

Surrogate-assisted push and pull search for expensive constrained multi-objective optimization problems

Wenji Li ^a, Ruitao Mai ^{a,1}, Zhaojun Wang ^{a,1}, Yifeng Qiu ^{a,1}, Biao Xu ^a, Zhifeng Hao ^b, Zhun Fan ^{c,*}

^a Department of Electronic Engineering, Shantou University, Shantou 515063, China

^b College of Science, Shantou University, Shantou 515063, China

^c Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen 518000, China

ARTICLE INFO

Keywords:

Expensive constrained multi-objective optimization
Surrogate model
Push and pull search
Bayesian active learning

ABSTRACT

In many real-world engineering optimizations, a large number of objective and constraint function values often need to be obtained through simulation software or physical experiments, which incurs significant computational costs and/or time expenses. These problems are known as expensive constraint multi-objective optimization problems (ECMOPs). This paper combines the push and pull search (PPS) framework and proposes a surrogate-assisted evolutionary algorithm to solve ECMOPs through Bayesian active learning, naming it the surrogate-assisted PPS (SA-PPS). Specifically, during the push search stage, candidate solutions are selected based on two indicators: hypervolume improvement and objective uncertainty. These aim to quickly guide the population towards the unconstrained Pareto front while ensuring diversity. During the pull search stage, the population is partitioned into many subregions through reference vectors, and different selection strategies are assigned to each subregion based on its state, aiming to guide the population towards the constrained Pareto front while ensuring diversity. Furthermore, we introduce a batch data selection strategy that utilizes Bayesian active learning to enable the surrogate model to focus on regions of interest in the pull search stage. Extensive experimental results have shown that the proposed SA-PPS algorithm exhibits superior convergence and diversity compared to 9 state-of-the-art algorithms across a variety of benchmark problems and a real-world optimization problem.

1. Introduction

In engineering practice, constrained multi-objective optimization problems (CMOPs) are commonly encountered. Without loss of generality, a CMOP can be defined as follows:

$$\begin{aligned} \min \quad & F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{s.t.} \quad & g_i(\mathbf{x}) \leq 0, \quad i \in \{1, 2, \dots, p\} \\ & h_j(\mathbf{x}) = 0, \quad j \in \{1, 2, \dots, q\} \\ & \mathbf{x} \in \mathbb{R}^D \end{aligned} \quad (1)$$

where \mathbf{x} is a D -dimensional decision vector, and $F(\mathbf{x})$ represents an m -dimensional objective vector. The expression $g_i(\mathbf{x}) \leq 0$ signifies the i th inequality constraint, while $h_j(\mathbf{x}) = 0$ denotes the j th equality constraint. To comprehensively handle both equality and inequality constraints, it is common practice to measure the feasibility of a solution using an overall constraint violation value, which is defined as

follows:

$$CV(\mathbf{x}) = \sum_{i=1}^p \max(g_i(\mathbf{x}), 0) + \sum_{j=1}^q \max(|h_j(\mathbf{x})| - \delta, 0) \quad (2)$$

where δ denotes an extremely small positive number. When $CV(\mathbf{x}) = 0$, the individual \mathbf{x} is regarded as a feasible solution; otherwise, it is considered as an infeasible solution.

In practical engineering optimizations, many CMOPs frequently entail computationally expensive or time-consuming evaluations of both objective and constraint functions. Consequently, identifying Pareto optimal solutions within a smaller number of function evaluations becomes imperative. Such problems are referred to as expensive CMOPs (ECMOPs).

Traditional evolutionary algorithms typically require a large number of evaluations of objective and constraint functions to achieve satisfactory results [1]. However, when dealing with ECMOPs, the high cost of function evaluations limits their widespread application. To

* Corresponding author.

E-mail address: fanzhun@uestc.edu.cn (Z. Fan).

¹ These authors contributed equally to this work.

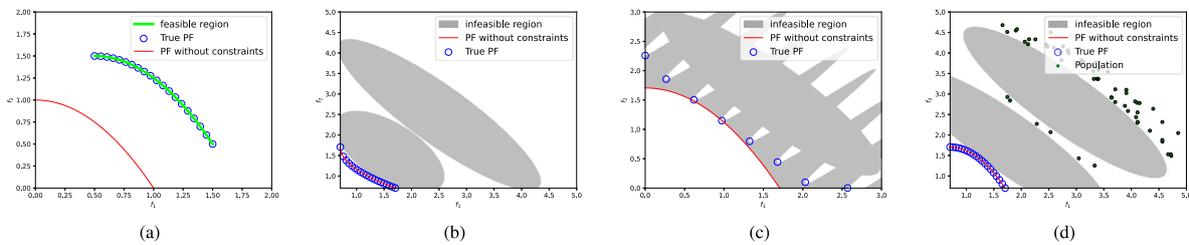


Fig. 1. Illustrations for the four main challenges of ECMOPs, which are: (a) feasibility difficulty, (b) convergence difficulty, (c) diversity difficulty, and (d) the challenge of searching for constrained Pareto solutions with limited evaluation budgets.

address this issue, surrogate models can be employed to approximate these expensive objective and constraint functions. By leveraging these models to guide the search process, satisfactory results can still be achieved with fewer function evaluations.

Recent significant advancements have been made in surrogate-assisted multi-objective evolutionary algorithms (SA-MOEA) for solving expensive multi-objective problems without constraints. These advancements primarily focus on three aspects: the construction, management, and design of infill criteria for surrogate models. Firstly, the establishment of surrogate models offers a cost-effective alternative to using more expensive original evaluation functions. Typical surrogate models include Gaussian processes [2,3], radial basis functions (RBF) [4,5], and support vector machines (SVM) [6]. Secondly, to enhance the predictive accuracy of surrogate models, effective management and periodic updates are necessary. This includes using surrogate model ensembles [7,8] and implementing model selection strategies [9]. Lastly, the purpose of the infill criterion is to select the most promising solutions for actual evaluation, which helps accelerate population convergence while maintaining diversity. Common metrics used in this process include expected improvement (EI) [10–12], lower confidence bound (LCB) [13], and probability of improvement (PoI) [14].

Constraint handling is also crucial for ECMOPs. Existing constraint handling methods are mainly categorized as follows [15]: (1) Methods based on the penalty function, which transform a constrained optimization problem into an unconstrained one by applying a penalty factor for optimization. (2) Methods based on the separation of objectives and constraints, such as the constrained dominance principle (CDP) [16], ϵ -CDP [17,18], and stochastic ranking [19]. (3) Multi-objective methods, for example, considering constraint violation (CV) as an additional optimization objective [20,21], or treating the constraint function as an objective function simultaneously [22]. (4) Methods that transform CMOPs into other types of problems, such as converting CMOPs into multi-stage optimization problems to be optimized sequentially [23–25], or transforming CMOPs into collaborative optimization of different subproblems within multi-subpopulations [26–28]. Although the CMOEAs that integrate these constraint handling techniques have made significant achievements in solving CMOPs, they typically require numerous evaluations and cannot be directly used to solve ECMOPs.

Resolving ECMOPs usually requires overcoming feasibility, convergence, and diversity challenges within a smaller number of function evaluations. Fig. 1 illustrates these difficulties. As shown in Fig. 1(a), when the feasible region is relatively small, it becomes challenging for the population to find a feasible solution, leading to feasibility difficulties. When encountering multiple infeasible regions, the population may fall into local optima, resulting in convergence difficulties, as illustrated in Fig. 1(b). Additionally, when there are multiple segmented feasible regions along the PF, the population may not fully cover all the solutions across the entire PF, leading to diversity challenges, as shown in Fig. 1(c). Furthermore, as depicted in Fig. 1(d), with fewer function evaluations, most existing CMOEAs struggle to search the true PF. To address these challenges, we propose a surrogate-assisted push and pull search algorithm (SA-PPS), which incorporates Bayesian active learning

to effectively solve ECMOPs. The main contributions are outlined as follows:

1. We propose an integrated adaptive infill criterion. During the push phase, candidate solutions are selected based on the HV contribution and uncertainty of the surrogate evaluated individuals under unconstrained conditions, which helps the population quickly converge to the UPF while maintaining diversity. In the pull phase, an adaptive selection strategy based on regional division is adopted. This strategy adaptively selects candidate solutions based on feasibility, convergence, and distribution infill criterion according to the status of the sub-regions. It guides the algorithm to perform more refined local search near the CPF, while simultaneously enhancing the population's feasibility, convergence, and diversity.
2. A batch training data selection strategy based on Bayesian active learning is introduced in the pull search stage. This strategy not only allows the surrogate model to focus on predicting areas of interest and thereby improving prediction accuracy but also accelerates model training by simplifying the datasets.
3. The proposed SA-PPS algorithm was compared with several state-of-the-art algorithms on the MW and LIRCMOP test problems, as well as a real-world optimization problem. The experimental results demonstrate its excellent performance in handling ECMOPs.

The remainder of this paper is organized as follows: Section 2 introduces related work on expensive constrained multi-objective evolutionary algorithms, along with brief descriptions of the push and pull search framework, the Gaussian process regression model, and Bayesian active learning for data selection. Section 3 provides a detailed explanation of the proposed SA-PPS algorithm. Section 4 presents the experimental results and analysis of SA-PPS and comparative algorithms on the MW and LIRCMOP test problems, as well as an aircraft design optimization problem. Finally, conclusions are drawn in Section 5.

2. Preliminary

2.1. Related work

Recently, ECMOPs have attracted considerable attention from researchers [29]. The primary method for addressing ECMOPs is surrogate-assisted CMOEAs (SA-CMOEAs). Similar to multi-objective evolutionary algorithms (MOEAs), SA-CMOEAs can typically be categorized into three types: (1) Indicator-based methods, (2) Dominance-based methods, and (3) Decomposition-based methods.

