solutions locate. Second, an active learning-based training sample selection method is designed to build Gaussian process regression inverse models, which selects individuals in the population with the maximum amount of information to effectively enhance the prediction accuracy of the inverse model. The outline of the proposed algorithm ALGM-MOEA is shown in Figure 1.

In each generation of ALGM-MOEA, the weight vectors representing search direction first divide the population into K subpopulations. Subsequently, environment selection, the establishment of Gaussian process regression models and reproduction are carried out independently in each sub-problem. The offspring produced by each subpopulation will be combined with the parents to form combined population to participate in the next generation. After several generations, the proposed

ALGM-MOEA dynamically adjusts the weight vector according to the distribution of the population. It can be shown that the key components of ALGM-MOEA, including population-guided weight vector evolution strategy and active learning based Gaussian inverse model, which will be elaborated in the following.

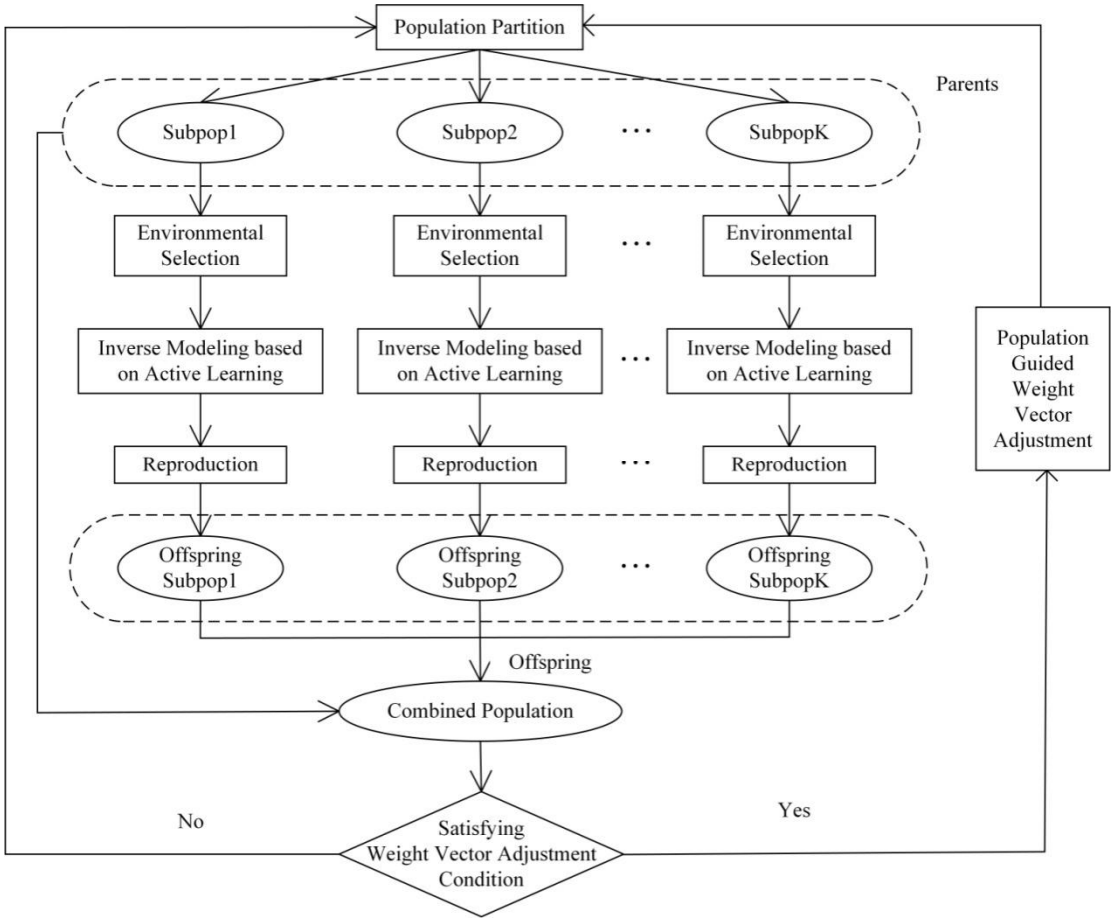

Figure 1. The overflow of the ALGM-MOEA algorithm.

3.2. Population-guided weight vector evolution strategy

In order to dynamically modify the weight vectors in MOPs with different shapes of PF, a population-guided weight vector evolution strategy is proposed. During the evolution process, the weight vectors co-evolve with the population, and the weight vectors can be dynamically adjusted by using the evaluation of the population, further guiding the evolution of the solution towards the optimal Pareto front. Specifically, we calculate the fitness of weight vector based on the number and density of individuals within its neighborhood. Individuals with high fitness are defined as effective weight vectors and survived. Weight vectors with little relevance to candidate solutions will be defined as invalid weight vectors and deleted. Thus, the selected weight vectors will be simultaneously improved with the evolution of the population.

The detail of the proposed population-guided weight vector evolution strategy is as follows. Initially, calculate the angles between each individual $s_i(i=1,\cdots,N)$ and all weight vectors $\omega_j(j=1,\cdots,K)$, which produce an angle matrix, denoted by $R=(r_{ij})_{N\times K}=(angle(s_i,w_j))_{N\times K}$. Subsequently,

for each solution s_i, we identify its neighborhood weight vectors by a predefined threshold θ. Specifically, if the angle between a solution s_i and the weight vector ω_j is less than the threshold θ, then the individual is labeled as a neighbor of the weight vector; otherwise, the solution is beyond the neighborhood, and the element r_{ij} in angle matrix R is set as Inf. Second, sort each row of angle matrix R in ascending order, and obtain another sorted matrix $R^{'} = (r_{ij}^{'})_{N\times K}$, and the calculation of $r_{ij}^{'}$ is shown in Eq (3).

$$r_{ij}^{'} = \begin{cases} \underset{j\in\{1,2,\cdots,K\}}{rank}(angle(s_i, w_j)) & angle(s_i, w_j) \le \theta \\ Inf & angle(s_i, w_j) > \theta \end{cases}. \tag{3}$$

Finally, the fitness of one weight vector w_j, termed $fit(w_j)$, is formulated in Eq (4), where $num(\underset{i\in\{1,2,\cdots,N\}\wedge r_{ij}^{'}<\inf}{r_{ij}^{'}})$ represents the number of individuals within the neighborhood of the weight vector w_j. Algorithm 2 presents the detailed implementation steps. In the operation, the weight vectors are updated every few generations.

