This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

A Multi-Stage Expensive Constrained Multi-Objective Optimization Algorithm Based on Ensemble Infill Criterion

Haofeng Wu, Qingda Chen, *Member, IEEE,* Jiaxin Chen, Yaochu Jin, *Fellow, IEEE,* Jinliang Ding, *Senior Member, IEEE*, Xingyi Zhang, *Senior Member, IEEE*, and Tianyou Chai, *Life Fellow, IEEE*

***Abstract*—Surrogate-assisted evolutionary algorithms (SAEAs) rely on the infill criterion to select candidate solutions for expensive evaluations. However, in the context of expensive constrained multi-objective optimization problems (ECMOPs) with complex feasible regions, guiding the optimization algorithm towards the constrained Pareto optimal front and achieving a balance between feasibility, convergence, diversity, exploration, and exploitation using a single infill criterion pose significant challenges. We propose an ensemble infill criterion-based multi-stage SAEA (EIC-MSSAEA) to tackle these challenges. Specifically, EIC-MSSAEA comprises three stages. In the first stage, we ignore constraints to facilitate the rapid traversal of infeasible obstacles. In the second stage, only one constraint is activated at a time to increase algorithm diversity. Finally, in the last stage, we activate all constraints to improve overall feasibility. In each stage, EIC-MSSAEA first employs NSGA-III as the underlying baseline solver to explore the search space, in which promising solutions are then selected by an ensemble infill criterion that incorporates multiple base-infill criteria to measure the feasibility, convergence, diversity, and uncertainty of candidate solutions. Experimental results demonstrate the competitiveness of EIC-MSSAEA against state-of-the-art SAEAs for ECMOPs.**

***Index Terms*—Constrained multi-objective optimization, evolutionary algorithm, expensive evaluations, ensemble infill criterion, multi-stage SAEA**

I. Introduction

EXPENSIVE constrained multi-objective optimization problems (ECMOPs) are prevalent in industrial and engineering domains, such as mechanical design [1] and chemical engineering optimization [2]. Evaluating these problems demands considerable computation time or significant expenses [3]. Additionally, ECMOPs involve multiple conflicting objectives and numerous constraints, often treated as black boxes, making acquiring gradient information challenging. In contrast to single-objective optimization, multi-objective optimization does not yield a globally optimal solution achieved by simultaneously optimizing all conflicting objectives. Instead, the primary goal of multi-objective optimization is to identify a set of compromise solutions. Thus, the main aim of ECMOPs is to discover a collection of compromise solutions that satisfy the constraint conditions within a limited number of function evaluations (FEs) [4]. This work primarily focuses on ECMOPs characterized by inequality constraints. Without the loss of generality, an ECMOP can be represented in the following form [5]:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\Omega}\ \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), ..., f_m(\boldsymbol{x})) \\ &s.t.\ \ g_j(\boldsymbol{x}) \le 0, j \in \{1, ..., q\}, \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ represents a decision vector (i.e., a solution) of dimension D, belonging to the decision space $\Omega = \prod_{i=1}^{D}[x_i^l, x_i^u]$. $\boldsymbol{F}(\boldsymbol{x})$ represents the objective vector of dimension m. The function $g_j(\boldsymbol{x})$ denotes the jth constraint function, and q is the number of constraint functions. The overall constraint violation degree of a solution $\boldsymbol{x}$ is defined as follows:

$$C(\boldsymbol{x}) = \sum_{j=1}^{q} \max(0, g_j(\boldsymbol{x})). \tag{2}$$

If $C(\boldsymbol{x}) = 0$, the solution $\boldsymbol{x}$ is considered feasible. Otherwise, it is considered infeasible. In ECMOPs, a solution $\boldsymbol{x}^*$ is deemed feasible Pareto optimal if it satisfies $C(\boldsymbol{x}^*) = 0$ and is not dominated by any other solution $\boldsymbol{x}$. The set of all feasible Pareto optimal solutions in the decision space is called a constrained Pareto optimal set (CPS). Meanwhile, as shown by the red curve in Fig. 1, its corresponding objective values constitute the constrained Pareto optimal front (CPF). The set of Pareto optimal solutions without considering whether the constraints are satisfied is called unconstrained Pareto optimal set (UPS). In the objective space, it is called unconstrained Pareto optimal front (UPF), as denoted by the blue curve in Fig. 1.

Manuscript received –. This work was supported in part by the National Key R&D Plan Project under Grant 2022YFB3304700, in part by the National Natural Science Foundation of China under Grant 61988101 and 62203101, in part by the 111 Project 2.0 under Grant B08015, in part by the Xinliao Talent Program of Liaoning Province under Grant XLYC2202002, and in part by the China Postdoctoral Science Foundation under Grant 2023T160086. *(Corresponding authors: Yaochu Jin and Qingda Chen)*

Haofeng Wu, Qingda Chen, Jiaxin Chen, Jinliang Ding, Yaochu Jin, and Tianyou Chai are with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China. Email: 798291659@qq.com; cqd0309@126.com; chenjiaxin1934@gmail.com; jlding@mail.neu.edu.cn; tychai@mail.neu.edu.cn.

Yaochu Jin is also with the School of Engineering, Westlake University, Hangzhou 310030, China. Email: jinyaochu@westlake.edu.cn.

Xingyi Zhang is with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230601, China. Email: xyzhanghust@gmail.com.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

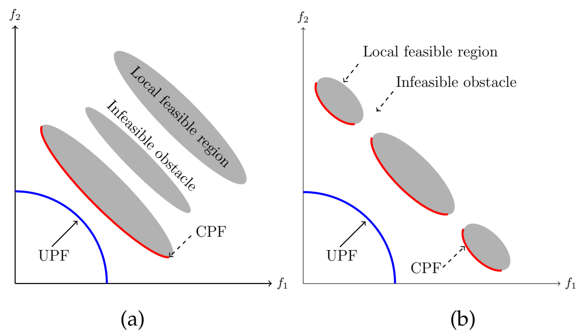


Fig. 1. Illustrative examples of infeasible obstacles in ECMOPs.

One of the difficulties encountered in ECMOPs is that the feasible region is discontinuous, and the CPF is split into multiple fragments. We illustrate these cases in Fig. 1(a), which shows that significant infeasible obstacles separate locally feasible regions. With limited FEs, the search can only reach in a locally feasible region far from the CPF, and traversing across these infeasible barriers and approaching the CPF become less likely. In Fig. 1(b), the CPF is divided into multiple segments by infeasible regions, making it challenging for the algorithm to search across the infeasible obstacles and explore more feasible regions. Secondly, the feasible areas of ECMOPs encountered in real-world scenarios are sometimes narrow. Consequently, the initial solutions may be distant from the CPS, making it extremely difficult to locate the vicinity of the CPS with a limited number of FEs.

Constrained multi-objective evolutionary algorithms (CMOEAs) have achieved remarkable success in addressing complex constrained multi-objective problems (CMOPs). It exhibits population-based characteristics and can approximate a set of non-dominated solutions that satisfy constraints within a single run, thereby providing a valuable approach for solving CMOPs [6]. CMOEAs can be categorized into three main types [7]: 1) Methods of using a constraint handling technology (CHT), exemplified by nondominated sorting-based constraint domination principle (CDP), NSGA-II-CDP [8]. 2) Multi-stage-based approaches, represented by C3M [9]. 3) Multiple groups-based methods, exemplified by MCCMO [7].

Population-based methods possess evolutionary characteristics, so CMOEAs require many FEs to obtain satisfactory solutions. However, this requirement conflicts with the goals of ECMOPs, which aim to minimize the number of FEs utilized [10]. Therefore, surrogate-assisted evolutionary algorithms (SAEAs), also known as data-driven evolutionary optimization techniques, have been proposed to address this issue [11], [12]. SAEAs consist of two interconnected design components [13], [14].

1) The first component involves surrogate modeling for expensive functions and enables faster evaluations, resulting in significant computational savings. Gaussian processes (GPs) [15], radial basis functions (RBFs) [16], [17], and neural networks [18] are commonly used for this purpose. Among these options, GPs are widely employed. One key advantage of GPs is their ability to provide point and uncertainty estimations [19]. Moreover, GP-assisted optimization is often used in the Bayesian optimization (BO) [20], [21] and efficient global optimization (EGO) [22].
2) The second component is meant for model management, which means selecting promising solutions from the surrogate-assisted search process for expensive FEs [23]. The quality of these solutions is measured using an acquisition function or infill criterion. They quantify the balance between exploration and exploitation. In ECMOPs, a good balance should also be achieved between feasibility, convergence, diversity, exploration and exploitation. A promising solution can be obtained either by directly optimizing the infill criterion or by first optimizing the surrogate approximation objective function and filtering it using the infill criterion. By carefully selecting solutions based on the infill criterion, the algorithm can guide the search towards optimal regions of the search space.

Achieving a balance between feasibility, convergence, diversity, exploration, and exploitation in ECMOPs is complex, and relying on a single infill criterion makes it even more challenging. Several reasons contribute to this difficulty: Firstly, striking a balance between convergence and diversity poses a challenge. The performance requirements of the algorithm vary at different search stages, as depicted in Fig. 1(b). In the presence of a discontinuous Pareto front, once convergence near the Pareto front is achieved, the focus needs to shift towards promoting diversity to improve overall performance.

Secondly, balancing between exploration and exploitation in the context of multiple conflicting objectives is challenging. For example, the accuracy of the surrogate model in approximating the landscape of each objective function may vary. The need for different levels of exploration and exploitation for various objective functions reduces the efficiency of finding optimal solutions when one only considers the balance between convergence and diversity.