Indicator-based SA-CMOEAs primarily focus on designing appropriate selection indicators for candidate solutions that can filter out solutions enhancing the diversity and convergence of the population, while guiding the population towards the feasible region as much as possible. For instance, De Winter et al. [30] convert the surrogate model-predicted objective value into an HV indicator and use the HV contribution as the objective function to select appropriate candidate solutions. Singh et al. [31] employ the product of the PoI [14] and

the HV value as a measure of individual fitness for optimization. Deb et al. suggested using the S-metric indicator in the M5 framework [32], which combines the sum of the achievement aggregate function and the CV value to determine the fitness of a solution. Currently, indicator-based methods fail to effectively utilize the potential of infeasible solutions to help populations escape local optima. Although individuals within the infeasible region may have greater potential, they are easily dominated by individuals in the feasible region.

Decomposition-based SA-CMOEAs decompose ECMOPs into many sub-problems for optimization. For instance, Yang et al. [29] introduced the ASA-MOEA/D algorithm which includes three search strategies: feasibility search, diversity search, and convergence search. This algorithm adaptively selects one of these strategies based on the state of the population. Deb et al. [33] employ the S-metric indicator to assess the fitness of an individual for a subproblem and utilize a reference vector to preserve the diversity of the population. Han et al. [34] use the product of the objective value and the feasibility probability of the decomposed subproblem as the measure of individual fitness, aiming to minimize the objective value while considering feasibility. Decomposition-based SA-CMOEAs can maintain population diversity and prevent the population from converging on local optima. However, the selection of candidate solutions prioritizes feasibility, which makes it difficult for the population to cross infeasible regions and converge to the true constrained PF.

Dominance-based SA-CMOEAs utilize surrogate models to predict all objective and constraint values of individuals and calculate multiple indicators for select candidate solutions by non-dominated ranking. For instance, Blank et al. [32] constructed a surrogate model for each objective and constraint function, and employed the NSGA-II algorithm to optimize ECMOPs. Regis [35] employed the minimum distance between the surrogate evaluated individuals and the expensive evaluated population in both the objective and decision spaces as two selection indicators. This approach selects candidate solutions through non-dominated ranking and screens valid individuals using predicted constraint violation values. Dominance-based SA-CMOEAs can comprehensively consider multiple optimization criteria. However, similar to decomposition-based and indicator-based methods, feasible individuals always dominate infeasible ones, which may prevent them from traversing infeasible regions in ECMOPs with complex constraints.

Overall, many existing CMOEAs [29,34,35] prioritize feasibility over convergence. However, prioritizing feasibility can make it challenging for these algorithms to handle complex CMOPs [23,26]. Therefore, a flexible strategy that toggles between feasibility and convergence is necessary. For example, in the KTS algorithm [36], an adaptive switching search strategy is introduced, which alternates between ignoring constraints and considering constraints in surrogate evolutionary searches based on the population's state. In the MGSAEA algorithm [26], a multi-stage search framework is employed, initially ignoring constraints but later considering them, thus reducing time costs with a coarse-grained surrogate model. However, these algorithms do not consider how to select appropriate data samples for training the surrogate model, which may impact the accuracy of the surrogate model.

To address the limitations of current SA-CMOEAs in handling ECMOPs, we propose an SA-CMOEA algorithm named SA-PPS, based on the push and pull search framework. It integrates surrogate modeling and Bayesian active learning within the PPS framework. To effectively search the constrained PFs, we have developed distinct infill criteria for both the push and pull search stages. Utilizing these strategies, SA-PPS is capable of exploring the unconstrained PFs with limited function evaluations during the push stage and swiftly identifying regions near constrained PFs in the pull stage.

2.2. Push and pull search framework

The push and pull search (PPS) framework [23] represents a classical multi-stage optimization algorithm within the realm of CMOEAs. To solve CMOPs, PPS [23] divides the optimization process into two stages: the push search stage, where the population optimizes without considering constraints, and the pull search stage, where the population considers both optimization objectives and constraints. This framework can flexibly integrate various search mechanisms into the push and pull search stages, enabling it to handle CMOPs with different characteristics [24].

2.3. Gaussian regression process model

The Gaussian process regression model (GPR) is one of the most commonly used surrogate models. For given data, the GPR model considers the output y of input \mathbf{x} as a random variable following a Gaussian distribution $N(\mu, \sigma)$, where μ is the mean value of prediction and σ is the variance of prediction, also called uncertainty. Let the training dataset be $(\mathbf{x}_i, y), \mathbf{x}_i \in \mathbb{R}^D, i \in \{1, \dots, D\}$. The output of the GPR model for the prediction point \mathbf{x} is as follows:

$$\hat{y}(\mathbf{x}^*) = \mu(\mathbf{x}^*) + \mathbf{k}^{*T} (K + \sigma_n^2 I)^{-1} (\mathbf{y} - \mu(X)) \quad (3)$$

$$\sigma^2(\mathbf{x}^*) = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^{*T} (K + \sigma_n^2 I)^{-1} \mathbf{k}^* \quad (4)$$

where $\hat{y}(\mathbf{x}^*)$ is the prediction mean, $\sigma^2(\mathbf{x}^*)$ is the prediction variance, \mathbf{k}^* is the covariance vector between the training set X and the prediction point \mathbf{x}^* , K is the covariance matrix of the training X , and $\mu(X)$ is the mean vector of the training set X . The covariance function is also referred to as the kernel function in the GPR model. In this paper, the Matérn 5/2 kernel [37] is used as the kernel function of the GPR model, and it is defined as follows:

$$k(\mathbf{x}, \mathbf{x}') = \sigma_n^2 \left(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2} \right) \exp\left(-\frac{\sqrt{5}d}{\rho}\right) \quad (5)$$

where ρ is a non-negative hyperparameter of the kernel function. $d = \sqrt{(\mathbf{x} - \mathbf{x}')^T (\mathbf{x} - \mathbf{x}')}$ is the Euclidean distance between \mathbf{x} and \mathbf{x}' .

2.4. Bayesian active learning for data selection

Bayesian active learning is an effective machine learning strategy aimed at optimizing model performance by selecting samples with the highest informational value in a limited data environment. This method has been widely used in classification and regression tasks [38]. Compared to many traditional predictors, it not only has clear statistical or physical meaning [39], but also can balance the use of experiments to explore the unknown function with experiments that exploit prior knowledge to identify extrema [40].

The specific process of Bayesian active learning involves the following steps: (1) In the initial stage, a surrogate model is trained using a small amount of labeled data, and the model estimates the prediction uncertainty of unlabeled data using posterior distribution. (2) A query strategy (such as uncertainty or information entropy) is then used to select key samples that can significantly reduce model's uncertainty, and these samples are labeled. The newly labeled data are used to update the surrogate model. (3) This process is repeated until the predefined performance goal is achieved or the algorithm's termination condition is met.

In this paper, we combine Gaussian process regression models with Bayesian active learning. Gaussian processes can predict the mean and variance of sampled points, allowing the model to optimize prediction accuracy by selecting samples with the highest prediction uncertainty with minimal computational expense. This strategy is particularly important during the pull search stage for building local surrogate models, as it enables more precise selection of candidate solutions that contribute most to improving the model's performance.

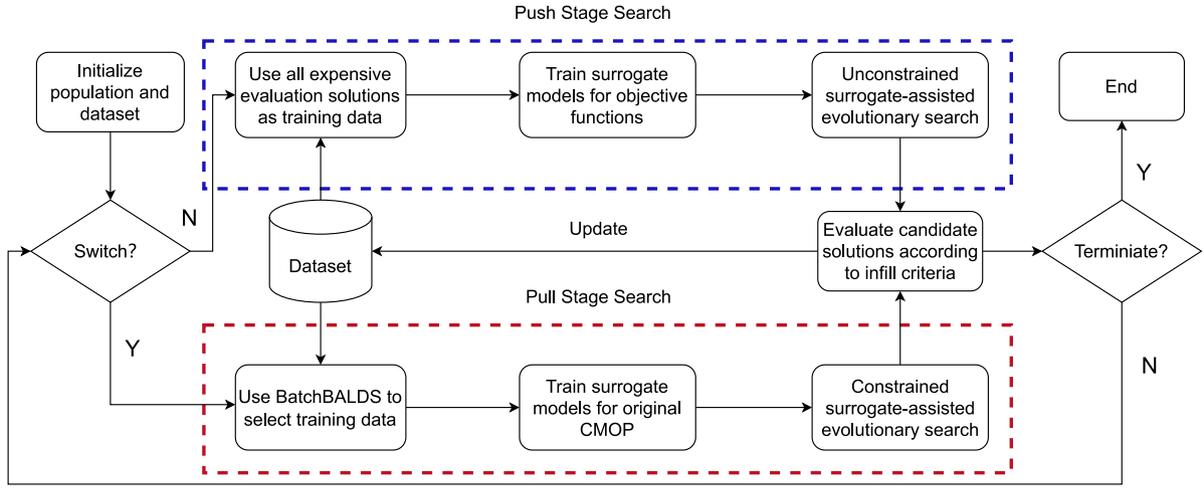


Fig. 2. Flowchart of the proposed SA-PPS.

3. Proposed algorithm

The proposed SA-PPS consists of three main components: a push search stage, a pull search stage, and a data selection mechanism based on Bayesian active learning. The functions of each component are briefly described as follows:

- 1. Push search stage:** In the push search stage, SA-PPS uses NSGA-II and surrogate models to guide the evolution of the surrogate evaluated population without considering any constraints. It uses hypervolume contribution and objective uncertainty to select candidate solutions for expensive evaluation, allowing the population to quickly cross the infeasible region and reach the UPF. Details can be found in Section 3.1.
- 2. Pull search stage:** During the pull search stage, surrogate models are constructed separately for each objective and constraint. By using reference vectors, the population is divided into multiple sub-regions. Different solution selection strategies are allocated to each sub-region based on their specific states, enabling the population to converge more effectively while maintaining diversity. Details can be found in Section 3.2.
- 3. Batch Bayesian active learning data selection (BatchBALDS):** During the pull search stage, as the amount of sample data increases, the training time for the surrogate model increases rapidly. The BatchBALDS method enables the selection of appropriate training data for model construction, which can significantly accelerate the training of the surrogate model and enable the model to focus on estimating the region around the CPF. This further enhances the performance of the proposed SA-PPS. Details can be found in Section 3.3.