$$fit(w_j) = 1 + num(\underset{i\in\{1,2,\cdots,N\}\wedge r_{ij}^{'}<\inf}{r_{ij}^{'}}) + \frac{1}{\sum\limits_{i\in\{1,2,\cdots,N\}\wedge(r_{ij}^{'}<\inf)} r_{ij}^{'}} \tag{4}$$

Algorithm 2. The pseudo-code of Population-guided Weight Vector Evolution Strategy

Input: Candidate solutions set S, weight vector set W; threshold of angle θ;

Output: The effective output weight vector set W_{output}

```
Calculate the angle matrix R
Construct the ranking matrix R'
for each weight vectors ω_j ∈ W do
    Calculate the fitness of weight vector fit(w_j)
endfor
Select C weight vectors with the highest fitness rank are selected as the effective weight vectors.
```

Figure 2 shows the population guided weight vector evolution strategy with a bi-objective minimization problem as an example. Initially, there are six evenly distributed weight vectors, denoted by w_1, w_2, w_3, w_4, w_5, w_6, in the objective space. Currently there are four individuals, denoted by s_1, s_2, s_3, s_4. For each individual, the angles between the individual and all weight vectors are calculated first. Then we use a threshold value θ to determine the neighborhood of each individual, and calculate its ranking value by fitness values of the weight vectors within its neighborhood. e.g., the different colored area is the neighborhood of each individual, w_5 is within the neighborhood of s_4 and s_3, w_4 is within the neighborhood of s_2 and w_3 is also within the neighborhood of s_1 and s_2. Then, calculate the fitness value of each weight vector. It is observed that w_3 and w_5 are in the neighborhood of two individuals, and gain the best two fitness function values. w_4 is in the neighborhood of one individual, and the fitness ranks the third. Finally, several weight vectors with

the highest fitness values are selected as effective weight vectors, and the rest of the weight vectors are discarded.

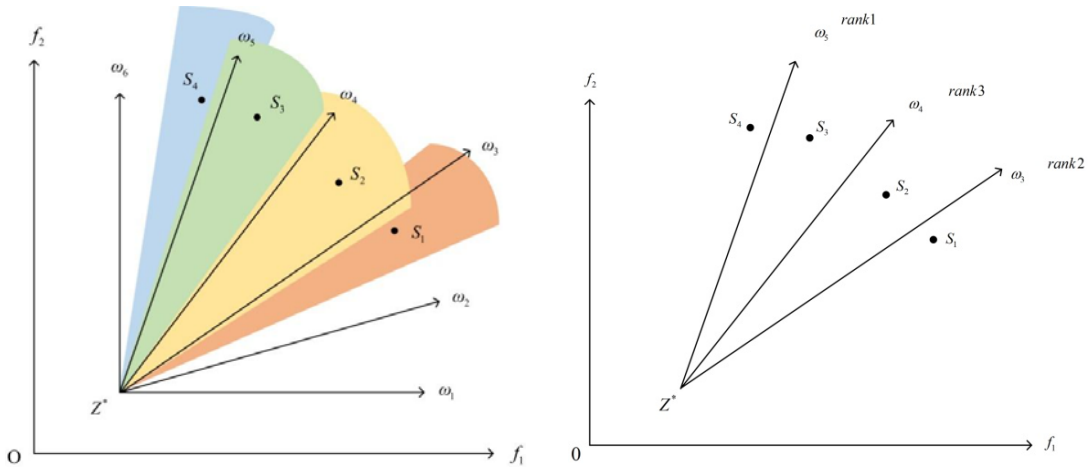

Figure 2. Illustration of population guided weight vector evolution strategy with a bi-objective minimization example.

3.3. Active learning based Gaussian inverse modeling

The generation of new individuals includes two stages, including the building of Gaussian inverse model and sampling. When evaluating the quality of generated solutions, two key factors need to be considered. One is how to enhance the accuracy of the established inverse Gaussian regression model to reflect the mapping relationship between the objective function and decision variables; the other is how to accomplish highly efficient search to balance the convergence and diversity by sampling in the objective space. To enhance the accuracy of Gaussian model and the quality of the generated individuals, a new individual generation method is developed.

In the proposed algorithm, we use active learning theory to continuously update Gaussian process regression inverse model by selecting the most uncertainty training samples. In the process of reproduction, an adaptive sampling strategy is proposed to dynamically adjust the range of the values of the test sampling points by using the established model. In the following, we will introduce active learning-based model adjustment strategy, followed by reproduction strategy.

3.3.1. Active learning-based model adjustment strategy

The quality of the training sample set directly determines the learning effectiveness of the model, and the information contained in the training samples directly affects the accuracy of the model. In order to enhance the prediction accuracy of Gaussian process regression inverse model, a sample selection method based on active learning with uncertainty is developed to choose test samples with large amount of information. As the training samples are constantly updated, the Gaussian process regression models are also continuously modified. Figure 3 presents the schematic diagram of the idea of active learning-based model adjustment strategy.

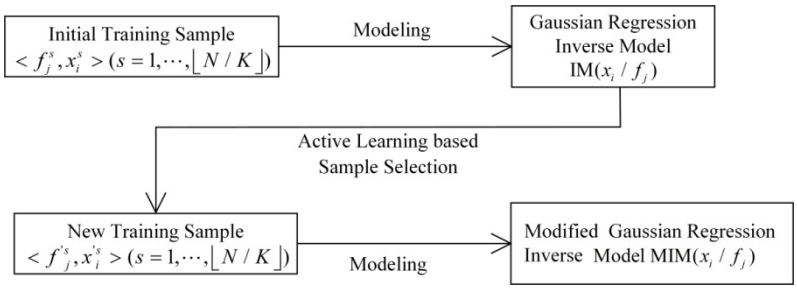

Figure 3. The schematic diagram of the idea of active learning-based model adjustment strategy.

Algorithm 3. A sample selection method based on active learning with uncertainty

Input: Alternative query test points set AQ; the current training sample set CS;

Output: New training sample pair $< y_j^{new}, x_i^{new} >$

Take the $i-th$ decision value of all individuals for consideration, select p individuals from AQ having maximum minimization distance with the current training sample set CS.

for $k = 1$ to $|\mathrm{AQ}|$ **do**

  for $t = 1$ to $|\mathrm{CS}|$ **do**

    $d_{kt}^i = \left\| x_i^k - x_i^t \right\|$

  endfor

  $d_k^i = \min_t d_{kt}^i$

endfor

for $num = 1$ to p

  Identify candidate test points with the maximum d_k^i from AQ as the test samples.