Lastly, balancing the priority between constraints and objectives is a challenging aspect. In the case of ECMOPs with multiple infeasible obstacles, as shown in Fig. 1(a), prioritizing objectives more than constraints is necessary. This prioritization ensures that the search process can navigate infeasible regions effectively while guiding the search toward UPF. Then, pull the search back to CPF. It is worth noting that the effectiveness of the ensemble learning method in enhancing the performance of surrogate models has been demonstrated in [24], [25]. Motivated by these findings, we propose a novel infill criterion that combines multiple independent infill criteria to improve the selection of candidate solutions and enhance overall quality. By integrating multiple infill criteria, we aim to better balance convergence and diver-

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

sity, exploration and exploitation between objectives and constraints.

The effectiveness of the infill criterion in selecting potential solutions from the candidate set heavily relies on the ability of surrogate-assisted search to provide a high-quality set of candidate solutions. However, in the context of ECMOPs, separated feasible regions pose significant challenges for the surrogate-assisted search process. These challenges arise due to the complex nature of constraint functions and the combination of multiple constraints, which give rise to intricate constraint landscapes. In the field of evolutionary algorithms, some multi-stage CMOEAs have been developed to address these difficulties by gradually increasing the consideration of constraints and leveraging infeasible solutions to improve search performance [9], [26]. However, these approaches require many FEs to obtain satisfactory solutions. To tackle these challenges, we extend the concepts of multi-stage CMOEAs to address ECMOPs. The contributions of this article can be summarized with the following three main aspects.

1) An ensemble infill criterion (EIC) is proposed for selecting potential solutions at each stage. EIC includes six criteria. The first one, grounded in CDP and angle, gauges feasibility, convergence, and diversity. The second criterion is based on CDP and penalty-based boundary intersection (PBI) to assess feasibility, convergence, and diversity. The third one is based on the probability of feasibility (PoF) [27] to indicate the expected proximity and diversity improvement (EPDI), promoting exploration and exploitation. The remaining three criteria correspond to the above three ones without considering the constraints.
2) A new multi-stage SAEA, EIC-MSSAEA, is proposed for tackling complex ECMOPs based on the proposed EIC in each stage. The initial stage optimizes the objective function, aiding navigation through infeasible regions and achieving swift convergence toward the CPF vicinity. The second stage introduces a single constraint to enhance solution diversity. The final stage deploys all constraints to enhance convergence and diversity within the optimal feasible region. EIC-MSSAEA effectively allocates computational load across these three stages.
3) The proposed EIC-MSSAEA is rigorously tested on various constraint benchmark problems to demonstrate its competitive performance. Furthermore, we apply EIC-MSSAEA to solve the operational parameter optimization problem of a crude distillation unit and compare its performance against other state-of-the-art algorithms, showcasing its effectiveness. We highlight the flexibility of EIC-MSSAEA, with its first stage, called EIC-S1, specifically tailored for addressing expensive multi-objective optimization problems (EMOPs). To underscore the effectiveness of EIC-S1, we evaluate its performance on two unconstrained benchmark problems.

The paper is structured as follows. Section II gives an overview of the related work and introduces the preliminaries of this work. Section III provides a detailed explanation of the proposed EIC-MSSAEA. Section IV presents the experimental results with discussion. Finally, Section V concludes the paper and provides future work.

II. Related Work and Preliminaries

A. Different Types of Surrogate-Assisted Evolutionary Algorithms

According to the model management strategy, SAEAs can be divided into three categories.

The first category involves obtaining potential solutions by optimizing the infill criterion. This type of model management prioritizes the impact of the infill criterion over the optimizer. Due to space limit, a detailed introduction to the general framework of this type of SAEAs is given in Section I of the Supplementary material. In [28], a selection function is proposed to transform a CMOP into an unconstrained single-objective problem, and the expected improvement (EI) is maximized to find a new solution. The infill criterion for single-objective constrained optimization aims to balance the feasibility, exploration, and exploitation in the search process without explicitly considering the convergence and diversity of the solution set.

To alleviate the problem of insufficient diversity improvement, in [29], EI is extended to a multi-objective form based on the proposed EI matrix (EIM) to enhance the ability of EI to balance convergence and diversity. When there is no feasible solution, the PoF is maximized until a feasible solution is found and switched to the constrained EIM, which is EIM multiplied by PoF. Compared to EI, hypervolume (HV) can better measure the multi-objective performance of a solution. A HV infill criterion based on multi-objective probability of improvement (PoI), HV-PoI, is proposed in [30], in which HV can balance convergence and diversity, and PoI is used to measure the multi-objective probability of improvement. In [31], HV contribution is used to select solutions with improved convergence and diversity, in addition to the principle of prioritizing feasible solutions to handle constraints. Finally, to balance exploration and exploitation, lower confidence bound [32] replaces the objective value to calculate the HV contribution. In [33], [34], the constraints infill criteria based on information entropy are proposed.

All the above infill criteria measure convergence, diversity, exploration and exploitation, and also feasibility information by a scalar, and are directly optimized by a single-objective optimizer to obtain potential solutions. In [35], an infill criterion based on multi-objective performance measures is proposed, which is obtained by

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

CMOEA to obtain the set of potential solutions. The first class of methods needs more diversity considerations because they almost directly optimize an infill criterion as a single objective.

In the second category of SAEAs, a multi-objective optimization problem (MOP) or a CMOP approximated by surrogate models is first optimized, and then the potential solutions are selected by an infill criterion. In [36], an MOP based on the uncertainty of the surrogate model is constructed and optimized by NSGA-III [37]. Then, the expected HV improvement (EHVI) is used as the infill criterion to evaluate the final population. In addition, the importance sampling method is proposed to improve the computational efficiency of EHVI. In [38], RVEA [39] is used to optimize the MOP approximated by GPs to obtain the final population. Then, an adaptive infill criterion is proposed to filter out potential solutions. This criterion can dynamically adjust the weights of exploration, convergence, and diversity at different search stages.

These SAEAs only use one infill criterion to balance all performance, while in [40], three infill criteria for convergence, diversity, and model uncertainty are proposed separately. Then, based on the search status of the two-archive MOEA, the most suitable infill criterion is selected (KTA2). In [41], MOEA/D [42] is used as a search engine to optimize the MOP approximated by RBFs. To balance feasibility, convergence, and diversity performance, an adaptive search procedure is designed based on the state of the subproblem, and then the solution obtained by each search procedure is selected by a specific infill criterion. In [43], a strategy is proposed to adaptively switch the optimization process. KTA2 is used during the search process that ignores the constrained surrogate model. In the search process of building the constrained surrogate model, the co-evolutionary algorithm is used to optimize the approximated CMOP, and an improved infill criterion considering feasibility, diversity, and convergence is proposed to select the best solution from the final population. In [44], different granularity surrogate models are used to approximate the constraint function at different optimization stages to reduce the impact of the error introduced by the constraint surrogate. SPEA2 [45] is used as the optimizer, and then the environmental selection of SPEA2 is employed as the infill criterion to screen better solutions from the final population. In summary, the second category needs a balance between exploration and exploitation in the constrained space.

The third class of model management strategies is transformation of the original problem. In [46]–[48], the constraints of the original problem are transformed into additional into additional objectives. These approaches allow for constraint handling without explicit CHT, simplifying the optimization process. Moreover, treating constraints as objectives enables trade-offs between objectives and constraints. However, they may experience decreased convergence due to the additional objectives. In [49], a surrogate-assisted local search strategy is proposed by transforming the MOP into a single-objective optimization problem to search for local optimal solutions. However, this strategy also suffers from a lack of diversity.

B. Gaussian Process Model and Infill Criterion

For the given n observed pairs: $\boldsymbol{X} = (\boldsymbol{x}^1, ..., \boldsymbol{x}^n)$, $\boldsymbol{Y} = (\boldsymbol{y}^1, ..., \boldsymbol{y}^n)$, a GP can approximate an unknown function $f(\boldsymbol{x})$, $\boldsymbol{x}^i$ is a D-dimensional vector. At any point $\boldsymbol{x}$, GP treats $f(\boldsymbol{x})$ as a Gaussian random variable. Therefore, GP predicts a new input point $\boldsymbol{z}^*$ and estimates the associated uncertainty as follows:

$$\hat{f}(\boldsymbol{z}^*) = \hat{\mu} + \boldsymbol{c}^T \boldsymbol{C}^{-1}(Y - \mathbf{1}\hat{\mu}), \tag{3}$$

$$\delta^2(\boldsymbol{z}^*) = \hat{\delta}\left(1 - \boldsymbol{c}^T \boldsymbol{C}^{-1}\boldsymbol{c} + \frac{(1 - \mathbf{1}^T \boldsymbol{C}^{-1}\boldsymbol{c})^2}{\mathbf{1}^T \boldsymbol{C}^{-1}\mathbf{1}}\right), \tag{4}$$

where $\boldsymbol{c}$ is the correlation vector between $\boldsymbol{z}^*$ and $\boldsymbol{Y}$, $\boldsymbol{C}$ is the correlation matrix of observed pairs, $\mathbf{1}$ is unit vectors, and the correlation between any two inputs $\boldsymbol{x}^i$, $\boldsymbol{x}^j$ is as follows:

$$c(\boldsymbol{x}^i, \boldsymbol{x}^j) = \exp\left(-\frac{\sum_{d=1}^{D} \theta_d |x_d^i - x_d^j|}{2l^2}\right), \tag{5}$$

where $\Theta = [\boldsymbol{\theta}, l]$ is the hyperparameter of GP. Θ is obtained through the maximum likelihood function, and its expression is as follows:

$$\begin{aligned} \log p(\Theta, \boldsymbol{Y}|\boldsymbol{X}) = &-\frac{1}{2} Y^T (\boldsymbol{C} + \delta^2 I)^{-1} \\ &- \frac{1}{2}\log|\boldsymbol{C} + \delta^2 I| - \frac{N}{2}\log 2\pi, \end{aligned} \tag{6}$$