The flowchart of the SA-PPS algorithm is shown in Fig. 2. First, the population is initialized, and an expensive evaluation is conducted. Then, the current search state is judged: if it is the push search stage, a surrogate model is established only for each objective, and candidate solutions are selected using the push stage search infill criterion. If it is in the pull search stage, the training data is selected based on BatchBALDS to establish a surrogate model for each objective and constraint, and the candidate solution is selected using the pull search stage infill criterion. The transition between the Push and Pull phases is determined by the HV value change rate γ_{HV} of the population. When γ_{HV} is less than the given threshold δ , the algorithm transitions from the push search stage to the pull search stage. The definition of γ_{HV} is as follows:

$$\gamma_{HV} = \frac{HV_k - HV_{k-l}}{\max(HV_{k-l}, \Delta)} \quad (6)$$

Algorithm 1: SA-PPS

Input: The max number of true evaluations FE_{max} , the number of selected candidates N_{sel} , the surrogate evaluated population size N_{surr} , the number of generations for surrogate assisted search t_{max}

Output: The archive of expensive evaluated solutions P_{arch}

```

1  $P_{arch} \leftarrow$  generate  $N$  individuals by Latin hypercube sampling;
2  $stage \leftarrow 0$ ;
3 while  $FE < FE_{max}$  do
4   if  $stage == 0$  then
5      $P_{cand} \leftarrow$  PushSearch( $P_{arch}, N_{sel}, N_{surr}, t_{max}$ );
6   else
7      $P_{cand} \leftarrow$  PullSearch( $P_{arch}, N_{sel}, N_{surr}, t_{max}$ );
8   end
9   Evaluate each solution in  $P_{cand}$  with a real function;
10   $P_{arch} \leftarrow P_{arch} \cup P_{cand}$ ;
11  update  $FE$ ;
12  if  $stage == 0$  and ( $\gamma_{HV} < \delta$  or  $FE > FE_{max}/2$ ) then
13     $stage \leftarrow 1$ ;
14  end
15 end

```

where HV_k is the HV value of the current k -generation population and HV_{k-l} is the HV value of the population from l generations earlier, and Δ is set to $1e^{-6}$. The rate of change in the HV indicator reflects the current population's convergence status. When the population cannot produce new non-dominated solutions during the evolutionary process, the HV change rate is 0. Conversely, if new non-dominated solutions are generated, the HV change rate is a value greater than 0, with larger values more strongly indicating that the population has not yet converged. When the HV change rate is extremely small, it suggests that the population has literally converged. At this point, switching to the pulling stage can effectively avoid wasting computational resources, thereby improving the search efficiency and performance of the SA-PPS algorithm. It is worth noting that if the expensive evaluation reaches 0.5 times the maximum evaluation and the algorithm has not yet transitioned to the pull search stage, the algorithm is forcibly transitioned to avoid wasting too many evaluations in the push search stage.

3.1. Push search stage

In the push stage, we build a surrogate model for each objective function, that is, there are M surrogate models in total. Additionally,

the NSGA-II is used as the optimizer to guide the search of the population without considering any constraints. When choosing candidate solutions for expensive evaluation, we combine hypervolume improvement (*HVI*) and uncertainty. The calculation of *HVI* is as follows:

$$HVI(\mathbf{x}_{\text{surr}}) = HV(\hat{f}(\mathbf{x}_{\text{surr}}) \cup y_e) - HV(y_e) \quad (7)$$

where \mathbf{x}_{surr} is a candidate solution, and y_e is a set of expensive evaluated solutions in the objective space. When the number of non-dominated solutions in y_e is less than the selected number N_{sel} , many individuals have an *HVI* value of zero. Then, we select solutions for expensive evaluation based on their average uncertainty. The definition of an individual's average uncertainty is as follows:

$$\hat{\sigma}_{\text{mean}}(\mathbf{x}) = \frac{1}{m} \sum_{i=1}^m \hat{\sigma}_i(\mathbf{x}) \quad (8)$$

where the uncertainty of the i th objective value, $\hat{\sigma}_i(\mathbf{x})$, of the individual \mathbf{x} is calculated using the trained GPR model.

The detailed steps of the push search are shown in Algorithm 2. Firstly, all the expensive evaluated individuals are used as training data to train the surrogate model for each objective function, as shown in line 1. Then, NSGA-II is used as the optimizer to guide the evolution of the surrogate evaluated population, as shown in line 2. After ω generations of evolution, the individual with the largest *HVI* in the surrogate evaluated population is selected to join the candidate solution set, as shown in lines 8–11. If the *HVI* of the individual with the largest *HVI* is 0, the individual with the largest $\hat{\sigma}_{\text{mean}}$ is selected to join the candidate solution set, as shown in lines 12–14.

Algorithm 2: PushSearch

Input: The archive of expensive evaluated solutions P_{arch} , the number of selected candidates N_{sel} , the surrogate evaluated population size N_{surr} , the number of generations for surrogate assisted search t_{max}

Output: Candidate solutions for expensive evaluation P_{arch}

```

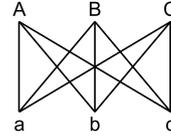
1  $S_{\text{models}} \leftarrow$  Train GPR models separately for each objective using
  the individuals in  $P_{\text{arch}}$ ;
2  $P_{\text{surr}} \leftarrow$  NSGA-II( $P_{\text{arch}}, S_{\text{models}}, t_{\text{max}}$ );
3  $P_{\text{cand}} = \emptyset$ ;
4 foreach  $\mathbf{x} \in P_{\text{surr}}$  do
5    $\hat{\sigma}_{\text{mean}}(\mathbf{x}) \leftarrow \frac{1}{m} \sum \hat{\sigma}_i(\mathbf{x})$ ;
6 end
7 for  $i \leftarrow 1 : N_{\text{sel}}$  do
8   foreach  $\mathbf{x} \in P_{\text{surr}}$  do
9      $HVI_{\mathbf{x}} \leftarrow HV(\mathbf{x} \cup P_{\text{cand}} \cup P_{\text{arch}}) - HV(P_{\text{cand}} \cup P_{\text{arch}})$ ;
10  end
11   $\mathbf{x} = \underset{\mathbf{x} \in P_{\text{surr}} \setminus P_{\text{cand}}}{\text{argmax}} HVI_{\mathbf{x}}$ ;
12  if  $HVI_{\mathbf{x}} = 0$  then
13     $\mathbf{x} \leftarrow \underset{\mathbf{x} \in P_{\text{surr}} \setminus P_{\text{cand}}}{\text{argmax}} \hat{\sigma}_{\text{mean}}(\mathbf{x})$ ;
14     $P_{\text{cand}} \leftarrow P_{\text{cand}} \cup \mathbf{x}$ ;
15  else
16     $P_{\text{cand}} \leftarrow P_{\text{cand}} \cup \mathbf{x}$ ;
17  end
18 end

```

3.2. Pull search stage

In the pull search stage, the algorithm aims to efficiently search for CPFs, for which an adaptive infill criterion based on subregion division is proposed. This method enables the population to effectively search for CPFs while maintaining diversity. The distributions of solutions within a subregion Δ_i for the expensive evaluated population and the surrogate evaluated population can be categorized into three scenarios:

Expensive evaluated individuals



- A, a: At least one feasible individual exists.
 B, b: Only infeasible individuals exist.
 C, c: No individuals exist.

Surrogate evaluated individuals

- S1: Aa, Ab \longrightarrow Convergence Infill Criterion
 S2: Ba, Bb \longrightarrow Feasibility Infill Criterion
 S3: Ac, Bc, Ca, Cb, Cc \longrightarrow Diversity Infill Criterion

Fig. 3. Illustrations for selecting candidate individuals based on three infill criteria during the pull search stage.

(1) At least one feasible solution exists, (2) Only infeasible solutions exist, and (3) No solutions exist.

Based on the different combinations of the three scenarios for each population, there are a total of nine different cases, as illustrated in Fig. 3. Here, A, B, and C represent the three scenarios for the expensive evaluated population, while a, b, and c represent the three scenarios for the surrogate evaluated population.

The specific correspondences between Figs. 3 and 4 are described as follows: Fig. 4 illustrates the process for selecting candidate individuals in the objective space. In Fig. 4, the objective space is divided into five sub-regions ($\Omega_1, \Omega_2, \Omega_3, \Omega_4$, and Ω_5), which correspond to five cases in Fig. 3, namely S3:Ac, S1:Aa, S3:Bc, S2:Bb, and S1:Ab. The specific selection processes of 9 corresponding cases are provided in Appendix I.

Specifically, in the cases of S3:Ac and S3:Bc, there are expensive evaluated individuals in sub-regions Ω_1 (or Ω_3), but no surrogate evaluated individuals. In these cases, the diversity infill criterion is employed. According to this criterion, the diversity values of the surrogate evaluated individuals in the neighborhoods are calculated, specifically the sum of the Chebyshev values and constraint violation values. The surrogate evaluated individual with the smallest diversity value is selected as the candidate solution. The candidate solutions selected in Ω_1 and Ω_3 are represented by diamonds with blue and yellow borders, respectively, in Fig. 4.

For the case of S1:Aa, there is at least one expensive evaluated individual and at least one surrogate evaluated individual within the feasible region in sub-region Ω_2 . In this case, we employ the convergence infill criterion to select a candidate solution. The surrogate evaluated individual with the highest *HVI* value from the feasible region is selected as the candidate solution, as indicated by the diamond with a blue border in Fig. 4. For the case of S1:Ab, there is at least one expensive evaluated individual within the feasible region, while the surrogate evaluated individuals are all in the infeasible region in sub-regions Ω_5 . In this case, we also use the convergence infill criterion to select a candidate solution. The surrogate evaluated individual with the smallest CV value from the infeasible region is chosen as the candidate solution, as indicated by the diamond with a purple border in Fig. 4.