  $Index_{sel} = \{k \mid \max_k (d_k^i)\}$;

  $AQ = AQ \setminus x_{Index_{sel}}$;

endfor

Calculate the variance of each candidate set individuals in $x_{Index_{sel}}$, and select the point with the max variance as new training sample for output. $l^{new} = \max_l \{\mathrm{var}(f_j^l) \quad l \in Index_{sel}\}$,

output $< y_j^{new}, x_i^{new} >$

Initially, for a given objective function f_j and one decision variable x_i, we randomly select some individuals in the current population as the training sample set $T_{j,i} = \{< f_j^s, x_i^s > \mid s = 1, 2, \cdots, \lfloor N / K \rfloor\}$, and establish a Gaussian regression inverse model $IM(x_i \mid f_j)$. In order to enhance the prediction accuracy of established model, a new sample selection method based on active learning is developed in the proposed algorithm. We will use the active learning method to select the individuals with the most information in the population as new training samples to constantly revise the Gaussian regression model, and the modified model is denoted by $MIM(x_i \mid f_j)$.

The framework of sample selection method based on active learning with uncertainty is presented in Algorithm 3. The key step in Algorithm 3 is to select some new individuals from the current population as unlabeled individuals in objective space in accordance with the rules of maximum and minimum distance, and use the established Gaussian model to calculate the variance of these

individuals. Lines 3–6 calculate the Euclidean distance between any two individuals in the alternative query test points set AQ and the current training sample set CS, and lines 7–12 fulfill the goal of selecting new individuals by maximum and minimum distance. Line 13 calculates the variance of each candidate set individuals in $x_{Index_{sel}}$. The individuals with the largest variance will be selected as the new query samples and added to the training sample set. Since the larger the variance is, the bigger the possibility of the test sample which cannot be represented by the model obtained by the current training samples. Therefore, we can add above samples with the maximal variance to the training set as the most informative sample, and train the modified Gaussian regression inverse model by new training samples.

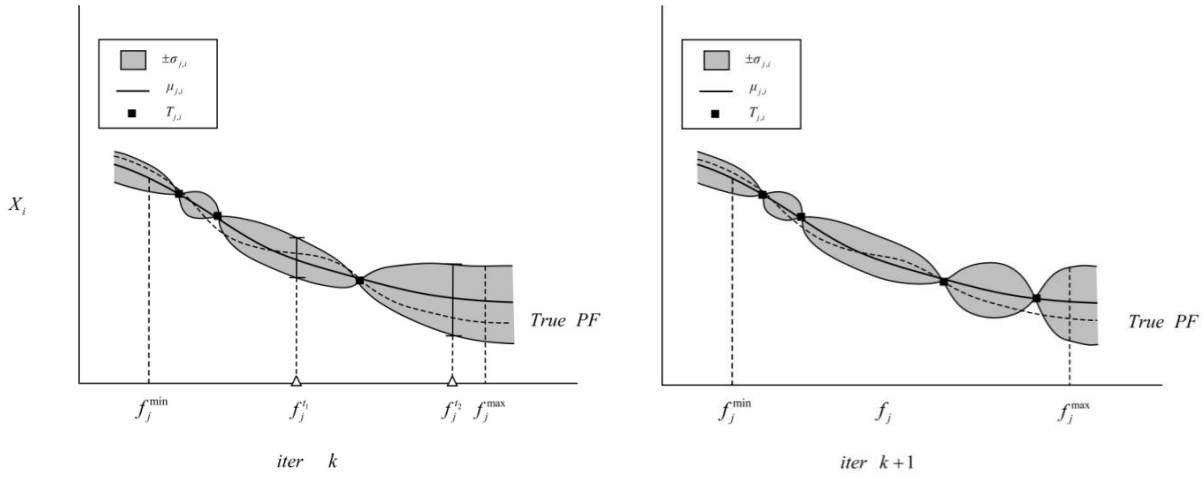

Figure 4. The example of active learning-based training sample selection process.

An example of active learning-based training sample selection process is illustrated in Figure 4. It is assumed that after the $k-th$ iteration, the Gaussian regression model is generated via current training samples, which is represented by three rectangles in the left subfigure. The mean and variance of the established Gaussian process regression model are shown, where the curve and the shaded part represent the mean and confidence interval with a confidence level of 95%. Then, the two query points f_j^{t1} and f_j^{t2} in test input set were selected according to the rules of maximum and minimum distance, represented by two triangles. According to the theory of active learning, the variance of f_j^{t1} and f_j^{t2} are calculated. It can be seen that the variance of f_j^{t2} is greater than that of f_j^{t1}, and it is the point with the largest amount of information. Then, f_j^{t2} is added to the new sample data set, and the new sample set was used to obtain the Gaussian model generated at the $(k+1)-th$ iteration. It can be seen from the right-hand side sub-figure that a new Gaussian regression model is established with the current four training samples.

3.3.2. Reproduction to generate new points

When the inverse Gaussian process regression models are obtained, we will sample the input test points f_j^* within the estimated range of the objective space, and then use the mapping relationship established by models to generate newly candidate solutions x_i^* as output.

In the proposed ALGM-MOEA, we adjust the range of estimating range dynamically [29] in different stages of the evolution. In the early stage of evolution, the search range is relatively narrow to speed up the convergence, while in the late stage of evolution, the search range will be extended to improve diversity and avoid to finding local optimal. The estimated range is calculated by Eq (5), where $f_j^{\max}$ and $f_j^{\min}$ are the upper and lower bounds of population in the $j-th$ objective, $\varepsilon=f_j^{\max}-f_j^{\min}$ and a_1 and a_2 are range coefficients set by the decision maker. gen is the current generation, and the threshold for range adjustment α_1 is set to 500.

$$range=\begin{cases}[f_j^{\min}-a_1\varepsilon, f_j^{\max}+a_1\varepsilon] & \text{gen}\le\alpha_1\\ [f_j^{\min}-a_2\varepsilon, f_j^{\max}+a_2\varepsilon] & \text{gen}>\alpha_1\end{cases} \tag{5}$$

4. Numerical experiments

4.1. Compared algorithms, test cases and performance metrics

In the experiments, we compare the performance of the proposed ALGM-MOEA with six algorithms, including IM-MOEA [25], RM-MEDA [24], MOEA/D [31], RVEA [30], RPD-NSGAII [32] and PREA [33]. IM-MOEA and RM-MEDA are two representative types of estimation of distribution-based algorithms. MOEA/D, RVEA and RPD-NSGAII are three popular decompositions based MOEA, which use weight vectors or adaptive reference points to divide the MOP into a set of subproblems. PREA is a promising-region-based MOEA which integrates the user's preference.