EI is an infill criterion used to balance exploration and exploitation. Based on the predicted distribution of GP for a new input point $\boldsymbol{z}^*$, the calculation formula for EI is as follows:

$$\mathrm{EI}(\boldsymbol{z}^*) = \int_{-\infty}^{f_{\min}} I(y) \cdot \phi\left(\frac{y - \hat{y}(\boldsymbol{z}^*)}{\hat{\delta}(\boldsymbol{z}^*)}\right) dy, \tag{7}$$

$$I(y) = f_{\min} - \hat{y}(\boldsymbol{z}^*), \tag{8}$$

where f_{min} is the minimum value in $\boldsymbol{Y}$, ϕ is the probability density function. PoF is an infill criterion for handling ECMOPs. The expression of PoF is as follows:

$$\mathrm{PoF}(\boldsymbol{z}^*) = \prod_{i=1}^{q}\left[\Phi\left(\frac{0 - \hat{g}_i(\boldsymbol{z}^*)}{\hat{\delta}_{gi}(\boldsymbol{z}^*)}\right)\right], \tag{9}$$

where Φ is the cumulative distribution function, $\mathcal{N} \sim (\hat{g}_i(\boldsymbol{z}^*), \hat{\delta}_{gi}(\boldsymbol{z}^*))$ is predictive distribution of the ith constraint function.

III. Proposed Algorithm

This section presents an overview of EIC-MSSAEA and delves into its two key components: The GP-assisted multi-stage optimization and EIC. Fig. 2 provides a flowchart of EIC-MSSAEA to aid in understanding, specifically tailored to address the challenges of solving ECMOPs with two objectives and constraint functions.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

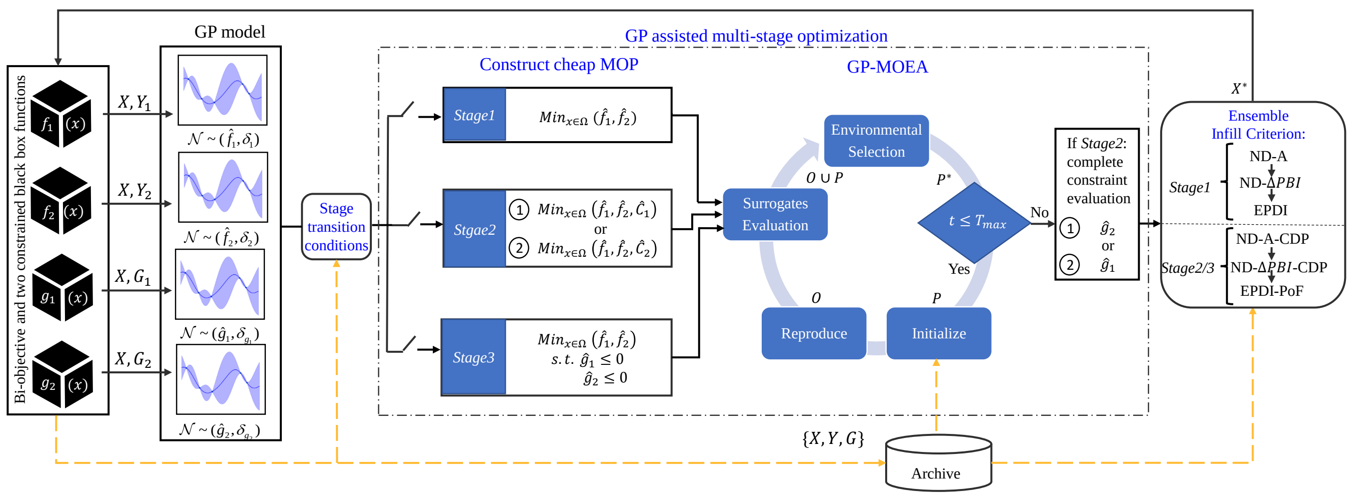


Fig. 2. A flowchart illustrating EIC-MSSAEA for an ECMOP with two objectives and constraints.

A. Overview of the Proposed Algorithm

As previously introduced, existing SAEAs that adopt a single infill criterion may experience performance degradation when dealing with ECMOPs with complex feasible regions. We propose an EIC that considers objectives, constraints, and uncertainty information of models to more effectively balance feasibility, convergence, diversity, exploration, and exploitation. Instead of directly optimizing the infill criterion, candidate solutions are obtained by searching based on surrogate models. After approximating a GP for each objective function and constraint function, the search process is divided into three stages, where the problem difficulty is gradually increased by increasing the number of constraints in each step. The search process is divided into multiple stages to explore potentially feasible regions and find high-quality solutions. The infill criterion is then applied to identify solutions that merit expensive FEs in each stage.

Algorithm 1 shows the pseudocode of EIC-MSSAEA. Before the iteration begins, an initialization phase is conducted. In this phase, N inputs, X, are obtained using the Latin Hypercube Sampling (LHS) method [50] (Line 1). Subsequently, X is evaluated to get the values of the objective and constraint functions (Line 2). All evaluated solutions are stored in an archive (A_r) (Line 3). Finally, the stage status flag is set to 1, indicating a sequential search starting from $Stage1$ (Line 4). The EIC-MSSAEA is an iterative algorithm consisting of four steps per round. A GP is constructed during the iteration stage for each objective and constraint function using the evaluated data (Line 6). The subsequent optimization stage is determined based on the available data information (Line 7). Each stage has a unique definition of the cost-effective optimization problem. The next step defines a cost-effective optimization problem through GP approximations of objective and constraint functions. During $Stage1$, the focus is solely on objective functions (Lines 8, 9). In $Stage2$, both the objective and a constraint function are optimized (Lines 14-16). Moving on to $Stage3$, all objectives and constraints are optimized (Lines 22, 23). Subsequently, the NSGA-III [37] is employed for solving the cost-effective optimization problems in $Stage1$ and $Stage2$, whereas the NSGA-III-CDP [51] is utilized for $Stage3$. This process yields a collection of promising candidate inputs denoted as X^* at each stage. It is worth noting that any MOEA can solve MOP_{Stage1} and MOP_{Stage2}, and CMOEA can solve MOP_{Stage3}. NSGA-III and NSGA-III-CDP are widely used in multi-objective problems and many-objective problems. We have made some comparisons in Section II-A of the Supplementary material to demonstrate their superiority. In the third step, an ensemble-based infill criterion is utilized to select the best set of candidate inputs, denoted as X^*, from the Pareto set P^* for expensive FEs (Lines 11, 19, 25). The fourth and final step involve subjecting the selected input X^* to an expensive evaluation to obtain the corresponding objective and constraint function values. Subsequently, the GP model is updated with the new data, and EIC-MSSAEA proceeds to the next loop.

B. The Gaussian Process-Assisted Multi-Stage Optimization

As shown in Fig. 1, infeasible obstacles can cause the algorithm to converge to local infeasible regions prematurely, thereby degrading the convergence performance, as shown in Fig. 1(a), and diversity performance, as shown in Fig. 1(b). Moreover, oversampling in the local infeasible area will lead to a decline in the global prediction accuracy of the objective and constraint surrogate models, thus affecting the performance of algorithm exploration and exploitation and wasting FEs. The root cause of these problems is the high complexity of constraints, which makes problem-solving difficult, and multiple constraints further exacerbate this issue. Reducing the number of constraints can increase the search range of the feasible space, promoting the population to explore more regions. Similar ideas such as [9], [26],

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

Algorithm 1: EIC-MSSAEA

Input: Ω, decision space; $F(\boldsymbol{x})$, objective functions; $g_j(\boldsymbol{x})$, $j \in \{1, ..., q\}$, constraint functions; FE_{max}, the maximum FEs; T_r, the parameter of stage transition; T_{max}, the number of generation; N_p, population size; N_m, the number of Monte Carlo samples.
Output: Expensive evaluated solutions

$X \leftarrow$ Get N initial inputs by LHS;
$\{Y, G\} \leftarrow$ Objective and constraint functions evaluation;
Initialize the archive $A_r = \{X, Y, G\}$, $FE = |A_r|$;
Initialize stage status flag $Stage = 1$;
while $FE < maxFE$ **do**
  $\hat{f}_i, \delta_i \leftarrow$ Each objective function is approximated by a GP($\{X, Y_i\}$), $i \in \{1, ..., m\}$, $\hat{g}_j, \delta_{g_j} \leftarrow$ Each constraint function is approximated by a GP($\{X, G_j\}$), $j \in \{1, ..., q\}$;
  $Stage \leftarrow$ `Stage Transition` (A_r, T_r);
  if $Stage == 1$ **then**
    Construct cheap MOP_{stage1};
    $P^* \leftarrow$ `GP-MOEA` (A_r, MOP_{Stage1}, T_{max}, V);
    $X^* \leftarrow$ `EIC` ($Stage$, P^*, A_r, N_m);
    $Y^*, \boldsymbol{G}^* \leftarrow$ Evaluate objective and constraint functions at X^*;
    $A_r \leftarrow A_r \bigcup \{X^*, Y^*, G^*\}, FE = |A_r|$;
  if $Stage == 2$ **then**
    $Index(j) \leftarrow$ Select the constraint function with the lowest feasible rate;
    Construct cheap MOP_{Stage2};
    $P^* \leftarrow$ `GP-MOEA` (A_r, MOP_{Stage2}, T_{max}, V);
    Obtain constraint function values that are not selected;
    $X^* \leftarrow$ `EIC` ($Stage$, P^*, A_r, N_m);
    $Y^*, \boldsymbol{G}^* \leftarrow$ Evaluate objective and constraint functions at X^*;
    $A_r \leftarrow A_r \bigcup \{X^*, Y^*, G^*\}, FE = |A_r|$;
  if $Stage == 3$ **then**
    Construct cheap MOP_{stage3};
    $P^* \leftarrow$ `GP-MOEA` (A_r, MOP_{Stage3}, T_{max}, V);
    $X^* \leftarrow$ `EIC` ($Stage$, P^*, A_r, N_m);
    $Y^*, G^* \leftarrow$ Evaluate objective and constraint functions at X^*;
    $A_r \leftarrow A_r \bigcup \{X^*, Y^*, G^*\}, FE = |A_r|$;

[52], [53] have achieved competitive results, assuming the objective and constraint functions are inexpensive. However, it is challenging to accomplish ECMOPs with only a few hundred FEs. Therefore, the GP-assisted multi-stage optimization is proposed to search for the optimal feasible region.