In the case of S2:Bb, both expensive evaluated individuals and surrogate evaluated individuals are located in the infeasible region of sub-region Ω_4 . In this scenario, the feasibility infill criterion is employed. The surrogate evaluated individual with the maximum feasibility value is chosen as the candidate solution, as indicated by the diamond with a red border in Fig. 4.

The specific descriptions of the convergence, feasibility, and diversity infilling strategies are as follows:

- **Convergence Infill Criterion (CIC).** The CIC is designed to guide the population towards more optimal directions within the feasible domain. To ensure that individuals within subregions search

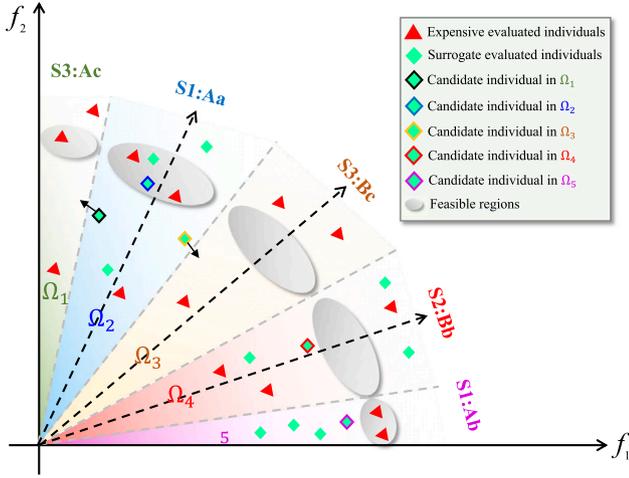


Fig. 4. Illustrations for selecting candidate individuals in the objective space.

while considering both distribution and convergence within the subregion, CIC integrates the Hypervolume Indicator (HVI) and the constraint dominance principle (CDP) to select candidate solutions, known as HVI-CDP. According to the HVI-CDP, solution \mathbf{x}_a is considered better than \mathbf{x}_b if it meets any of the following conditions:

1. \mathbf{x}_a and \mathbf{x}_b are both infeasible, and $CV(\mathbf{x}_a) < CV(\mathbf{x}_b)$.
2. \mathbf{x}_a is feasible, but \mathbf{x}_b is infeasible.
3. \mathbf{x}_a and \mathbf{x}_b are both feasible, and $HVI(\mathbf{x}_a) > HVI(\mathbf{x}_b)$.

- **Feasibility Infill Criterion (FIC).** The FIC prioritizes individuals with the highest feasibility as candidate solutions. To quantify the feasibility of surrogate evaluated individuals, we introduce the feasibility indicator I_{fea} . A smaller I_{fea} indicates greater feasibility of the surrogate evaluated individual. If \mathbf{x} is infeasible, I_{fea} is defined as follows:

$$I_{fea} = \sum_{i=1}^p \max(\hat{g}_i(\mathbf{x}), 0) + \sum_{j=1}^q \max(\hat{h}_j(\mathbf{x}) - \delta, 0) \quad (9)$$

If \mathbf{x} is feasible, I_{fea} is defined as follows:

$$I_{fea} = \sum_{i=1}^p \min(\hat{g}_i(\mathbf{x}), 0) + \sum_{j=1}^q \min(\hat{h}_j(\mathbf{x}) - \delta, 0) \quad (10)$$

It can be observed that when the surrogate evaluated individual \mathbf{x} is predicted to be an infeasible solution, the value of I_{fea} for \mathbf{x} equals the overall constraint violation. When the surrogate evaluated individual \mathbf{x} is predicted to be a feasible solution, the value of I_{fea} equals the sum of constraint function values. A smaller value of I_{fea} indicates that \mathbf{x} is deeper within the feasible domain, implying greater feasibility.

- **Diversity Infill Criterion (DIC).** The DIC aims to encourage the population to explore different subregions as extensively as possible to prevent convergence to local optima. When a subregion lacks expensive evaluated individuals or surrogate evaluated individuals, we design a corresponding subproblem for this subregion as part of the infill criterion. It is defined as follows:

$$f_{sub}(\mathbf{x}|\lambda) = g_{tch}(\mathbf{x}|\lambda) + CV(\mathbf{x}) \quad (11)$$

where $g_{tch}(\mathbf{x}|\lambda)$ represents the Chebyshev decomposition function.

The pseudo-code for the pull search stage is depicted in Algorithm 3. In lines 1–2, BatchBALDS is first used to select training data, and then surrogate models for objectives and constraints are trained separately. In line 3, the CCMO algorithm is used as the optimizer, utilizing

the trained surrogate models to replace the real function evaluation. In lines 5–6, each individual in the surrogate evaluated population P_{surr} and in the expensive evaluated archive set A is associated with the weight vector λ_i , respectively. In line 7, the objective space is divided into N_{sel} subregions, each denoted as Δ_i . The regional division strategy used in this paper aims to divide the objective space into N_{sel} subregions, where N_{sel} is the number of selected candidates. First, N_{sel} weight vectors are uniformly generated in the objective space. Next, for any point in the objective space, the distance from that point to all the weight vectors is calculated, and the closest weight vector is identified. This point is then assigned to the subregion associated with this nearest weight vector. Consequently, the objective space is divided into N_{sel} non-overlapping subregions. In this paper, we set the value of N_{sel} to 5. In lines 10–11, the surrogate evaluated population is divided into N_{sel} sub-populations, and the state of the solutions in the sub-region Δ_i is assessed. In lines 12–13, if the state of S_{Δ_i} is S1, a candidate solution is chosen according to the CIC criterion. In lines 14–15, if the state of S_{Δ_i} is S2, a candidate solution is chosen according to the FIC criterion. In lines 16–17, if the state of S_{Δ_i} is S3, a candidate solution is chosen according to the DIC criterion.

Algorithm 3: PullSearch

Input: The archive of expensive evaluated solutions P_{arch} , the number of selected candidates N_{sel} , the surrogate evaluated population size N_{surr} , the number of generations for surrogate assisted search t_{max}

Output: Candidate solutions for expensive evaluation P_{arch}

```

1  $D \leftarrow \text{BatchBALDS}(P_{arch}, N_{D1}, N_{D2});$ 
2  $S_{models} \leftarrow \text{Train GPR models separately for each objective and constraint using the individuals in } D;$ 
3  $P_{surr} \leftarrow \text{CCMO}(P_{arch}, S_{models}, t_{max});$ 
4 Set uniform reference vectors  $\{\lambda_1, \dots, \lambda_{N_{sel}}\};$ 
5 Assign each individual  $\mathbf{x} \in P_{surr}$  to its closest reference vector;
6 Assign each individual  $\mathbf{x} \in P_{arch}$  to its closest reference vector;
7 Divide the objective space into  $N_{sel}$  subregions  $\Delta_i$ , for  $i \in \{1, \dots, N_{sel}\};$ 
8  $P_{cand} = \emptyset;$ 
9 for  $i \leftarrow 1 : N_{sel}$  do
10  $P_{surr}^{\Delta_i} \leftarrow P_{surr}$  in  $\Delta_i;$ 
11 Determine the state  $S_{\Delta_i}$  of subregion  $\Delta_i;$ 
12 if  $S_{\Delta_i}$  is S1 then
13    $\mathbf{x} \leftarrow \text{HVICDP}(P_{surr}^{\Delta_i}, 1);$ 
14 else if  $S_{\Delta_i}$  is S2 then
15    $\mathbf{x} \leftarrow \max_{\mathbf{x} \in P_{surr}^{\Delta_i}} I_{fea}(\mathbf{x});$ 
16 else if  $S_{\Delta_i}$  is S3 then
17    $\mathbf{x} \leftarrow \min_{\mathbf{x} \in P_{surr}^{\Delta_i}} g_{tch}(\mathbf{x}|\lambda_i) + CV(\mathbf{x});$ 
18 end
19  $P_{cand} \leftarrow P_{cand} \cup \mathbf{x};$ 
20 end
```

3.3. Batch data selection based on Bayesian active learning

As the training data increases, the training time for the surrogate model also grows, potentially leading to significant computational costs. Meanwhile, training data far from the Pareto front might distract the model's attention, leading to lower prediction accuracy in areas near the CPF. To address this issue, we propose a batch Bayesian active learning method called BatchBALDS. This method is divided into two stages: the first stage involves selecting a set of data samples, $D1$, to improve the accuracy of the surrogate model near the CPF region. The second stage involves selecting another set of data samples, $D2$, aimed at reducing the global error of the surrogate model.

To construct $D1$, we select a sample set $D1$ from both the infeasible and feasible regions near the CPF. For the infeasible part, inspired by the BiCo [41] algorithm, we first treat the constraint violation as

an additional objective for non-dominated sorting. From the obtained set of non-dominated solutions, we select the infeasible solutions of the original problem as the population U , which is the population located on the infeasible side of the CPF. Then, non-dominated sorting is performed on population U while ignoring the constraints of the original optimization problem. In population U , individuals in the front non-dominated layers are farther from the CPF, while individuals in the back layers are closer to the CPF. Therefore, we filter population U starting from the last non-dominated layer in selecting individuals while using crowding distance to choose individuals within the same non-dominated layer. This process is essentially the reverse of the selection step in the NSGA-II algorithm. For the selection of data in the feasible region, the NSGA-II algorithm is utilized. To select data D_2 , a batch data selection strategy inspired by Bayesian active learning is employed. By choosing a set of data points that exhibit the highest uncertainty, we aim to minimize the prediction error of the surrogate model. We use a greedy approximation approach, which involves iteratively selecting individuals with the highest uncertainty and adding them to D_2 until a predetermined quantity is reached.