In the simulation, two types widely used test cases are employed with different characteristics of Pareto front. The first type is the modified benchmarks [25] of ZDT and DTLZ cases, named IMMOEA_F1 to IMMOEA_F9. They have various complex features, such as convex or concave with linear variable linkage or non-linear variable linkage. The two control parameters $\alpha=5$ and $\beta=3$, and the number of decision variable is 30. The second type test cases contain various irregular shape of PFs, which have degenerated PF, disconnected PF and convex multimodal PF.

4.2. Parameter settings

Table 1. Parameter settings of the compared algorithms.

Algorithms	Parameter settings
IM-MOEA	the number of reference vectors $W=30$; the group size $L=3$.
RM-MEDA	$K=5$
RVEA	$\alpha=2, f_r=0.1$
MOEA/D	$p_c=1, p_m=1, \eta_c=20, \eta_m=20$

The parameters for the compared algorithms are configured to the recommended values in their original publications, which are shown in Table 1. In the simulation, the population size is set to 300 for all algorithms, and the termination condition is set to 300,000 fitness evaluations. For the proposed ALGM-MOEA and IM-MOEA, the number of reference vectors and the group size are set to 30 and 3. In the proposed ALGM-MOEA, the range coefficient a_1 is set to 0.2 and a_2 is set to 0.5. The

threshold of angle θ is set to $\pi/18$.

4.3. Results

4.3.1. Performance comparison on IMMOEA_F1 to IMMOEA_F9

Table 2. The average of IGD for compared algorithms.

Test instances	ALGP-MOEA		IM-MOEA		RM-MEDA		PREA		RPDNSGAII		RVEA		MOEA/D
IMMOEA_F1	**1.228E-03**	+	1.697E-03	+	1.334E-03	+	1.771E-01	+	2.018E-01	+	1.292E-01	+	5.868E-02
	2.325E-04		3.320E-05		5.200E-05		3.190E-02		-5.420E-02		3.290E-02		2.950E-02
IMMOEA_F2	**1.433E-03**	+	1.812E-03	+	1.530E-03	+	6.095E-01	+	3.559E-01	+	2.162E-01	+	4.864E-02
	3.316E-04		3.950E-05		1.100E-04		7.200E-17		1.400E-01		6.280E-02		4.870E-02
IMMOEA_F3	1.249E-03	+	1.769E-03	+	6.141E-02	−	**1.169E-03**	+	2.588E-03	+	1.086E-02	+	2.028E-03
	2.114E-04		2.760E-05		8.280E-02		**1.990E-05**		2.580E-04		2.020E-03		3.800E-04
IMMOEA_F4	4.585E-02	+	5.560E-02	−	**4.349E-02**	+	3.522E-01	+	3.149E-01	+	1.291E-01	+	4.274E-01
	1.120E-02		1.410E-03		**1.200E-03**		2.130E-02		8.090E-02		1.590E-02		4.070E-01
IMMOEA_F5	**8.304E-04**	+	1.788E-03	+	4.392E-03	+	1.954E-03	+	4.847E-02	+	5.207E-02	+	6.483E-02
	8.374E-05		2.600E-05		8.610E-04		1.280E-03		7.970E-03		7.380E-03		1.130E-02
IMMOEA_F6	**1.409E-03**	+	2.462E-03	+	1.173E-02	+	4.587E-03	+	5.801E-02	+	9.588E-02	+	4.854E-02
	8.487E-05		9.160E-05		7.920E-03		5.710E-03		1.840E-02		8.640E-03		2.300E-02
IMMOEA_F7	1.914E-03	+	2.177E-03	+	1.776E-02	−	**1.176E-03**	+	2.738E-03	+	8.775E-03	+	1.388E-03
	3.179E-04		5.740E-05		6.250E-03		**1.310E-05**		2.810E-04		1.360E-03		1.000E-04
IMMOEA_F8	4.251E-02	+	6.035E-02	+	8.852E-02	+	2.385E-01	+	2.544E-01	+	1.158E-01	−	**3.264E-02**
	4.615E-03		1.380E-03		6.350E-02		7.460E-02		6.620E-02		8.660E-03		**5.860E-04**
IMMOEA_F9	3.115E-03	−	**2.379E-03**	+	8.085E-02	+	2.492E-03	+	8.707E-03	+	3.256E-02	+	3.882E-03
	1.389E-03		**1.190E-03**		3.320E-02		2.690E-03		2.540E-03		5.390E-03		1.540E-03
(+/=/−)			(8/0/1)		(8/0/1)		(7/0/2)		(9/0/0)		(9/0/0)		(8/0/1)

The statistical results of IGD and HV of compared seven algorithms are displayed in Tables 2 and 3, and 20 independent runs for each algorithm on each test case are executed. In the table, the mean and the standard deviation for each algorithm are displayed in the first and second lines, respectively, and the best results are highlighted. The Wilcoxon rank sum test at the significance level of 0.05 is used to pairwise compare the proposed ALGM-MOEA and other algorithm in each test case. Symbol '+', '-' and '=' in front of the results indicate that the proposed ALGM-MOEA is superior to, inferior to or not significantly different from the compared algorithm. Some observations in Table 2 can be concluded below.

1) The proposed ALGM-MOEA performs the best on IMMOEA_F1, IMMOEA_F2, IMMOEA_F5 and IMMOEA_F6, while IM-MOEA obtains the best mean value of IGD on IMMOEA_F9. Besides, promising-region-based PREA which incorporates user preferences obtains the best IGD mean on test questions IMMOEA_F3 and IMMOEA_F7.