After building a GP model for each objective and constraint function, the Algorithm 2 introduces the `Stage Transition` to reallocate computational resources and avoid wasting FEs. `Stage Transition` is proposed for two primary reasons. Firstly, in certain ECMOPs, the CPF and UPF do not intersect, and there is a significant distance between them. Neglecting constraints in $Stage1$ can cause the search to deviate from the CPF, as shown in Fig. 3(a), resulting in wasted FEs. Secondly, after reducing the constraints, the sub-CPF still does not intersect with the CPF, making the search in $Stage2$ ineffective. To avoid wasting FEs caused by solving the

Algorithm 2: `Stage Transition` (A_r, T_r)

Calculate $\Delta C(\boldsymbol{x})$ by equation (10);
Calculate F_r ;
if $FE == 3 \cdot T_r$ **then**
  if $\Delta C(\boldsymbol{x}) < 0$ *or* $F_r > 0$ **then**
    $Stage = 1$;
  else if $\Delta C(\boldsymbol{x}) \geq 0$ *and* $F_r = 0$ **then**
    $Stage = 3$;
else if $\tau_1 FE_{max} < FE \leq \tau_2 FE_{max}$ *and* $Stage == 1$ **then**
  $Stage = 2$;
else if $FE > \tau_2 FE_{max}$ **then**
  $Stage = 3$;

problem with CPF and UPF being far apart in $Stage1$ and $Stage2$, EIC-MSSAEA first detects whether the CPF of the current problem is far from the UPF based on the difference of the constraint violation values $\Delta C(\boldsymbol{x})$ and the feasibility ratio F_r when the number of FEs reaches $3 \cdot T_r$ ($T_r \ll \tau_1 FE_{max}$) in $Stage1$, using the data obtained from the previous T_r rounds. $\Delta C(\boldsymbol{x})$ and F_r are calculated as:

$$\Delta C(\boldsymbol{x}) = \sum_{i=\frac{T_r}{2}+1}^{T_r} C^i_{\min}(\boldsymbol{x}) - \sum_{i=1}^{\frac{T_r}{2}} C^i_{\min}(\boldsymbol{x}), \tag{10}$$

$$F_r = \frac{N_f}{N_{T_r}}, \tag{11}$$

where $C^i_{\min}(\boldsymbol{x})$ represents the minimum constraint violation value in solution X^* obtained in ith iteration. N_{T_r} represents the number of solutions obtained after T_r rounds at $Stage1$, usually $N_{T_r} = 3 \cdot T_r$. N_f denotes the number of feasible solutions among these solutions.

Second, if $\Delta C(\boldsymbol{x}) \geq 0$ and $F_r = 0$, it indicates that the search in $Stage1$ deviates from CPF, terminating the current search phase $Stage1$ and assigning the remaining FEs to $Stage3$; If not, then continue searching from $Stage1$ until the number of FEs reach $\tau_1 FE_{max}$.

After $Stagei$ is determined, an inexpensive MOP based on GP is developed (MOP_{Stagei}, $i \in \{1, 2, 3\}$), which is different for each stage. For example, in complex ECMOPs, the CPF is typically located near UPF. Therefore, in $Stage1$, only objectives are considered to converge quickly near the UPF and overcome infeasible obstacles. The MOP_{Stage1} is as follows:

$$\min_{\boldsymbol{x} \in \Omega} \ (\hat{f}_1(\boldsymbol{x}), ..., \hat{f}_m(\boldsymbol{x})), \tag{12}$$

where $\hat{f}_i(\boldsymbol{x})$ is the GP predicted mean function (equation (3)) for the ith objective function. The MOP_{Stage1} is then solved using the NSGA-III to obtain potential solutions P^*. Algorithm 3 provides the details of `GP-MOEA`. In the initialization phase (Lines 1-6), a set of reference vectors V is first generated and obtains the minimum value corresponding to each objective in the A_r. Finally, the initial population is selected from the A_r using the

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

Algorithm 3: GP-MOEA (A_r, MOP_{Stagei}, T_{max}, N)

1 $V \leftarrow$ Generate a set of reference vectors;
2 $Z \leftarrow$ Get the minimum value corresponding to each objective in the A_r;
3 **if** $Stage == 3$ **then**
4 $\quad P \leftarrow$ Environmental selection of NSGA-III-CDP(A_r, Z, V);
5 **else**
6 $\quad P \leftarrow$ Environmental selection of NSGA-III(A_r, Z, V);
7 $t = 1$;
8 **while** $t < T_{max}$ **do**
9 $\quad O \leftarrow$ Reproduction (P);
10 $\quad P \leftarrow O \bigcup P$;
11 $\quad$ Evaluation by MOP_{Stagei};
12 $\quad$ **if** $Stage == 3$ **then**
13 $\quad\quad P \leftarrow$ Environmental selection of NSGA-III-CDP(A_r, Z, V);
14 $\quad$ **else**
15 $\quad\quad P \leftarrow$ Environmental selection of NSGA-III(A_r, Z, V);
16 $\quad t = t + 1$;

environmental selection method of NSGA-III. In the evolutionary stage (Lines 8-16), an offspring population O is generated by simulated binary crossover and polynomial mutation. The parents and offsprings are merged to create a new population P. Next, the MOP_{Stage1} is used to evaluate P and update the minimum value Z. Finally, the environmental selection is performed to select the elite population P and start the next evolution. When the evolution loop reaches T_{max}, the final population, P, is output.

In $Stage1$, the search is pushed to the vicinity of UPF. To explore more locally feasible regions and improve diversity, we only consider a constraint with the lowest feasible ratio in the $Stage2$ (Line 15 Algorithm 1). We increase the search in the sub-CPF areas by using infeasible solutions to converge to the CPF quickly. Therefore, the MOP_{Stage2} constructed in $Stage2$ is as follows:

$$\min_{\boldsymbol{x}\in\Omega} \ (\hat{f}_1(\boldsymbol{x}), ..., \hat{f}_m(\boldsymbol{x}), \hat{C}_{Index(j)}(\boldsymbol{x})), \tag{13}$$

$$\hat{C}_{Index(j)}(\boldsymbol{x}) = \max(0, \hat{g}_j(\boldsymbol{x})), \tag{14}$$

where $\hat{g}_j(\boldsymbol{x})$ is the constraint function with the lowest feasible ratio, approximated by GP.

In $Stage2$, sufficient feasible regions are discovered; therefore, in $Stage3$, it is necessary to improve the feasibility, convergence, and diversity of the solution. Thus, the MOP_{Stage3} constructed in $Stage3$ is as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}\in\Omega} \ (\hat{f}_1(\boldsymbol{x}), ..., \hat{f}_m(\boldsymbol{x})) \\ &s.t. \ \ \hat{g}_j(\boldsymbol{x}) \leq 0, j \in \{1, ..., q\}. \end{aligned} \tag{15}$$

The blue arrows in Fig. 3 (b)-(d) show the behavior of three stages.

Algorithm 4: EIC($Stage, P^*, A_r, N_m$)

if $Stage == 1$ **then**
$\quad$ Select $\boldsymbol{x}_1^*$ using ND-A;
$\quad$ Delete $\boldsymbol{x}_1^*$ from P_{nd}^* and store it in A_{rnd};
$\quad$ Select $\boldsymbol{x}_2^*$ using ND-ΔPBI;
$\quad$ Delete $\boldsymbol{x}_1^*$ from P_{nd}^* and store it in A_{rnd};
$\quad$ Select $\boldsymbol{x}_3^*$ using EPDI;
$\quad x^* \leftarrow \{\boldsymbol{x}_1^* \cup \boldsymbol{x}_2^* \cup \boldsymbol{x}_2^*\}$
else
$\quad$ Select $\boldsymbol{x}_1^*$ using ND-A-CDP;
$\quad$ Delete $\boldsymbol{x}_1^*$ from P_{nd}^* and store it in A_{rnd};
$\quad$ Select $\boldsymbol{x}_2^*$ using ND-ΔPBI-CDP;
$\quad$ Delete $\boldsymbol{x}_1^*$ from P_{nd}^* and store it in A_{rnd};
$\quad$ Select $\boldsymbol{x}_3^*$ using EPDI-PoF;
$\quad x^* \leftarrow \{\boldsymbol{x}_1^* \cup \boldsymbol{x}_2^* \cup \boldsymbol{x}_2^*\}$

C. Ensemble Infill Criterion

Considering the intricate nature of the CPF and landscape, it becomes crucial to incorporate various factors such as feasibility, convergence, diversity, and model uncertainty at different stages of the optimization process. These factors serve as guidance for the search for finding the optimal CPF. Drawing inspiration from ensemble learning methods, different types of models combined as ensemble members, leveraging their diversity to enhance prediction accuracy. We propose the EIC, which integrates multiple infill criteria balancing feasibility, convergence, diversity, exploration and exploitation. Therefore, it improves the accuracy of selecting a good solution. By utilizing EIC, we can effectively choose candidate solutions during different stages of the search process. The pseudocode of EIC is provided in algorithm 4.