Algorithm 4: BatchBALDS

Input: The archive of expensive evaluated solutions P_{arch} , the number of selected candidates in the first stage N_{D_1} , the number of selected candidates in the second stage N_{D_2}

Output: The data for training surrogate models D_{train}

- 1 $U \leftarrow$ Select infeasible solutions from P_{arch} and construct a new solution set using CV as an additional objective;
- 2 $D_{1_1} \leftarrow$ Select $N_{D_1}/2$ individuals from U using the NSGA-II selection operator in reverse order;
- 3 $D_{1_2} \leftarrow$ Select $N_{D_1}/2$ individuals from P_{arch} using the NSGA-II selection operator;
- 4 $D_1 \leftarrow D_{1_1} \cup D_{1_2}$;
- 5 $D_{\text{pool}} \leftarrow P_{\text{arch}} \setminus D_1$;
- 6 $D_2 \leftarrow \emptyset$;
- 7 **for** $i \leftarrow 1 : m$ **do**
- 8 Use D_1 to train the GPR model for the i -th objective;
- 9 $D_{\text{sel}} \leftarrow \emptyset$;
- 10 **for** $j \leftarrow N_{D_2}$ **do**
- 11 **foreach** $x \in D_{\text{pool}} \setminus D_{\text{sel}}$ **do**
- 12 $S_x \leftarrow \sigma(x|D_{\text{sel}})$;
- 13 **end**
- 14 $x_j = \underset{x \in D_{\text{pool}} \setminus D_{\text{sel}}}{\text{argmax}} S_x$;
- 15 $D_{\text{sel}} \leftarrow D_{\text{sel}} \cup x_j$;
- 16 **end**
- 17 $D_2 \leftarrow D_2 \cup D_{\text{sel}}$;
- 18 **end**
- 19 **for** $i \leftarrow 1 : q$ **do**
- 20 Use D_1 to train the GPR model for the i -th constraint;
- 21 $D_{\text{sel}} \leftarrow$ Use the same method as shown in lines 10-16 to obtain the training data;
- 22 $D_2 \leftarrow D_2 \cup D_{\text{sel}}$;
- 23 **end**
- 24 Remove duplicate individuals from D_2 ;
- 25 $D_{\text{train}} \leftarrow D_1 \cup D_2$;

The specific process of the BatchBALDS method is depicted as Algorithm 4. Initially, we select individuals near the CPF as the training dataset D_1 , as shown in lines 1–4. Then, for each objective and constraint, we train the corresponding GPR models as shown in lines 8 and 20. Subsequently, we iteratively select individuals with the highest uncertainty to be added to D_2 in a greedy manner, as shown in lines 10–17 and 21–22. Finally, we remove duplicate individuals from D_2 and then merge D_1 and D_2 to serve as the training data for this round of GPR model training.

During the sample selection process, it is necessary to calculate the uncertainty $\sigma(x|D_2)$. According to the Gaussian process assumption,

the joint distribution between the test point x^* and the training set D_1 containing N_{D_1} individuals, as well as the dataset D_2 containing l individuals, follows a Gaussian distribution in the absence of noise:

$$\begin{bmatrix} y_{D_1 \cup D_2} \\ y^* \end{bmatrix} \sim N \left(\begin{bmatrix} \mu(X_{D_1 \cup D_2}) \\ \mu(x^*) \end{bmatrix}, \begin{bmatrix} K(X_{D_1 \cup D_2}, X_{D_1 \cup D_2}) & K(x^*, X_{D_1 \cup D_2})^T \\ K(x^*, X_{D_1 \cup D_2}) & K(x^*, x^*) \end{bmatrix} \right) \quad (12)$$

Here, $K(X_{D_1 \cup D_2}, X_{D_1 \cup D_2})$ represents the covariance matrix between the training set D_1 and D_2 , while $K(x^*, X_{D_1 \cup D_2})$ denotes the covariance vector between x^* and $X_{D_1 \cup D_2}$. Their definitions are as follows:

$$K(X_{D_1 \cup D_2}) = \begin{bmatrix} k(x_{D_1}^1, x_{D_1}^1) & \dots & k(x_{D_1}^1, x_{D_2}^l) \\ \vdots & \ddots & \vdots \\ k(x_{D_2}^l, x_{D_2}^l) & \dots & k(x_{D_2}^l, x_{D_2}^l) \end{bmatrix} \quad (13)$$

$$K(x^*, X_{D_1 \cup D_2}) = [k(x^*, x_{D_1}^1) \quad \dots \quad k(x^*, x_{D_2}^l)] \quad (14)$$

Since the conditional distribution of a Gaussian distribution is still a Gaussian distribution, the predictive variance corresponding to the test point x^* can be derived as follows [42]:

$$\sigma(x^*) = K(x^*, X_{D_1 \cup D_2})K(X_{D_1 \cup D_2}, X_{D_1 \cup D_2})^{-1} (y_{D_1 \cup D_2} - \mu(X_{D_1 \cup D_2})) \quad (15)$$

4. Experimental study

To evaluate the performance of the proposed SA-PPS, we selected 9 algorithms for comparison, including CCMO [26], PPS-MOEA/D [23], MultiObjectiveEGO [43], M1-2 [33], M2-2 [33], MGSAEA [44], KTS [36], PAC-MOO [45] and SA-CRVEA-AS [46]. Among these, CCMO and PPS-MOEA/D are CMOEAs that do not use surrogate models. MultiObjectiveEGO, M1-2, M2-2, MGSAEA, KTS, PAC-MOO and SA-CRVEA-AS are surrogate-assisted CMOEAs specifically designed for ECMOPs. We selected two test problem sets, MW [47] and LIRCOP [17], with the number of decision variables set to 10. The parameter settings for the SA-PPS are as follows:

1. The maximum number of expensive evaluation is set to 1000.
2. The surrogate evaluated population size is set to 300, and the number of generations for surrogate-assisted evolution is 100.
3. The number of individuals for the expensive evaluation in each generation is set to 5.
4. For BatchBALDS, the sizes of the training sets D_1 and D_2 , denoted as N_{D_1} and N_{D_2} , are both set to 200.

For the comparison algorithms, the population size, the number of expensive evaluation, and the number of initial individuals are kept consistent with those of the SA-PPS, while other parameters are set according to the specifications in their respective papers.

We adopt the IGD [48], HV [49], and the success rate (SR) of obtaining feasible solutions as evaluation indicators. Both the IGD and HV metrics measure the convergence and diversity of an algorithm, where a lower IGD value and a higher HV value indicates better performance. The SR represents the ratio of the number of times an algorithm successfully finds a feasible solution in 30 independent runs on a given test problem. Additionally, the significance level for the Wilcoxon rank-sum test with the significance level of 0.05 is employed. When the search for any feasible solution fails, the IGD value is set to a large number (in this paper, it is set to $1e20$) and the HV value is set to 0 for the Wilcoxon rank-sum test. The results of comparison experiments are displayed in following subsections.

It is worth noting that the experiments related to the effectiveness of the BatchBALDS mechanism and its parameter sensitivity analysis are discussed in Sections B and C of Appendix II. The experiments on the sensitivity analysis of the forcible transition factor are introduced in Section D of Appendix II. The experiments on the effectiveness of the proposed filling criteria are detailed in Section E of Appendix II.

Table 1

Wilcoxon Rank Sum significance test results (win/loss/tie) of SA-PPS and the other 9 ECMOEAs on MW1-MW14 and LIRC MOP1-LIRC MOP14.

Test problem set	Metrics	Algorithms									
		PPS-MOEA/D	CCMO	MultiObjectiveEGO	M1-2	M2-2	MGSAEA	KTS	PAC-MOO	SA-CRVEA-AS	SA-PPS
MW1-14	IGD	0/14/0	0/13/1	1/12/1	2/11/1	2/11/1	2/9/3	4/6/4	0/14/0	3/6/5	—
	HV	0/14/0	0/13/1	1/12/1	0/13/1	0/13/1	1/10/3	4/6/4	0/14/0	0/10/4	—
LIRC MOP1-14	IGD	0/14/0	0/14/0	0/14/0	0/14/0	1/12/1	3/11/0	6/6/2	0/14/0	1/5/8	—
	HV	0/14/0	0/14/0	0/14/0	0/14/0	0/13/1	3/11/0	6/6/2	0/14/0	1/10/3	—

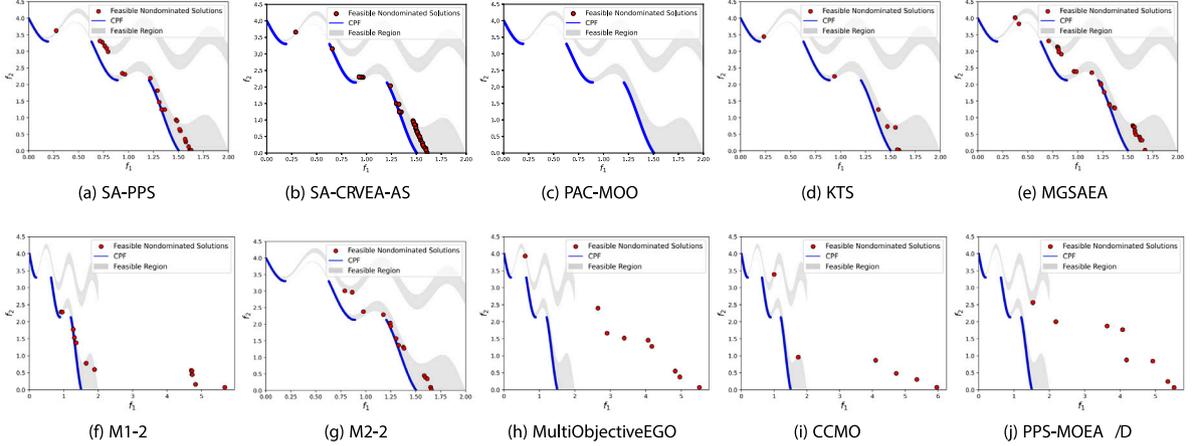


Fig. 5. SA-PPS, SA-CRVEA-AS, PAC-MOO, KTS, MGSAEA, M1-2, M2-2, MultiObjectiveEGO, CCMO and PPS-MOEA/D were independently run on the MW13 problem 30 times each, and the feasible non-dominated solutions with the median HV indicator were obtained.