2) From the results, we can also observe that the performance of three distribution estimation algorithms, ALGM-MOEA, IM-MOEA and RM-MEDA are significantly outperforming that of the

rest of the four algorithms on test cases IMMOEA_F1, IMMOEA_F2 and IMMOEA_F4. The reason is that for the adopted test cases IMMOEA_F1-F9 with correlation between decision variables, the probabilistic models built by estimation distribution algorithms is very effective for generating offspring in the preference area by sampling, and maintains better convergence and diversity.

3) It can also be seen that the IGD mean value of ALGM-MOEA is superior to that of IM-MOEA, especially in problems IMMOEA_F5, F6 and F8, where the relationship between decision variables is non-linear. This verifies that the proposed active learning-based training sample selection method can improve the prediction accuracy of Gaussian model and enhance the quality of solutions.

Table 3 summarizes the results of HV of seven algorithms. The best results on IMMOEA_F2, IMMOEA_F4, IMMOEA_F5 and IMMOEA_F6 are performed by ALGM-MOEA, while IM-MOEA wins on two test cases, IMMOEA_F3 and IMMOEA_F9. The decomposition-based algorithm MOEA/D obtained the best HV values on IMMOEA_F7 and IMMOEA_F8.

Table 3. The average of HV for compared algorithms.

Test instances	ALGP-MOEA		IM-MOEA		RM-MEDA		PREA		RPDNSGAII		RVEA		MOEA/D
IMMOEA_F1	8.745E-01	+	8.739E-01	−	**8.747E-01**	+	6.113E-01	+	5.713E-01	+	7.522E-01	+	8.249E-01
	4.660E-04		9.780E-05		**1.770E-04**		2.020E-02		2.510E-02		1.610E-02		2.050E-02
IMMOEA_F2	**5.408E-01**	+	5.402E-01	=	5.407E-01	+	9.091E-02	+	1.716E-01	+	2.857E-01	+	4.665E-01
	6.691E-04		7.000E-05		4.130E-04		3.600E-17		7.200E-02		4.780E-02		5.970E-02
IMMOEA_F3	4.333E-01	=	**4.342E-01**	+	3.627E-01	+	3.904E-01	+	3.872E-01	+	4.196E-01	+	4.323E-01
	5.036E-04		**7.380E-05**		9.510E-02		1.230E-04		7.130E-04		3.030E-03		7.050E-04
IMMOEA_F4	**7.384E-01**	+	7.096E-01	+	7.352E-01	+	4.094E-01	+	3.915E-01	+	5.668E-01	+	4.364E-01
	1.752E-02		3.010E-03		3.900E-03		3.100E-03		5.390E-02		3.080E-02		2.820E-01
IMMOEA_F5	**8.751E-01**	+	8.729E-01	+	8.687E-01	+	7.217E-01	+	6.713E-01	+	8.013E-01	+	8.212E-01
	1.634E-04		8.250E-05		1.650E-03		1.820E-03		6.020E-03		8.290E-03		7.700E-03
IMMOEA_F6	**5.416E-01**	+	5.396E-01	+	5.345E-01	+	4.425E-01	+	3.689E-01	+	3.818E-01	+	4.603E-01
	2.815E-05		1.320E-04		1.000E-02		9.890E-03		1.790E-02		1.060E-02		2.880E-02
IMMOEA_F7	4.332E-01	−	4.337E-01	+	4.148E-01	+	3.904E-01	+	3.870E-01	+	4.229E-01	−	**4.339E-01**
	9.350E-04		6.650E-05		7.920E-03		4.950E-05		5.320E-04		1.860E-03		**2.560E-04**
IMMOEA_F8	7.468E-01	+	6.889E-01	+	7.088E-01	+	4.347E-01	+	4.222E-01	+	5.871E-01	−	**7.517E-01**
	5.691E-03		3.130E-03		3.960E-02		2.280E-02		2.290E-02		1.560E-02		**1.830E-03**
IMMOEA_F9	8.715E-01	−	**8.726E-01**	+	7.468E-01	+	7.212E-01	+	7.121E-01	+	8.239E-01	+	8.689E-01
	2.197E-03		**2.290E-03**		5.170E-02		3.920E-03		3.630E-03		8.370E-03		2.670E-03
(+/=/−)			(6/1/2)		(7/1/1)		(9/0/0)		(9/0/0)		(9/0/0)		(7/0/2)

The non-dominated solutions with the best IGD values obtained by the ALGM-MOEA and the state-of-art compared algorithms on IMMOEA_F1 and IMMOEA_F6 are displayed in Figures 5 and 6. On IMMOEA_F1 with convex shape of Pareto front, the three algorithms based on estimation distribution (ALGM-MOEA, IM-MOEA and RM-MEDA) achieve good convergence and distribution. The non-dominated solution sets obtained by the remaining four algorithms only cover part of the PF, and the distribution is not good. On IMMOEA_F6 with concave shape of Pareto front, the solutions achieved by ALGM-MOEA show the best convergence and diversity, and IM-MOEA and PREA obtain

the second-best performance. The quality of the solutions obtained by the remaining algorithms cannot guarantee convergence and diversity. In test instance IMMOEA_F6, the correlation relationship between the decision variables is nonlinear. The sample selection method based on active learning in the proposed algorithm can more effectively get information between the decision variables and the objective function, and obtain offspring with high quality by Gaussian inverse model.