We provide a detailed introduction to each component of the EIC in $Stage1$. The first criterion, ND-A, involves ranking the solutions in the candidate solutions set P^* and the archive A_r using non-dominated sorting, resulting in the non-dominated solution sets P_{nd}^* and A_{rnd}. The subsequent step calculates the angle θ_x^{min} between each solution in P_{nd}^* and the A_{rnd} solution set in the objective space. The calculation of θ_x^{min} is as follows:

$$\theta_{xy} = \arccos \frac{\sum_{i=1}^{m} (\hat{f}_i(x) \cdot f_i(y))}{\sqrt{\sum_{i=1}^{m} \hat{f}_i(x)^2} \cdot \sqrt{\sum_{i=1}^{m} f_i(y)^2}}, \tag{16}$$

$$\theta_x^{\min} = \min_{y\in A_{rnd}, x\in P_{nd}^*} \theta_{xy}, \tag{17}$$

A solution $\boldsymbol{x}_1^*$ with the largest angle is then selected from P_{nd}^*. ND-A utilizes non-dominated sorting and selecting the maximum angle to improve convergence and diversity. $\boldsymbol{x}_1^*$ is stored in A_{rnd} along with its predicted objective values and removed from P_{nd}^* to enhance the diversification of solutions. The process of selecting subsequent solutions then follows a serial way. Notably, adopting a parallel strategy for simultaneously selecting multiple sampling points is worth considering. A comparison between these two methods is available in Section II-B of the Supplementary material. The second infill criterion member ND-ΔPBI is proposed to select

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

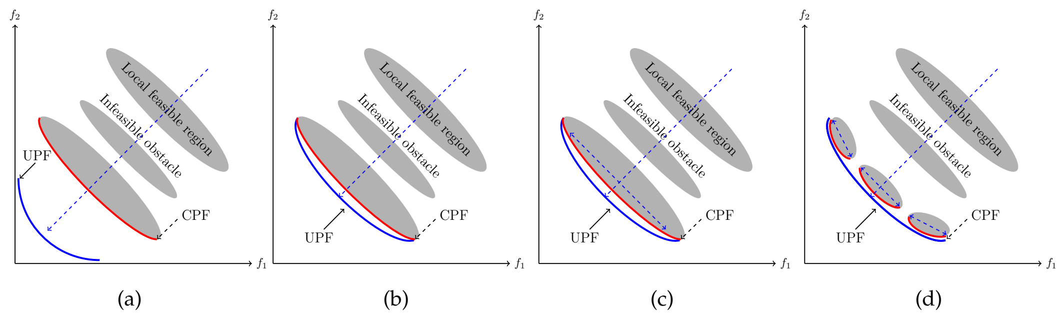


Fig. 3. An Illustration of the search behavior at different stages. (a) In $Stage1$, MOP_{Stage1} ignores the constraints and guides the search towards the objective optimization, as indicated by the blue dashed arrows. For the ECMOP, the CPF is disjointed and far from the UPF, so searching near the UPF wastes computational resources. (b) For the ECMOP, the UPF is close to the CPF, so searching near the UPF helps to find the CPF. (c) When converging from a distant location to the UPF, $Stage2$ is initiated. In $Stage2$, MOP_{Stage2} considers only one constraint. It intensifies the search around the sub-CPF, as shown by the blue dashed arrows perpendicular to the direction of the objective optimization. (d) The search around the sub-CPF can exploit infeasible solutions to increase diversity. Finally, in $Stage3$, MOP_{Stage3} uses all constraints. It ensures that the search stays within the feasible region, as indicated by the blue dashed arrows in each part of the optimal feasible region.

the second candidate solution $\boldsymbol{x}_2^*$. We normalize P_{nd}^* and A_{rnd}, arrange the solutions in A_{rnd} based on their vertical distance to the reference vectors V, identify the nonempty reference vectors V_n, and calculate the PBI value for each solution in A_{rnd}.

$$d_{k,1}(\boldsymbol{x}) = \frac{|\tilde{F}(\boldsymbol{x}) \cdot V_k|}{|V_k|}, \tag{18}$$

$$d_{k,2}(\boldsymbol{x}) = \left|\tilde{F}(\boldsymbol{x}) - d_{k,1}(\boldsymbol{x})\frac{V_k}{|V_k|}\right|, \tag{19}$$

$$PBI_k(\boldsymbol{x}) = d_{k,1}(\boldsymbol{x}) + 5d_{k,2}(\boldsymbol{x}), \tag{20}$$

where k is a nonempty reference vector index, and $\tilde{F}(x)$ represents the normalized objective vector. Using the same method, arrange the solutions in P_{nd}^* into V_n and calculate the $P\hat{B}I$ value of each solution in P_{nd}^*. Finally, calculate the ΔPBI and select the maximum solution for ΔPBI improvement to evaluation.

$$\Delta PBI_j(\boldsymbol{x}) = PBI_{jmin} - P\hat{B}I_j(\boldsymbol{x}), j \in \{1, ..., |V_n'|\}, \tag{21}$$

where PBI_{jmin} is the minimum PBI value associated with the reference vector V_j'. V_n' are the nonempty reference vectors, and $V_n' \subseteq V_n$ (The explanation of $V_n' \subseteq V_n$ can be found in Section II-C of the Supplementary material). ND-ΔPBI selects the solution $\boldsymbol{x}_2^*$ with the highest PBI improvement from all candidate solutions associated with nonempty reference vectors to improve convergence and diversity. After that $\boldsymbol{x}_2^*$ is stored in the A_{rnd}, and then delete it from candidate solutions P_{nd}^*.

The third component of EIC is called EPDI, which considers model uncertainty to balance exploration and exploitation. The proximity and diversity (PD) function can effectively measure convergence and diversity, and compared to other scalarized functions, PD has better versatility [54]. The PD function is defined as follows:

$$PD(\hat{F}(\boldsymbol{x}), V_r) = \frac{1}{m}\sum_{i=1}^{m}\hat{f}_i(\boldsymbol{x}) + 5\left|\hat{F}(\boldsymbol{x})\right|_2 \sin(\hat{F}(\boldsymbol{x}), V_r) \tag{22}$$

where $\hat{F}(\boldsymbol{x})$ is the predicted objective vector. Unlike V_k, V_r is a random selection of reference vector. $\sin(\hat{F}(\boldsymbol{x}), V_r)$ is used to calculate the vertical distance from $\hat{F}(\boldsymbol{x})$ to V_r. EPDI is an extension of EI in the context of multi-objective optimization. The improvement function PDI is defined as follows:

$$PDI(\hat{F}(\boldsymbol{x}), V_r) = \max(PD_{\min} - PD(\hat{F}(\boldsymbol{x}), V_r), 0), \tag{23}$$

where PD_{min} is the minimum value evaluated by PD function for the solution in A_r. Based on equations (7) and (23), EPDI can be derived as follows:

$$EPDI(\boldsymbol{x}, \hat{F}, V_r) = \int\limits_{y \in m} PDI(\hat{F}, V_r)\prod_{i=1}^{m}\frac{1}{\delta_i}\phi(\frac{y_i - \hat{y}_i}{\delta_i})dy_i. \tag{24}$$

The Monte Carlo sampling method is used to calculate equation (24), and the number of samples is 1000. Finally, the solution $\boldsymbol{x}_3^*$ with the highest EPDI value is selected.

In $Stage2$ and $Stage3$, the feasibility of solutions becomes a priority for each member of the EIC. Therefore, in ND-A and ND-ΔPBI, the CDP is applied to P^* and A_r to select the best feasible solutions. Subsequently, the maximum angle and ΔPBI criteria are used to choose the best solution for evaluation. Consequently, ND-A and ND-ΔPBI are referred to as ND-A-CDP and ND-ΔPBI-CDP in $Stage2$ and $Stage3$ respectively. In the third member of EIC, uncertain information of the constraint functions are also considered. If there are no feasible solutions in A_r, PoF is first utilized to find a feasible solution, and then the product of EPDI and PoF is used as the infill criterion. To provide a clearer understanding of EIC, the roles of each EIC member are illustrated in Fig. 4. $Stage1$ ignores constraints, while in $Stage2$ and $Stage3$, constraints have higher priority. $Stage1$ and $Stage2/Stage3$ aim to balance the objective and constraints. Additionally, the six members collectively achieve a balance of feasibility, convergence, diversity, exploration, and exploitation.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

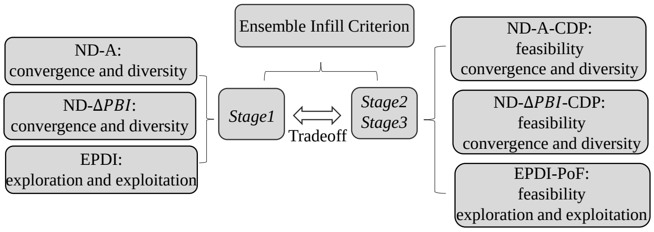


Fig. 4. An illustration delineates each member's roles and responsibilities within the EIC.