4.1. Experimental results on the MW test problems

Table 1 displays the Wilcoxon Rank Sum significance test results of SA-PPS and the other 9 ECMOEAs on MW1-MW14. These results were obtained from 30 independent runs of the SA-PPS algorithm and 9 other comparison algorithms, including CCMO, PPS-MOEA/D, MultiObjectiveEGO, M1-2, M2-2, MGSAEA, KTS, PAC-MOO and SA-CRVEA-AS on the MW test problem set. From the IGD metric, it is clear that SA-PPS achieves the best performance on 6 test problems, followed by KTS, MGSAEA, MultiObjectiveEGO and SA-CRVEA-AS, which achieve the best performance on 4, 2, 1 and 1 problems, respectively. For the HV metric, SA-PPS significantly outperforms all comparison algorithms on 8 test problems, demonstrating that SA-PPS exhibits better diversity and convergence compared to existing SA-CMOEAs. The detailed results are presented in Supplementary Tables A.1 – A.2. Tables A.1 – A.2 also show the success rates (SR) of all algorithms in finding feasible solutions during 30 independent runs. SA-PPS achieves 100% SR on 12 problems in the MW test set, while the comparison algorithms CCMO, PPS-MOEA/D, MultiObjectiveEGO, M1-2, M2-2, MGSAEA, KTS, PAC-MOO and SA-CRVEA-AS achieve 100% SR on 6, 1, 8, 6, 6, 10, 9, 2 and 10 problems, respectively. This indicates that SA-PPS demonstrates a higher SR in finding feasible solutions compared to the 9 comparison algorithms.

Fig. 5 displays the feasible non-dominated solution sets with median HV values obtained by SA-PPS and 9 comparison algorithms after 30 independent runs on the MW13 test problem. The CPF of MW13 is located in three discontinuous and narrow feasible regions, posing a significant challenge to SA-CMOEAs. However, SA-PPS exhibits superior convergence and diversity compared to the other comparison algorithms. This advantage arises because, during the push stage, SA-PPS does not consider constraints while searching, thereby reducing the challenges posed by the narrow feasible regions. Additionally, the proposed solution selection mechanism employs a self-region partitioning approach, further enhancing the diversity of the population. On the other hand, although MGSAEA and KTS are able to search near the three PF regions, their convergence still needs improvement. M1-1 and M2-2 can approach one of the PFs, but their diversity remains inferior to

that of SA-PPS. MultiObjectiveEGO, CCMO, and PPS-MOEA/D, perform significantly worse in terms of diversity and convergence compared to SA-PPS. SA-CRVEA-AS obtains a similar performance to SA-PPS on MW13, while PAC-MOO fails to find any feasible solutions.

Fig. 6 illustrates the feasible non-dominated solution sets with median HV values obtained by SA-PPS and 9 comparison algorithms after 30 independent runs on MW14. The MW14 problem features multiple disconnected PF surfaces. SA-PPS demonstrates superior diversity compared to the 9 comparison algorithms, as it is able to locate all PF surfaces. In contrast, the comparison algorithms failed to search all PF surfaces. This further highlights the effectiveness of SA-PPS in handling complex ECMOPs with disconnected PF surfaces.

4.2. Experimental results on LIRC MOPs

We further compared the performance of SA-PPS with the 9 comparison algorithms on the LIRC MOP test problems. The LIRC MOP test suite includes some problems with very small feasible regions, requiring an algorithm of traversing multiple infeasible regions to reach the true CPF.

Table 1 displays the Wilcoxon Rank Sum significance test results of SA-PPS and the other 9 ECMOEAs on LIRC MOP1-LIRC MOP14. It is clear that SA-PPS significantly outperformed CCMO, PPS-MOEA/D, M1-2, MultiObjectiveEGO and PAC-MOO across all problems. Additionally, SA-PPS demonstrated significant superiority over M2-2 on 12 problems, over MGSAEA on 11 problems, over KTS on 6 problems and over SA-CRVEA-AS on 4 problems. The detailed results are presented in Supplementary Tables A.3 – A.4. In Tables A.3 – A.4, SA-PPS achieved 100% SR on LIRC MOP1-4, while the MGSAEA that also uses a two-phase search strategy did not. This discrepancy could be due to the significant distance between the CPFs and UPFs, necessitating more computational resources for the population to transition from the UPFs to the CPFs. However, the infill criterion proposed by SA-PPS accelerates this process, thereby enhancing the feasibility. Additionally, SA-PPS exhibited the best performance on LIRC MOP3-4, LIRC MOP7, and LIRC MOP12-14. KTS achieved the best performance on 6 problems, including LIRC MOP1-2, LIRC MOP5, and LIRC MOP9-11, because its

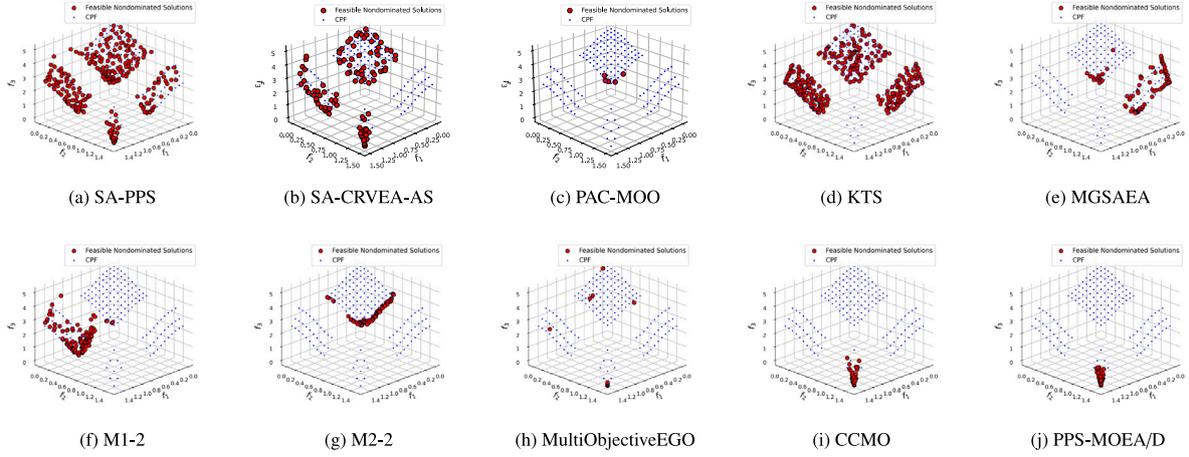


Fig. 6. SA-PPS, SA-CRVEA-AS, PAC-MOO, KTS, MGSAEA, M1-2, M2-2, MultiObjectiveEGO, CCMO and PPS-MOEA/D were independently run on the MW14 problem 30 times each, and the feasible non-dominated solutions with the median HV indicator were obtained.

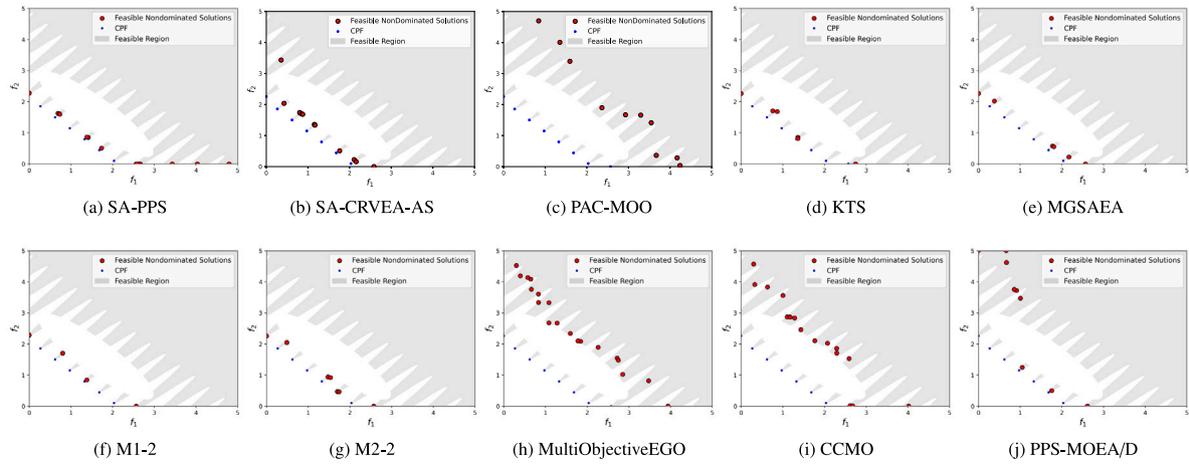


Fig. 7. SA-PPS, SA-CRVEA-AS, PAC-MOO, KTS, MGSAEA, M1-2, M2-2, MultiObjectiveEGO, CCMO and PPS-MOEA/D were independently run on the LIRCOP12 problem 30 times each, and the feasible non-dominated solutions with the median HV indicator were obtained.

adaptive search modes reduced the computational cost of searching the UPFs.

Figs. 7 and 8 display the feasible non-dominated solution sets with median HV values obtained by SA-PPS and 9 comparison algorithms after 30 independent runs on the LIRCOP12 and LIRCOP14 test problems, respectively. As evident from Fig. 7, SA-PPS significantly outperforms the comparison algorithms in terms of convergence. This performance is attributed to SA-PPS’s ability to mitigate the impact of infeasible regions during the push search stage. Additionally, the pull search stage utilizes an infill criterion that enhances the diversity of the population by selecting candidate solutions based on sub-region partitioning. This approach enables SA-PPS to search for feasible non-dominated solutions near the multiple disconnected CPFs in LIRCOP12.

Furthermore, as shown in Fig. 8, SA-PPS achieves better convergence and diversity on LIRCOP14 compared to the 9 comparison algorithms. This superiority can be attributed to the same factors discussed earlier. Overall, these results further validate the effectiveness of SA-PPS in handling complex ECMOPs with narrow feasible regions and disconnected PFs.