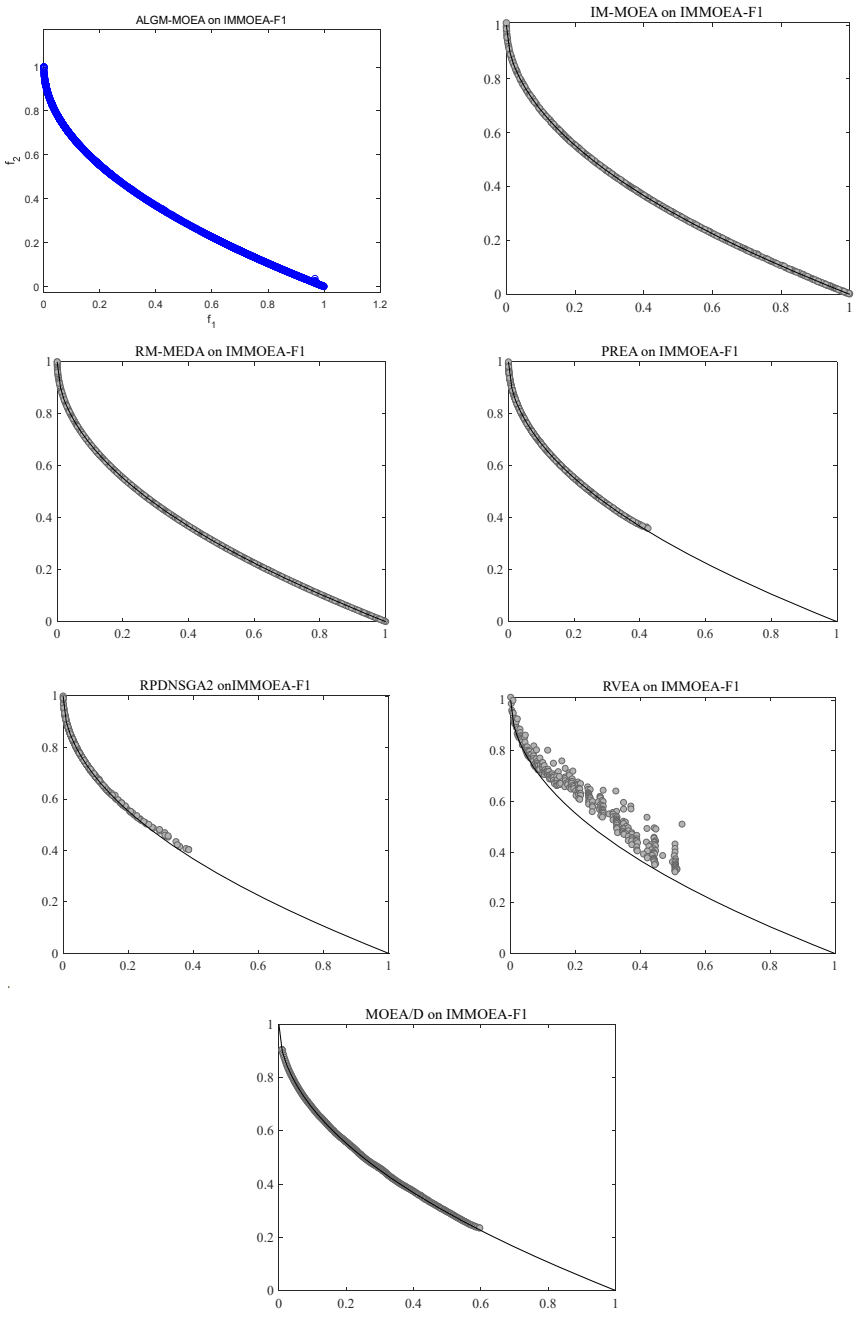

Figure 5. Non-dominated solutions with the best IGD values obtained by each algorithm in objective space on IMMOEA_F1.

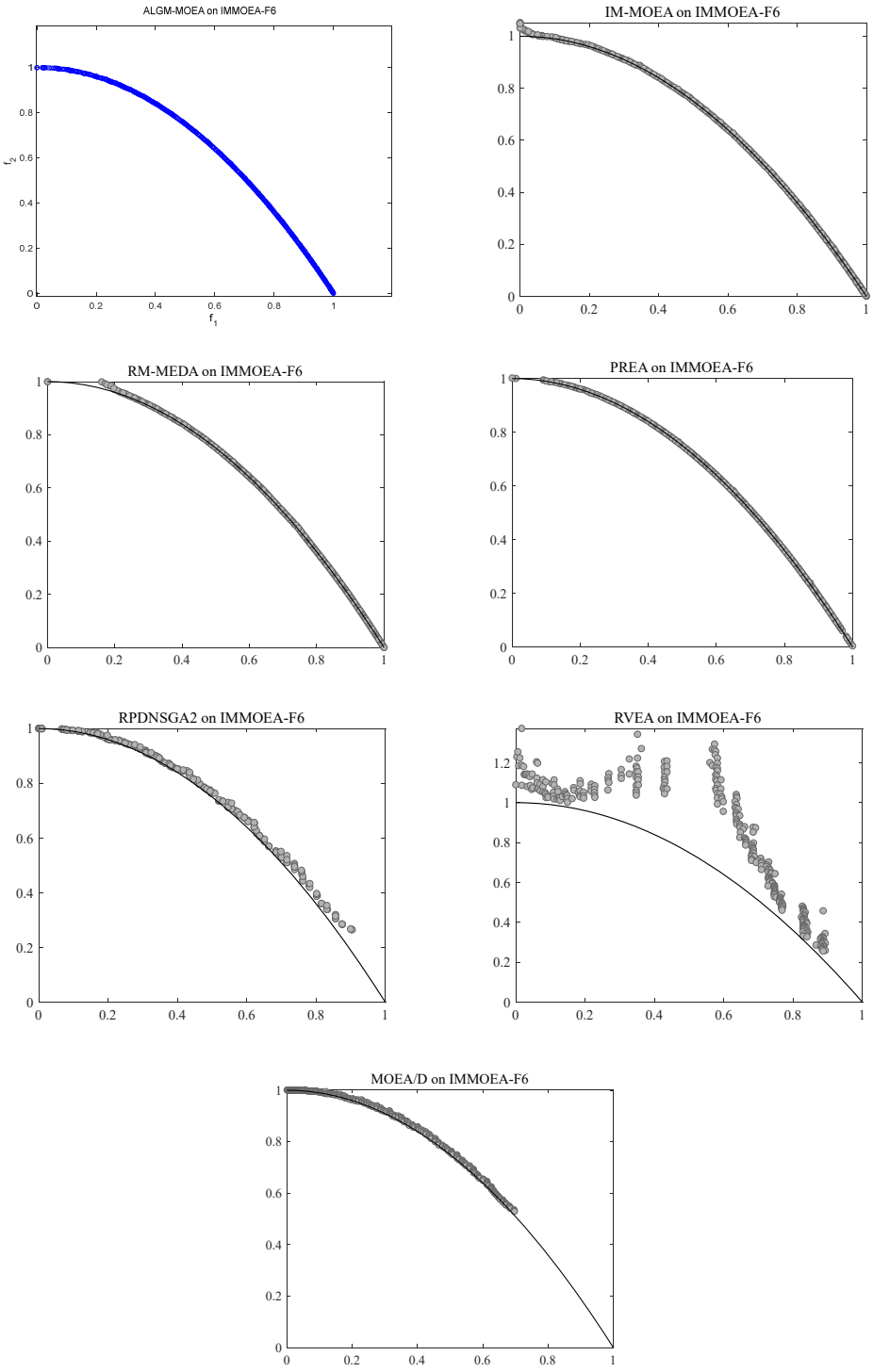

Figure 6. Non-dominated solutions with the best IGD values obtained by each algorithm in objective space on IMMOEA_F6.