IV. Experimental Studies

A. Experimental Setting

To evaluate the performance of EIC-MSSAEA, we conduct a sensitivity analysis of its crucial parameters. Subsequently, ablation studies examine the effects of the GP-assisted multi-stage optimization and EIC. Furthermore, EIC-MSSAEA is compared with seven state-of-the-art SAEAs, including ASA-MOEA/D [41], MGSAEA [44], KTS [40], USeMOC [35], HSMEA [49], EIM-PoF [29], and MultiObjectiveEGO [28], on LIRCMOP [55], MW [56], C-DTLZ [51] benchmark problems and a real-world optimization problem. In addition, EIC-MSSAEA is compared with three CMOEAs, including MCCMO, C3M, and MSCMO on the benchmark problems. As the first stage of EIC-MSSAEA, EIC-S1 can be independently applied to EMOPs. Therefore, we compare its performance with leading SAEAs on DTLZ and WFG [57] test suites, such as ABSAEA, EIMEGO, and NSGAIII-EHVI [36]. All experiments are performed in PlatEMO [58] and independently run 20 times.

This work assesses the performance of the algorithm using the inverted generational distance (IGD) [59] and HV [60]. A set of 10,000 reference points is generated to calculate the IGD values. For HV evaluation, reference points are fixed as (1.1,...,1.1). Furthermore, statistical analysis is conducted using the Wilcoxon rank sum test to determine if a significant difference exists between the proposed EIC-MSSAEA and the compared algorithm, with a significance level of 5%. The symbols '-', '+', and '=' indicate that the compared algorithm is significantly inferior, superior, and comparable to EIC-MSSAEA, respectively.

Table I provides the parameter settings used in EIC-MSSAEA. The parameter settings of the other algorithms under comparison are given in the Supplementary material.

B. Sensitivity Analysis of τ_1, τ_2, T_r

EIC-MSSAEA introduces three additional parameters: τ_1, τ_2, and T_r. τ_1 determines the maximum number of FEs consumed in $Stage1$, while τ_2 determines the number of FEs consumed in $Stage2$. T_r determines the positional relationship between CPF and UPF. Different values of τ_1, τ_2, and T_r can influence the performance of EIC-MSSAEA. Due to space limit, sensitivity analysis is presented in Section III-B of the Supplementary Material.

TABLE I
PARAMETER SETTINGS OF EIC-MSSAEA.

Parameter name	Meaning	Value
N	The number of initial inputs by LHS	$11D$-1
N_p	Population size	100
FE_{max}	Maximum number of FEs for ECMOPs	400
	Maximum number of FEs for EMOPs	300
T_r	the parameter of `Stage Transition`	8
τ_1	Number of FEs at $Stage1$ termination	0.5
τ_2	Number of FEs at $Stage2$ termination	0.7
N_m	Number of Monte Carlo samples	1000
T_{max}	Number of generation for NSGA-III	20
η_c	Distribution index of SBX	20
η_m	Distribution index of PM	20
p_c	The crossover probability	1
p_m	The mutation probability	$1/D$

C. Effect of the Gaussian Process-Assisted Multi-Stage Optimization

EIC-MSSAEA combines three independent optimization processes, allocating a certain amount of FEs to each stage. To assess the effectiveness of each stage, we compare the combination of $Stage2$ and $Stage3$ (referred to as EIC-S2S3) with $Stage3$ alone (referred to as EIC-S3), using the third stage as a baseline. This comparison allows us to evaluate the impact of $Stage2$. Subsequently, we compare EIC-S2S3 with the complete EIC-MSSAEA that includes all three stages to examine the effectiveness of $Stage1$. The experiments are conducted on the LIRCMOP test suite, with parameter settings provided in Table I. We perform 20 independent experiments and present the statistical test results for the IGD in the second to fifth columns of Table II.

1) EIC-S3 VS EIC-S2S3: Compared to EIC-S3, EIC-S2S3 focuses on the optimization process in the second stage, which considers only one constraint. This simplification reduces the complexity of the problem. Typically, ECMOPs involve multiple constraints. Assuming that each constraint has the same difficulty, the complexity of solving ECMOPs depends on the number of constraints. The more constraints there are, the more challenging it becomes to solve ECMOPs. In contrast, problems with only one constraint are relatively easier to solve. EIC-S2S3 leverages infeasible solutions to accelerate convergence and discover more feasible regions by considering only one constraint instead of all constraints simultaneously. As shown in the second and third columns of Table II, EIC-S2S3 outperforms EIC-S3 in 8 out of 14 test instances, verifying the effectiveness of $Stage2$.

2) EIC-S2S3 VS EIC-MSSAEA: Compared to EIC-S2S3, EIC-MSSAEA adds an extra optimization stage ($Stage1$) without constraints. This stage aids the search process in overcoming infeasible obstacles and converging toward the vicinity of UPF. Subsequently, guided by $Stage2$ and $Stage3$, it facilitates rapid convergence towards CPF near the UPF. The fourth and fifth columns of Table II show that EIC-MSSAEA outperformed EIC-S2S3 in 7 out of 14 test cases. These experimental results provide evidence of the effectiveness of $Stage1$.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

TABLE II
THE IGD VALUES OBTAINED BY EIC-S3, EIC-S2S3, EICP1-MSSAEA, EICP2-MSSAEA, EICP3-MSSAEA AND EIC-MSSAEA ON LIRCMOP TEST SUITE.

Problem	EIC-S3	EIC-S2S3	EIC-S2S3	EIC-MSSAEA	EICp1-MSSAEA	EICp2-MSSAEA	EICp3-MSSAEA	EIC-MSSAEA
LIRCMOP1	**1.7464e-1 (1.02e-1) +**	4.4046e-1 (2.04e-1)	4.4046e-1 (2.04e-1) -	**2.2246e-1 (1.66e-1)**	2.8898e-1 (1.06e-1) -	3.8626e-1 (6.39e-2) -	**1.7845e-1 (1.58e-1) +**	2.2246e-1 (1.66e-1)
LIRCMOP2	**2.2657e-1 (1.76e-1) +**	4.3304e-1 (2.17e-1)	4.3304e-1 (2.17e-1) -	**1.9448e-1 (1.10e-1)**	3.1968e-1 (5.66e-2) -	3.6744e-1 (7.39e-2) -	**1.7333e-1 (1.22e-1) +**	1.9448e-1 (1.10e-1)
LIRCMOP3	**1.7228e-1 (7.98e-2) +**	3.6185e-1 (1.79e-1)	3.6185e-1 (1.79e-1) =	**2.4271e-1 (1.49e-1)**	4.1927e-1 (1.53e-1) -	4.3677e-1 (1.08e-1) -	**2.1531e-1 (1.22e-1) +**	2.4271e-1 (1.49e-1)
LIRCMOP4	**1.8797e-1 (8.83e-2) +**	3.3449e-1 (1.68e-1)	3.3449e-1 (1.68e-1) -	**2.2224e-1 (1.60e-1)**	3.0030e-1 (1.49e-1) -	4.3401e-1 (1.49e-1) -	**1.7329e-1 (6.59e-2) =**	2.2224e-1 (1.60e-1)
LIRCMOP5	6.3942e-1 (4.69e-1) -	**7.6053e-2 (2.93e-2)**	7.6053e-2 (2.93e-2) -	**5.5751e-2 (1.33e-2)**	2.5452e-1 (7.49e-2) -	2.6867e-1 (6.93e-2) -	8.7510e-2 (2.59e-2) -	**5.5751e-2 (1.33e-2)**
LIRCMOP6	7.0960e-1 (4.62e-1) -	**9.1151e-2 (8.04e-2)**	9.1151e-2 (8.04e-2) -	**5.0099e-2 (1.18e-2)**	2.8014e-1 (1.18e-1) -	3.0623e-1 (9.03e-2) -	6.2582e-2 (1.93e-2) -	**5.0099e-2 (1.18e-2)**
LIRCMOP7	1.1315e-1 (5.82e-2) =	**8.9023e-2 (1.58e-2)**	**8.9023e-2 (1.58e-2) =**	1.0509e-1 (3.45e-2)	2.6251e-1 (1.72e-1) -	3.8720e-1 (2.75e-1) -	1.2806e-1 (4.84e-2) -	**1.0509e-1 (3.45e-2)**
LIRCMOP8	2.8844e-1 (3.66e-1) =	**1.2576e-1 (9.09e-2)**	1.2576e-1 (9.09e-2) =	**9.2308e-2 (5.13e-2)**	3.7316e-1 (2.58e-1) -	5.2406e-1 (2.52e-1) -	1.4080e-1 (1.27e-1) =	**9.2308e-2 (5.13e-2)**
LIRCMOP9	6.3884e-1 (1.44e-1) -	**2.8412e-1 (1.02e-1)**	2.8412e-1 (1.02e-1) -	**2.0538e-1 (9.25e-2)**	7.6741e-1 (2.93e-1) -	4.3346e-1 (8.41e-2) -	2.1736e-1 (7.50e-2) =	**2.0538e-1 (9.25e-2)**
LIRCMOP10	5.6523e-1 (1.60e-1) -	**1.1155e-1 (4.59e-2)**	1.1155e-1 (4.59e-2) -	**5.1161e-2 (4.64e-2)**	6.6037e-1 (1.15e-1) -	2.6415e-1 (1.19e-1) -	8.4901e-2 (5.17e-2) =	**5.1161e-2 (4.64e-2)**
LIRCMOP11	5.2529e-1 (1.01e-1) -	**2.0342e-1 (1.21e-1)**	2.0342e-1 (1.21e-1) =	**2.0073e-1 (1.28e-1)**	5.9159e-1 (1.77e-1) -	2.8821e-1 (1.23e-1) -	2.9709e-1 (1.12e-1) -	**2.0073e-1 (1.28e-1)**
LIRCMOP12	4.6148e-1 (1.73e-1) -	**2.2149e-1 (7.05e-2)**	2.2149e-1 (7.05e-2) =	**2.0907e-1 (5.01e-2)**	9.1358e-1 (2.80e-1) -	4.7785e-1 (1.94e-1) -	2.7715e-1 (1.07e-1) -	**2.0907e-1 (5.01e-2)**
LIRCMOP13	6.3559e-1 (5.87e-1) -	**9.8047e-2 (8.60e-3)**	9.8047e-2 (8.60e-3) =	**9.0673e-2 (7.55e-3)**	8.1277e-1 (2.57e-1) -	3.3035e-1 (1.19e-1) -	1.0839e-1 (9.70e-3) -	**9.0673e-2 (7.55e-3)**
LIRCMOP14	6.5060e-1 (5.68e-1) -	**1.3330e-1 (1.93e-2)**	1.3330e-1 (1.93e-2) =	**1.1661e-1 (1.00e-2)**	8.9059e-1 (3.19e-1) -	4.5758e-1 (1.52e-1) -	1.4853e-1 (2.08e-2) -	**1.1661e-1 (1.00e-2)**
$+/-/=$	4/8/2		0/7/7		0/14/0	0/14/0	3/7/4	