4.3. Aircraft design optimization

Namura et al. [50] introduced a set of practical problems related to the design optimization of aircraft wing shapes. In this study, we

selected one of these optimization problems, HPA222, as a real-world problem for our experiments. For a detailed description and mathematical expressions of this optimization problem, please refer to Appendix III.

In this experiment, the maximum number of evaluations is set to 2000. We employ 9 comparison algorithms: SA-CRVEA-AS, PAC-MOO, KTS, MGSAEA, M1-2, M2-2, MultiObjectiveEGO, CCMO and PPS-MOEA/D. The parameters for these algorithms are set to the default values specified in their respective papers. Since the true Pareto front for HPA222 is unknown, we use the HV metric to evaluate the performance of the algorithms, with the reference point set to [700, 2.5].

Table 2 presents the statistical results for the proposed SA-PPS algorithm and the 9 comparison algorithms applied to the HPA222 problem, with each algorithm being run independently 30 times. Additionally, Fig. 9 displays the non-dominated solution sets with median HV values for SA-PPS and the comparison algorithms. The results indicate that SA-PPS demonstrates superior performance, significantly outperforming all 9 comparison algorithms.

5. Conclusion

In this paper, we propose a surrogate-assisted CMOEA named SA-PPS to solve ECMOPs, which incorporates Bayesian active learning within the push and pull search framework. The proposed SA-PPS has three key components: the push search strategy, the pull search

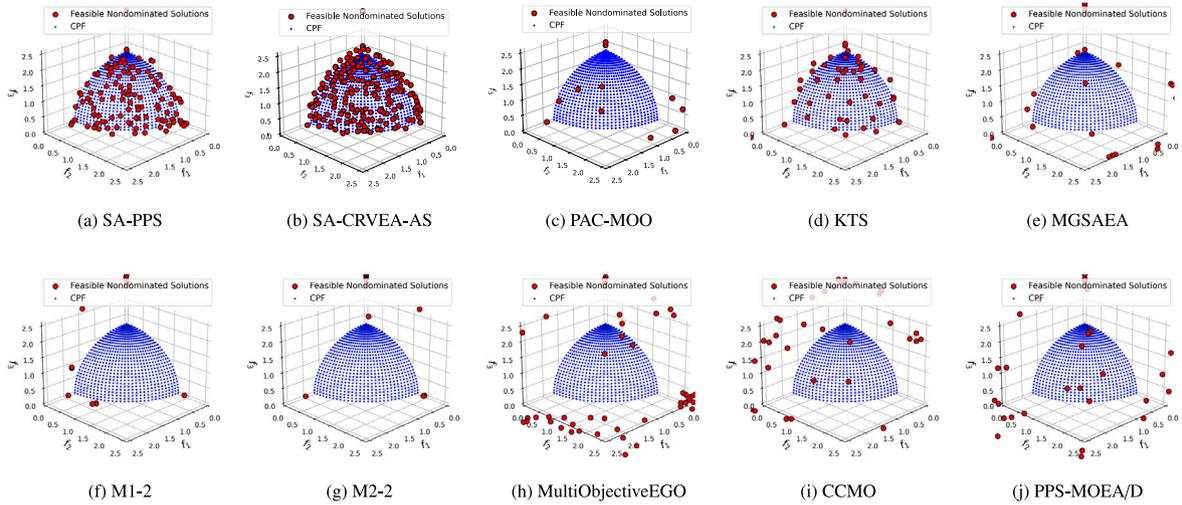


Fig. 8. SA-PPS, SA-CRVEA-AS, PAC-MOO, KTS, MGSAEA, M1-2, M2-2, MultiObjectiveEGO, CCMO and PPS-MOEA/D were independently run on the LIRC MOP14 problem 30 times each, and the feasible non-dominated solutions with the median HV indicator were obtained.

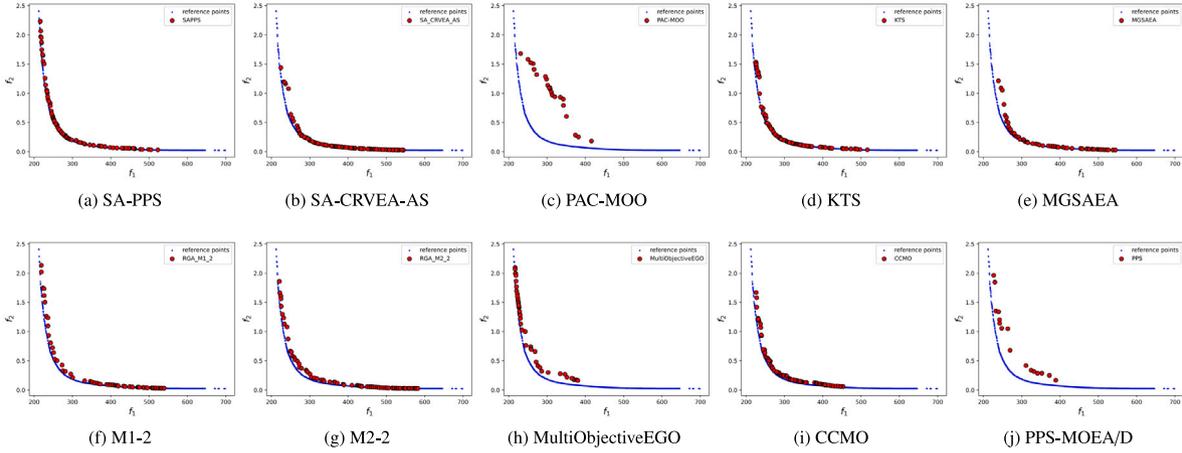


Fig. 9. Non-dominated solution sets corresponding to the median HV indicator obtained from 30 independent runs of SA-PPS and the comparison algorithms on HPA222.

Table 2

HV results of SA-PPS and the other 9 ECMOEs on HPA222. The symbols '+', '-' and '≈' indicate that the result is significantly better, significantly worse, or statistically similar to the results obtained by SA-PPS, respectively. The best result for each problem is highlighted in gray.

HPA222	CCMO	PPS-MOEA/D	MultiObjectiveEGO	M1-2	M2-2	MGSAEA	KTS	PAC-MOO	SA-CRVEA-AS	SPPS
Mean	1.342e+03 -	1.254e+03 -	1.305e+03 -	1.320e+03 -	1.336e+03 -	1.317e+03 -	1.359e+03 -	9.537e+02 -	1.353e+03 -	1.369e+03
Std	6.619e+00	3.441e+01	8.977e+00	5.873e+01	4.959e+01	3.035e+01	8.674e+00	7.737e+01	9.074e+00	4.401e+00

strategy, and a batch data selection strategy based on Bayesian active learning. At the push stage, the population is able to quickly approximate the UPFs within a limited number of evaluations without considering any constraints. At the pull stage, a new infill criterion is proposed to effectively guide the population towards the CPFs. Furthermore, a batch data selection strategy based on Bayesian active learning is suggested to enable the population to eliminate redundant data and select only valuable solutions for expensive evaluation as training data, thereby enhancing the accuracy of the surrogate model in estimating the region near the CPFs while maintaining overall surrogate accuracy. Comprehensive experimental results demonstrate that the proposed SA-PPS achieves the best performance on most test problems. When ECMOPs involve high-dimensional decision variables, it can lead to a significant decrease in the accuracy of the surrogate model. Therefore, a future research task is to enhance the capability of SA-PPS to handle ECMOPs with higher-dimensional decision variables.

CRedit authorship contribution statement

Wenji Li: Writing – review & editing. Ruitao Mai: Writing – original draft. Zhaojun Wang: Writing – review & editing. Yifeng Qiu: Writing – review & editing. Biao Xu: Writing – review & editing. Zhifeng Hao: Writing – review & editing. Zhun Fan: Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgments

This research was supported in part by the National Science and Technology Major Project, China (grant numbers 2021ZD0111501, 2021ZD0111502), the National Natural Science Foundation of China (grant number 62176147, 62476163), the Science and Technology Planning Project of Guangdong Province of China (grant numbers 2022A1515110660, 2021JC06X549), the Science and Technology Special Funds Project of Guangdong Province of China (grant numbers STKJ2021176, STKJ2021019), the Guangdong Basic and Applied Basic Research Foundation, China (2023B1515120020, 2024A1515012450), and the STU Scientific Research Foundation for Talents (grant numbers NTF21001, NTF22030).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.swevo.2024.101728>.