4.3.2. The performance comparison of ALGM-MOEA and IM-MOEA on MOPs with irregular PFs

In order to verify the performance of the proposed ALGM-MOEA on MOPs with various irregular shape of PFs, we compared the average of IGD and HV of IM-MOEA and ALGM-MOEA on five test

cases, which have degenerated PF, disconnected PF and convex multimodel PF. The characteristics of test cases are summarized in Table 4.

Table 4. The properties of test cases with irregular PFs.

Problems	Properties
DTLZ5	Concave, degenerated
DTLZ7	disconnected
MaF5	Convex, multimodal
MaF6	Concave, degenerated
MaF7	Mixed, disconnected, multimodal

Table 5. The statistical results of IGD on test cases with irregular PFs.

Test cases	ALGM-MOEA	IM-MOEA
DTLZ5	1.0291e-2 (1.98e-3) -	**6.2741e-3 (5.00e-4)**
DTLZ7	**1.162e-1 (3.45e-2)** +	1.6573e-1 (1.52e-2)
MaF5	**1.985e-1 (5.84e-3)** +	2.0036e-1(6.45e-3)
MaF6	**8.9997e-3 (1.69e-3)** +	9.5047e-3 (9.82e-4)
MaF7	**1.3453e-1 (9.14e-2)** +	1.6326e-1 (1.79e-2)
+/=/-	4/0/1	

Table 6. The statistical results of HV on test cases with irregular PFs.

Test cases	ALGM-MOEA	IM-MOEA
DTLZ5	1.2893e-1(9.40e-4)-	**1.9734e-1 (5.29e-4)**
DTLZ7	**1.5154e+0(4.62e-2)+**	2.2577e-1 (6.10e-3)
MaF5	**4.7806e+1(1.46e-1)+**	4.7651e+1(1.72e-1)
MaF6	1.2805e-1(1.00e-3)-	**1.9243e-1 (1.18e-3)**
MaF7	**1.5238e+0(6.44e-2)+**	2.2723e-1 (7.43e-3)
+/=/−	3/0/2	

The statistical results of IGD and HV of IM-MOEA and ALGM-MOEA are shown in Tables 5 and 6. The Wilcoxon rank sum test at the significance level of 0.05 is used to pairwise compare the proposed ALGM-MOEA and other algorithm in each test case. Symbol '+' '-' and '=' in front of the results indicate that the proposed ALGM-MOEA is superior to, inferior to or not significantly different from IM-MOEA. It can be shown that ALGM-MOEA outperforms IM-MOEA in problems DTLZ7 and MaF7 with disconnected PFs, MaF5 with convex multimodal PF, since the population-guided weight vector evolution strategy can dynamically adjust search direction by learning the distribution of the population, and delete invalid weight vectors in the region where no solution locates. Therefore, the convergence and diversity are enhanced compared with IM-MOEA. In addition, ALGM-MOEA behaves worse than IM-MOEA on DTLZ5 with degenerated PF. This may be due to the fact that the new sample point selection strategy proposed in ALGM-MOEA pays more attention on individuals with uncertainty, and this will lead to a certain deviation in the establishment of the inverse model, thus affecting the distribution of the offspring.

5. Conclusions

This work introduces an active learning Gauss modeling based multi-objective evolutionary algorithm using population-guided weight vector evolution strategy (ALGM-MOEA). A new population-guided weight vector evolution strategy is proposed to alter the weight vectors with the knowledge of the current non-dominant solution set. Besides, we develop a Gaussian regression modeling based on active learning to generate new individuals, and an adaptive searching strategy to explore the unexplored sparse area in objective space and improve the diversity. The proposed ALGM-MOEA exhibits its competitiveness on search ability for test cases with linear or non-linear correlations between the decision variables. Moreover, the performance of ALGM-MOEA and IM-MOEA for test cases with irregular PF is compared, and it can be observed that the proposed ALGM-MOEA is superior to IM-MOEA, which verifies the effectiveness of the proposed ALGM-MOEA.

In the future, we will study how to build a more efficient new model to reflect the mapping between variables and objective functions, thus further reducing computational complexity.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by National Natural Science Foundation of China (No.62102304, 62372353).

Conflict of interest

The authors declare that there are no conflicts of interest.

References

1. Y. Xue, Y. Tang, X. Xu, J. Liang, F. Neri, Multi-objective feature selection with missing data in classification, *IEEE Trans. Emerging Top. Comput. Intell.*, **6** (2021), 355–364. https://doi.org/10.1109/TETCI.2021.3074147
2. Y. Xue, B. Xue, M. Zhang, Self-adaptive particle swarm optimization for large-scale feature selection in classification, *ACM Trans. Knowl. Discovery Data*, **13** (2019), 1–27. https://doi.org/10.1145/3340848
3. Y. Xue, X. Cai, F. Neri, A multi-objective evolutionary algorithm with interval based initialization and self-adaptive crossover operator for large-scale feature selection in classification, *Appl. Soft Comput.*, **127** (2022), 1–14. https://doi.org/10.1016/j.asoc.2022.109420
4. Y. Hu, Y. Zhang, D. Gong, Multiobjective particle swarm optimization for feature selection with fuzzy cost, *IEEE Trans. Cybern.*, **51** (2020), 874–888. https://doi.org/10.1109/TCYB.2020.3015756
5. W. Liu, A. Li, C. Liu, Multi-objective optimization control for tunnel boring machine performance improvement under uncertainty, *Autom. Constr.*, **139** (2022), https://doi.org/10.1016/j.autcon.2022.104310