The gray background represents the best result for each test instance. The second and third columns show the comparative results between EIC-S3 and EIC-S2S3. Similarly, the fourth and fifth columns show the comparisons between EIC-S2S3 and EIC-MSSAEA. Finally, the sixth to last columns show the comparisons of EICp1-MSSAEA, EICp2-MSSAEA, and EICp3-MSSAEA with EIC-MSSAEA.

D. Effect of Ensemble Infill Criterion

We conduct a comparative analysis using three variants, namely EICp1-MSSAEA, EICp2-MSSAEA, and EICp3-MSSAEA, to investigate the role of EIC. Each variant utilizes a different pair of members in the EIC as the infill criterion. Specifically, EICp1-MSSAEA replaces each member of EIC with ND-A in $Stage1$ and ND-A-CDP in $Stage2$ and $Stage3$. EICp2-MSSAEA replaces each member of EIC with ND-ΔPBI in $Stage1$ and ND-ΔPBI-CDP in $Stage2$ and $Stage3$. EICp3-MSSAEA, on the other hand, utilizes EPDI in $Stage1$ and EPDI-PoF in $Stage2$ and $Stage3$. These experiments are conducted on the LIRCMOP test suite, and the specific experimental settings are outlined in Table I. The results are summarized in the sixth to last columns of Table II. It is observed that EIC-MSSAEA outperformed EICp1-MSSAEA, EICp2-MSSAEA, and EICp3-MSSAEA in 14, 14, and 7 out of 14 test cases, respectively. EICp1-MSSAEA and EICp2-MSSAEA consider feasibility, convergence, and diversity as infill criteria. Still, they do not take into account the uncertainties associated with objective and constraint functions to achieve a balance between exploration and exploitation. On the other hand, EICp3-MSSAEA effectively controls exploration and exploitation; however, it cannot simultaneously balance feasibility, convergence, diversity, exploration, and exploitation. Additionally, EICp3-MSSAEA requires Monte Carlo integration, leading to lower computational efficiency than the first two variants. Given the limitations of each EIC member, combining these members compensates for their shortcomings. The advantages of using EIC are further highlighted based on the obtained results and analysis.

We perform additional analysis below to fully understand the role of each member in the algorithm. First, we replace each member with ND-A, ND-PBI, and EPDI, resulting in three distinct variants: EICp11, EICp21, and EICp31. By comparing the performance of these variants, we can identify the specific contributions of each member within $Stage1$. Then, we replace each member with ND-A-CDP, ND-PBI-CDP, and EPDI-PoF, creating the variants EICp12, EICp22, and EICp32. Through a comparative evaluation of these variants, we elucidated the distinct roles played by each member within $Stage2$ and $Stage3$. For a comprehensive assessment of algorithmic performance, we used performance indicators such as the generation distance (GD) [61], pure diversity (PD) [62], feasibility ratio (FR), IGD, and CPU time. The experimental results are summarized in Tables SV to SIX in the Supplementary material.

Fig. 5(a) clearly shows the average rankings for each performance indicator across EICp11, EICp21, and EICp31. EICp11 achieves the highest ranking in FR because EICp11 uses elite solutions as reference points and selects solutions around them based on the angle. The contribution of EICp11 to the effective maintenance of feasible solutions is underlined. In terms of GD, EICp21 secures the top position. EICp21 plays a key role in improving convergence performance. Although EICp11, EICp21, and EICp31 all consider convergence and diversity, EICp31 additionally incorporates uncertain information from the objective models to improve exploration and exploitation performance. EICp31 achieves the highest ranking in PD and IGD, illustrating its contribution. However, EICp31 has the lowest computational efficiency.

EICp12, EICp22, and EICp32 are constraints based on EICp11, EICp21, and EICp31, respectively. Therefore, the feasible ratios of EICp12, EICp22, and EICp32 are generally higher than those of EICp11, EICp21, and EICp31, respectively. Figure 5(b) summarizes the average ranking results of EICp12, EICp22, and EICp32 on LIRCMOP for each metric. EICp12 ranks the first in the FR, indicating that it contributes the most to maintaining feasible solutions in the feasible region. EICp22 ranks first in the GD, indicating that it contributes the most to improving convergence performance. Although EICp12, EICp22, and EICp32 all consider feasibility, convergence,

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

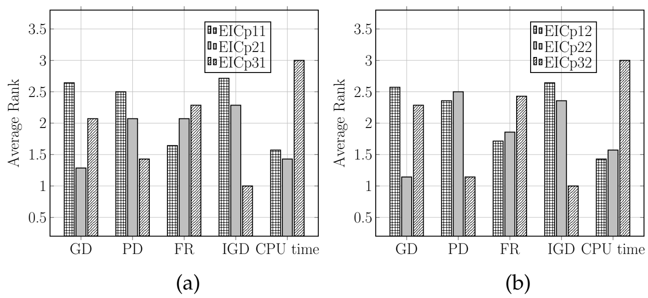


Fig. 5. (a) is the average ranking of EICp11, EICp21, and EICp31 on LIRCMOP; (b) is the average ranking of EICp12, EICp22, and EICp32 on LIRCMOP.

and diversity, EICp32 additionally incorporates uncertain information from the objective and constraint models to improve exploration and exploitation performance. EICp32 achieves the highest ranking in PD and IGD, illustrating its contribution. However, EICp32 ranks last in CPU time.

E. Comparisons with Peers for Expensive Constrained Multi-Objective Optimization Problems

In this section, we compare EIC-MASSAEA with seven SAEAs (ASA-MOEA/D, MGSAEA, KTS, USeMOC, HSMEA, EIM-PoF, and MlutiObjectiveEGO) and three CMOEAs (MCCMO, C3M, MSCMO) on LIRCMOP MW, C-DTLZ and the practical application to evaluate the performance of the proposed algorithm.

1) Results of LIRCMOP test suite: As shown in Table III, the IGD values of EIM-MSSAEA are better than ASA-MOEA/D, MGSAEA, KTS, USeMOC, HSMEA, EIM-PoF, and MultiObjectiveEGO in 14, 11, 9, 12, 10, 12, and 12 out of 14 cases on LIRCMOP test suite, respectively. In LIRCMOP1-4, the UPF and CPF of each problem are disjointed and far apart. MGSAEA fails to find feasible solutions in LIRCMOP3 and LIRCMOP4. Because its initial search phase and the search phase near the feasible region do not consider constraints, too many FEs are wasted in these two phases. As a result, feasible solutions cannot be generated and maintained. MultiObjectiveEGO, with its selection function, cannot effectively handle objectives and constraints, so it cannot find feasible solutions in LIRCMOP3 and LIRCMOP4. HSMEA and EIM-PoF adopt a CHT where constraint priority is higher than the objective, enabling them to find feasible solutions within a few FEs. However, their convergence and diversity are worse than EIC-MSSAEA. The CPF and UPF of LIRCMOP5 and LIRCMOP6 coincide, which means these two test cases have low requirements for constraint processing. In LIRCMOP6, MGSAEA achieves the best performance. HSMEA also achieves relatively good performance. It stands out among other algorithms as it does not utilize uncertainty information but instead employs multiple models to enhance the approximation performance of the surrogate model, effectively saving FEs for exploring unknown feasible regions.