References

- [1] Z. Song, H. Wang, B. Xue, M. Zhang, Y. Jin, Balancing objective optimization and constraint satisfaction in expensive constrained evolutionary multi-objective optimization, *IEEE Trans. Evol. Comput.* (2023) 1–14.
- [2] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, K. Sindhya, A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization, *IEEE Trans. Evol. Comput.* 22 (1) (2018) 129–142.
- [3] K. Li, R. Chen, Batched data-driven evolutionary multiobjective optimization based on manifold interpolation, *IEEE Trans. Evol. Comput.* 27 (1) (2023) 126–140.
- [4] G. Sun, G. Li, Z. Gong, G. He, Q. Li, Radial basis functional model for multi-objective sheet metal forming optimization, *Eng. Optim.* 43 (12) (2011) 1351–1366.
- [5] T. Akhtar, C.A. Shoemaker, Multi objective optimization of computationally expensive multi-modal functions with RBF surrogates and multi-rule selection, *J. Global Optim.* 64 (1) (2016) 17–32.
- [6] I. Loshchilov, M. Schoenauer, M. Sebag, Dominance-based Pareto-surrogate for multi-objective optimization, in: K. Deb, A. Bhattacharya, N. Chakraborti, P. Chakraborty, S. Das, J. Dutta, S.K. Gupta, A. Jain, V. Aggarwal, J. Branke, S.J. Louis, K.C. Tan (Eds.), *Simulated Evolution and Learning*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 230–239.
- [7] Y. Zhao, J. Zhao, J. Zeng, Y. Tan, A two-stage infill strategy and surrogate-ensemble assisted expensive many-objective optimization, *Complex Intell. Syst.* 8 (6) (2022) 5047–5063.
- [8] Q. Lin, X. Wu, L. Ma, J. Li, M. Gong, C.A.C. Coello, An ensemble surrogate-based framework for expensive multiobjective evolutionary optimization, *IEEE Trans. Evol. Comput.* 26 (4) (2022) 631–645.
- [9] H. Wang, Y. Jin, C. Sun, J. Doherty, Offline data-driven evolutionary optimization using selective surrogate ensembles, *IEEE Trans. Evol. Comput.* 23 (2) (2019) 203–216.
- [10] J. Mockus, Application of Bayesian approach to numerical methods of global and stochastic optimization, *J. Global Optim.* 4 (4) (1994) 347–365.
- [11] K. Yang, M. Emmerich, A. Deutz, T. Bäck, Multi-objective Bayesian global optimization using expected hypervolume improvement gradient, *Swarm Evol. Comput.* 44 (2019) 945–956.
- [12] S. Qin, C. Sun, Q. Liu, Y. Jin, A performance indicator-based infill criterion for expensive multi-/many-objective optimization, *IEEE Trans. Evol. Comput.* 27 (4) (2023) 1085–1099.
- [13] B. Liu, Q. Zhang, G.G.E. Gielen, A Gaussian process surrogate model assisted evolutionary algorithm for medium scale expensive optimization problems, *IEEE Trans. Evol. Comput.* 18 (2) (2014) 180–192.
- [14] M. Emmerich, K. Giannakoglou, B. Naujoks, Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodelling, *IEEE Trans. Evol. Comput.* 10 (4) (2006) 421–439.
- [15] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, K.C. Tan, A survey on evolutionary constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* 27 (2) (2023) 201–221.
- [16] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197.
- [17] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, E. Goodman, An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions, *Soft Comput.* 23 (23) (2019) 12491–12510.
- [18] Q. Zhu, Q. Zhang, Q. Lin, A constrained multiobjective evolutionary algorithm with detect-and-escape strategy, *IEEE Trans. Evol. Comput.* 24 (5) (2020) 938–947.
- [19] T. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, *IEEE Trans. Evol. Comput.* 4 (3) (2000) 284–294.
- [20] C. Peng, H.-L. Liu, F. Gu, An evolutionary algorithm with directed weights for constrained multi-objective optimization, *Appl. Soft Comput.* 60 (2017) 613–622.
- [21] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, J. Zhang, Tri-goal evolution framework for constrained many-objective optimization, *IEEE Trans. Syst. Man Cybern.: Syst.* 50 (8) (2020) 3086–3099.
- [22] D. Vieira, R. Adriano, J. Vasconcelos, L. Krahenbuhl, Treating constraints as objectives in multiobjective optimization problems using niched Pareto genetic algorithm, *IEEE Trans. Magn.* 40 (2) (2004) 1188–1191.
- [23] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Push and pull search for solving constrained multi-objective optimization problems, *Swarm Evol. Comput.* 44 (2019) 665–679.
- [24] Z. Fan, Z. Wang, W. Li, Y. Yuan, Y. You, Z. Yang, F. Sun, J. Ruan, Push and pull search embedded in an M2M framework for solving constrained multi-objective optimization problems, *Swarm Evol. Comput.* 54 (2020) 100651.
- [25] Y. Tian, Y. Zhang, Y. Su, X. Zhang, K.C. Tan, Y. Jin, Balancing objective optimization and constraint satisfaction in constrained evolutionary multiobjective optimization, *IEEE Trans. Cybern.* 52 (9) (2022) 9559–9572.
- [26] Y. Tian, T. Zhang, J. Xiao, X. Zhang, Y. Jin, A coevolutionary framework for constrained multiobjective optimization problems, *IEEE Trans. Evol. Comput.* 25 (1) (2021) 102–116.
- [27] K. Li, R. Chen, G. Fu, X. Yao, Two-archive evolutionary algorithm for constrained multiobjective optimization, *IEEE Trans. Evol. Comput.* 23 (2) (2019) 303–315.
- [28] J. Wang, Y. Li, Q. Zhang, Z. Zhang, S. Gao, Cooperative multiobjective evolutionary algorithm with propulsive population for constrained multiobjective optimization, *IEEE Trans. Syst. Man Cybern.: Syst.* 52 (6) (2022) 3476–3491.
- [29] Z. Yang, H. Qiu, L. Gao, L. Chen, J. Liu, Surrogate-assisted MOEA/D for expensive constrained multi-objective optimization, *Inform. Sci.* 639 (2023) 119016.
- [30] R. de Winter, P. Bronkhorst, B. van Stein, T. Bäck, Constrained multi-objective optimization with a limited budget of function evaluations, *Memet. Comput.* 14 (2) (2022) 151–164.
- [31] P. Singh, M. Rossi, I. Couckuyt, D. Deschrijver, H. Rogier, T. Dhaene, Constrained multi-objective antenna design optimization using surrogates, *Int. J. Numer. Modell. Electron. Netw. Devices Fields* 30 (6) (2017) e2248.
- [32] J. Blank, K. Deb, Constrained bi-objective surrogate-assisted optimization of problems with heterogeneous evaluation times: Expensive objectives and inexpensive constraints, in: H. Ishibuchi, Q. Zhang, R. Cheng, K. Li, H. Li, H. Wang, A. Zhou (Eds.), *Evolutionary Multi-Criterion Optimization*, Springer International Publishing, Cham, 2021, pp. 257–269.
- [33] K. Deb, R. Hussein, P.C. Roy, G. Toscano-Pulido, A taxonomy for metamodelling frameworks for evolutionary multiobjective optimization, *IEEE Trans. Evol. Comput.* 23 (1) (2019) 104–116.
- [34] Z. Han, F. Liu, C. Xu, K. Zhang, Q. Zhang, Efficient multi-objective evolutionary algorithm for constrained global optimization of expensive functions, in: 2019 IEEE Congress on Evolutionary Computation, CEC, 2019, pp. 2026–2033.
- [35] R.G. Regis, Multi-objective constrained black-box optimization using radial basis function surrogates, *J. Comput. Sci.* 16 (2016) 140–155.
- [36] Z. Song, H. Wang, B. Xue, M. Zhang, Y. Jin, Balancing objective optimization and constraint satisfaction in expensive constrained evolutionary multi-objective optimization, *IEEE Trans. Evol. Comput.* (2023).
- [37] M.G. Genton, Classes of kernels for machine learning: a statistics perspective, *J. Mach. Learn. Res.* 2 (2002) 299–312.
- [38] Y. Gal, R. Islam, Z. Ghahramani, Deep Bayesian active learning with image data, in: D. Precup, Y.W. Teh (Eds.), *Proceedings of the 34th International Conference on Machine Learning*, in: *Proceedings of Machine Learning Research*, vol. 70, PMLR, 2017, pp. 1183–1192.
- [39] P. Liu, L. Wang, R. Ranjan, G. He, L. Zhao, A survey on active deep learning: From model driven to data driven, *ACM Comput. Surv.* 54 (10s) (2022) <http://dx.doi.org/10.1145/3510414>.
- [40] A.G. Kusne, H. Yu, C. Wu, H. Zhang, J. Hattrick-Simpers, B. DeCost, S. Sarker, C. Oses, C. Toher, S. Curtarolo, et al., On-the-fly closed-loop materials discovery via Bayesian active learning, *Nat. Commun.* 11 (1) (2020) 5966.
- [41] Z.-Z. Liu, B.-C. Wang, K. Tang, Handling constrained multiobjective optimization problems via bidirectional coevolution, *IEEE Trans. Cybern.* 52 (10) (2022) 10163–10176, <http://dx.doi.org/10.1109/TCYB.2021.3056176>.
- [42] C.K. Williams, C.E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2, MIT press Cambridge, MA, 2006.
- [43] R. Hussein, K. Deb, A generative kriging surrogate model for constrained and unconstrained multi-objective optimization, in: *Proceedings of the Genetic and Evolutionary Computation Conference 2016, GECCO '16*, Association for Computing Machinery, New York, NY, USA, 2016, pp. 573–580.
- [44] Y. Zhang, H. Jiang, Y. Tian, H. Ma, X. Zhang, Multigranularity surrogate modeling for evolutionary multiobjective optimization with expensive constraints, *IEEE Trans. Neural Netw. Learn. Syst.* 35 (3) (2024) 2956–2968.
- [45] A. Ahmadianshalchi, S. Belakaria, J.R. Doppa, Preference-aware constrained multi-objective Bayesian optimization, in: *Proceedings of the 7th Joint International Conference on Data Science & Management of Data (11th ACM IKDD CODS and 29th COMAD)*, CODS-COMAD '24, Association for Computing Machinery, New York, NY, USA, 2024, pp. 182–191, <http://dx.doi.org/10.1145/3632410.3632427>.

- [46] H. Wu, Q. Chen, Y. Jin, J. Ding, T. Chai, A surrogate-assisted expensive constrained multi-objective optimization algorithm based on adaptive switching of acquisition functions, *IEEE Trans. Emerg. Top. Comput. Intell.* 8 (2) (2024) 2050–2064, <http://dx.doi.org/10.1109/TETCI.2024.3359517>.
- [47] Z. Ma, Y. Wang, Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons, *IEEE Trans. Evol. Comput.* 23 (6) (2019) 972–986.
- [48] P. Bosman, D. Thierens, The balance between proximity and diversity in multiobjective evolutionary algorithms, *IEEE Trans. Evol. Comput.* 7 (2) (2003) 174–188.
- [49] K. Deb, M. Abouhawwash, H. Seada, A computationally fast convergence measure and implementation for single-, multiple-, and many-objective optimization, *IEEE Trans. Emerg. Top. Comput. Intell.* 1 (4) (2017) 280–293.
- [50] N. Namura, Single and multi-objective benchmark problems focusing on human-powered aircraft design, 2023, arXiv preprint [arXiv:2312.08953](https://arxiv.org/abs/2312.08953).