6. S. Luo, X. Guo, Multi-objective optimization of multi-microgrid power dispatch under uncertainties using interval optimization, *J. Ind. Manage. Optim.*, **19** (2023), 823–851. https://doi.org/10.3934/jimo.2021208
7. R. Tanabe, H. Ishibuchi, A framework to handle multimodal multi-objective optimization in decomposition-based evolutionary algorithms, *IEEE Trans. Evol. Comput.*, **24** (2020), 720–734. https://doi.org/10.1109/TEVC.2019.2949841
8. M. Q. Li, X. Yao, What weights work for you? Adapting weights for any Pareto front shape in decomposition-based evolutionary multi-objective optimization, *Evol. Comput.*, **28** (2020), 227–253. https://doi.org/10.1162/EVCO_A_00269
9. T. Zhang, F. Li, X. Zhao, W. Qi, T. Liu, A convolutional neural network-based surrogate model for multi-objective optimization evolutionary algorithm based on decomposition, *Swarm Evol. Comput.*, **72** (2022), 101081. https://10.1016/j.swevo.2022.101081
10. J. Shen, P. Wang, H. Dong, J. Li, W. Wang, A multistage evolutionary algorithm for many-objective optimization, *Inf. Sci.*, **589** (2022), 531–549. http://10.1016/j.ins.2021.12.096
11. Y. Liu, Y. Hu, N. Zhu, K. Li, M. Li, A decomposition-based multi-objective evolutionary algorithm with weights updated adaptively, *Inf. Sci.*, **572** (2021), 343–377. https://doi.org/10.1016/j.ins.2021.03.067
12. P. Serafini, Simulated annealing for multi objective optimization problems, *Multiple Criteria Decision Making*, (1994), 283–292. https://doi.org/10.1007/978-1-4612-2666-6_29
13. Y. Jin, T. Okabe, B. Sendhoff, Dynamic weighted aggregation of evolutionary multi-objective optimization: why does it work and how?, in *Proceedings of the Genetic and Evolutionary Computation Conference*, (2001), 1042–1049.
14. F. Q. Gu, H. L. Liu, A novel weight design in multi-objective evolutionary algorithm, in *2010 International Conference on Computational Intelligence and Security*, (2010), 137–141. https://doi.org/10.1109/CIS.2010.37
15. F. Gu, Y. M. Cheung, Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm, *IEEE Trans. Evol. Comput.*, **22** (2017), 211–225. https://doi.org/10.1109/TEVC.2017.2695579
16. R. Wang, R. C. Purshouse, P. J. Fleming, Preference-inspired co-evolutionary algorithms using weight vectors, *Eur. J. Oper. Res.*, **243** (2015), 423–441. https://doi.org/10.1016/j.ejor.2014.05.019
17. X. Yi, Y. Zhou, M. Li, Z. Chen, A vector angle-based evolutionary algorithm for unconstrained many-objective optimization, *IEEE Trans. Evol. Comput.*, **21** (2017), 131–152. https://doi.org/10.1109/TEVC.2016.2587808
18. H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, et al., A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning, *IEEE Trans. Evol. Comput.*, **23** (2019), 572–586. https://doi.org/10.1109/TEVC.2018.2874465
19. T. Liu, X. Li, L. Tan, S. Song, An incremental-learning model-based multi-objective estimation of distribution algorithm, *Inf. Sci.*, **569** (2021), 430–449. https://doi.org/10.1016/j.ins.2021.04.011
20. M. Wu, K. Li, S. Kwong, Q. Zhang, J. Zhang, Learning to decompose: a paradigm for decomposition-based multiobjective optimization, *IEEE Trans. Evol. Comput.*, **23** (2019), 376–390. https://doi.org/10.1109/TEVC.2018.2865931
21. T. Liu, X. Li, L. Tan, S. Song, A novel adaptive greedy strategy based on Gaussian mixture clustering for multiobjective optimization, *Swarm Evol. Comput.*, **61** (2021), 1–43. https://doi.org/10.1016/j.swevo.2020.100815

22. M. Laumanns, J. Ocenasek, Bayesian optimization algorithms for multi-objective optimization, in *Proceedings of the 7th International Conference Parallel Problem Solving from Nature*, (2002), 298–307. https://doi.org/10.1007/3-540-45712-7_29
23. H. Karshenas, R. Santana, C. Bielza, P. Larranaga, Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables, *IEEE Trans. Evol. Comput.*, **18** (2014), 519–542. https://doi.org/10.1109/TEVC.2013.2281524
24. Q. Zhang, A. Zhou, Y. Jin, RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm, *IEEE Trans. Evol. Comput.*, **12** (2008), 41–63. https://doi.org/10.1109/TEVC.2007.894202
25. R. Cheng, Y. Jin, K. Narukawa, B. Sendhoff, A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling, *IEEE Trans. Evol. Comput.*, **19** (2015), 838–856. https://doi.org/10.1109/TEVC.2015.2395073
26. L. R. Farias, A. F. Araújo, IM-MOEA/D: an inverse modeling multi-objective evolutionary algorithm based on decomposition, in *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, (2021), 462–467.
27. J. Shen, H. Dong, P. Wang, J. Li, W. Wang, An inverse model-guided two-stage evolutionary algorithm for multi-objective optimization, *Expert Syst. Appl.*, **225** (2023), 120198. http://doi.org/10.1016/j.eswa.2023.120198
28. Z. Zhang, S. Liu, W. Gao, J. Xu, S. Zhu, An enhanced multi-objective evolutionary optimization algorithm with inverse model, *Inf. Sci.*, **530** (2020), 128–147. http://doi.org/10.1016/j.ins.2020.03.111
29. X. Guo, A multi-objective decomposition based evolutionary algorithm using adaptive search, in *17th International Conference on Computational Intelligence and Security*, 2021.
30. R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, *IEEE Trans. Evol. Comput.*, **20** (2016), 773–791. https://doi.org/10.1109/TEVC.2016.2519378
31. Q. Zhang, H. Li, MOEA/D: A multi-objective evolutionary algorithm based on decomposition, *IEEE Trans. Evol. Comput.*, **11** (2007), 712–731. https://doi.org/10.1109/ICALT.2005.160
32. M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, Y. S. Ong, A new decomposition-based NSGA-II for many-objective optimization, *IEEE Trans. Syst. Man Cybern.: Syst.*, **48** (2018), 1191–1210. https://doi.org/10.1109/TSMC.2017.2654301
33. J. Yuan, H. Liu, F. Gu, Q. Zhang, Z. He, Investigating the properties of indicators and an evolutionary many-objective algorithm based on a promising region, *IEEE Trans. Evol. Comput.*, **25** (2020), 75–86. http://doi.org/10.1109/TEVC.2020.2999100