On the other hand, USeMOC, EIM-PoF, and MultiObjectiveEGO exhibit poor convergence performance, failing to converge near UPF on LIRCMOP5 and LIRCMOP6. LIRCMOP7 and LIRCMOP8 present multiple significant infeasible obstacles, with the UPF located within the infeasible region. EIC-MSSAEA demonstrated the best performance in these test cases, attributed to its robust capability to infeasible obstacles. LIRCMOP9-12 exhibit discontinuous CPFs. EIC-MSSAEA attaines the best performance in LIRCMOP10-12 by mitigating the complexity of the problem and enhancing the exploration capability of the algorithm through constraint selection. LIRCMOP13 and LIRCMOP14 are MOPs with three objectives. Compared to LIRCMOP1-12, these problems are more complex. Our algorithm also achieves the best performance in terms of all three objectives. Furthermore, Fig. 6 illustrates the solution distribution corresponding to the minimum IGD achieved by all algorithms on LIRCMOP13. EIC-MSSAEA obtains the most significant number of solutions located on the CPF. This outcome reinforces the effectiveness of our approach.

2) Results of MW test suite: As shown in Table III, the IGD values of EIC-MSSAEA are better than ASA-MOEA/D, MGSAEA, KTS, USeMOC, HSMEA, EIM-PoF, and MultiObjectiveEGO in 14, 10, 9, 13, 11, 11, and 14 out of 14 cases on MW test suite, respectively. On MW1 and MW2, the feasible region is tiny, and the initial sampling fails to obtain a feasible solution. As a result, the initial constrained surrogate models have significant errors. Our algorithm ignores the constraint function in the first stage, reducing the error guidance caused by the constraint model. Therefore, within 400 FEs, our algorithm achieves the best performance on MW1 and MW2. It is challenging to model and optimize for ECMOPs with multimodal properties such as MW2, MW8, MW10, and MW13, and it is beneficial to consider uncertain information in such issues. In addition, MW10 is discontinuous. EIM-PoF and MultiObjectiveEGO use infill criterion to multiple scalar objectives into a single objective, resulting in a loss of diversity in multimodal problems. However, USeMOC does not consider the uncertain information of the constraint function. In contrast, HSMEA does not view the uncertain information of objective and constraint, resulting in model errors that mislead the search of the MOEA. EIC-MSSAEA can achieve good performance due to its consideration of the uncertainty of the surrogate models, effectively balancing exploration and exploitation. For MW5-7, MW11, and MW14, their CPFs are discontinuous, and the distribution of feasible regions is irregular. The poor performance of USeMOC, EIM-PoF, and MultiObjectiveEGO on these problems is mainly due to their preference for feasible solutions and neglect of infeasible information during optimization. On MW11, KTS outperforms all, with our algorithm coming in second. On MW14, HSMEA outperforms all algorithms, and our algorithm ranks the third. MW3, MW9, and MW12 specifically evaluate the capability of the algorithm to handle constraints. In this regard, $Stage3$ of EIC-MSSAEA demonstrates effective constraint handling, resulting in satisfactory solutions.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

TABLE III
THE IGD VALUES OBTAINED BY ASA-MOEA/D, MGSAEA, KTS, USEMOC, HSMEA, EIM-POF, MULTIOBJECTIVEEGO AND EIC-MSSAEA ON LIRCMOP AND MW TEST SUITES.

Problem	ASA-MOEA/D	MGSAEA	KTS	USeMOC	HSMEA	EIM-PoF	MultiObjectiveEGO	EIC-MSSAEA
LIRCMOP1	NaN (NaN)	4.5757e-1 (1.23e-1) -	4.0027e-1 (1.77e-1) -	NaN (NaN)	3.9657e-1 (7.02e-2) -	4.3076e-1 (1.07e-1) -	4.8653e-1 (1.00e-1) =	**2.2246e-1 (1.66e-1)**
LIRCMOP2	NaN (NaN)	3.8724e-1 (0.00e+0) =	2.7290e-1 (1.01e-1) =	3.2879e-1 (0.00e+0) =	3.1583e-1 (3.51e-2) -	3.4404e-1 (2.46e-2) -	4.5513e-1 (0.00e+0) =	**1.9448e-1 (1.10e-1)**
LIRCMOP3	NaN (NaN)	NaN (NaN)	3.2957e-1 (1.45e-1) =	3.5536e-1 (0.00e+0) =	3.8087e-1 (8.97e-2) -	3.7623e-1 (3.25e-2) =	NaN (NaN)	**2.4271e-1 (1.49e-1)**
LIRCMOP4	3.2620e-1 (0.00e+0) -	NaN (NaN)	2.7343e-1 (9.73e-2) =	NaN (NaN)	3.7414e-1 (6.39e-2) -	3.3965e-1 (3.34e-2) =	NaN (NaN)	**2.2224e-1 (1.60e-1)**
LIRCMOP5	1.3030e+0 (3.22e-1) -	**3.9938e-2 (1.15e-2) +**	4.7423e-2 (2.20e-2) +	6.3675e-1 (6.40e-1) -	7.6541e-2 (4.69e-2) =	1.3306e+0 (3.43e-1) -	1.3089e+0 (4.00e-2) -	5.5751e-2 (1.33e-2)
LIRCMOP6	1.3880e+0 (3.90e-1) -	**4.0416e-2 (1.02e-2) +**	5.6467e-2 (2.57e-2) =	5.1759e-1 (4.19e-1) -	4.3933e-2 (3.67e-2) +	1.0923e+0 (6.13e-1) -	1.3943e+0 (2.16e-2) -	5.0099e-2 (1.18e-2)
LIRCMOP7	1.2700e+0 (8.60e-1) -	4.6173e-1 (2.67e-1) -	4.6406e-1 (1.91e-1) -	4.0909e-1 (2.70e-1) -	6.2367e-1 (3.79e-1) -	1.1435e+0 (7.95e-1) -	1.0307e+0 (6.20e-1) -	**1.0509e-1 (3.45e-2)**
LIRCMOP8	1.4760e+0 (7.16e-1) -	4.8312e-1 (2.65e-1) -	3.3166e-1 (8.68e-2) -	4.1385e-1 (3.74e-1) -	3.9901e-1 (1.93e-1) -	1.3330e+0 (6.74e-1) -	1.3009e+0 (5.38e-1) -	**9.2308e-2 (5.13e-2)**
LIRCMOP9	9.6974e-1 (2.81e-1) -	6.6339e-1 (1.34e-1) -	5.0520e-1 (2.15e-1) -	1.0972e+0 (4.66e-1) -	**1.4502e-1 (1.06e-1) =**	1.7011e+0 (3.69e-1) -	8.4666e-1 (2.10e-1) -	2.0538e-1 (9.25e-2)
LIRCMOP10	7.9885e-1 (1.55e-1) -	5.4804e-1 (1.48e-1) -	1.7984e-1 (8.58e-2) -	9.9588e-1 (3.40e-1) -	8.1012e-2 (9.36e-2) =	1.2747e+0 (2.30e-1) -	1.0653e+0 (1.21e-1) -	**5.1161e-2 (4.64e-2)**
LIRCMOP11	8.1797e-1 (2.04e-1) -	5.4073e-1 (1.28e-1) -	3.2976e-1 (1.70e-1) -	9.0570e-1 (3.13e-1) -	3.7965e-1 (2.16e-1) -	1.5330e+0 (3.96e-1) -	9.5341e-1 (1.38e-1) -	**2.0073e-1 (1.28e-1)**
LIRCMOP12	9.7677e-1 (2.70e-1) -	7.1582e-1 (2.12e-1) -	5.4349e-1 (2.53e-1) -	1.4240e+0 (3.88e-1) -	4.0744e-1 (1.39e-1) -	1.8897e+0 (5.19e-1) -	8.1691e-1 (2.95e-1) -	**2.0907e-1 (5.01e-2)**
LIRCMOP13	1.9616e+0 (3.48e-1) -	1.3229e+0 (2.70e-1) -	1.6080e-1 (3.00e-2) -	2.8065e+0 (2.75e-1) -	3.8411e-1 (2.70e-1) -	2.2426e+0 (2.94e-1) -	7.6635e-1 (1.52e-1) -	**9.0673e-2 (7.55e-3)**
LIRCMOP14	2.0057e+0 (2.98e-1) -	1.3351e+0 (2.21e-1) -	3.1040e-1 (6.87e-2) -	2.6523e+0 (4.55e-1) -	5.3516e-1 (1.30e-1) -	2.3073e+0 (4.55e-1) -	7.7682e-1 (1.91e-1) -	**1.1661e-1 (1.00e-2)**
MW1	NaN (NaN)	6.7555e-1 (8.70e-2) -	NaN (NaN)	NaN (NaN)	5.7801e-1 (1.57e-1) -	NaN (NaN)	NaN (NaN)	**1.7636e-1 (2.11e-1)**
MW2	6.6810e-1 (2.17e-1) -	4.3864e-1 (3.67e-1) =	3.3074e-1 (2.62e-1) =	6.4511e-1 (4.15e-2) -	4.7031e-1 (1.93e-1) -	2.3819e-1 (1.64e-1) =	5.4356e-1 (1.09e-1) -	**1.8489e-1 (9.99e-2)**
MW3	1.4467e-1 (6.30e-2) -	5.9137e-2 (1.30e-2) -	2.6798e-2 (8.93e-3) -	5.9895e-1 (3.09e-1) -	4.0727e-2 (7.60e-3) -	5.3755e-2 (6.87e-3) -	NaN (NaN)	**2.0739e-2 (3.66e-3)**
MW4	NaN (NaN)	3.2182e-1 (1.83e-1) -	4.9482e-1 (2.33e-1) -	NaN (NaN)	6.9278e-1 (3.01e-1) -	NaN (NaN)	NaN (NaN)	**1.9307e-1 (1.09e-1)**
MW5	NaN (NaN)	6.7029e-1 (1.66e-1) -	5.8592e-1 (2.63e-1) -	NaN (NaN)	4.7659e-1 (3.11e-1) =	NaN (NaN)	NaN (NaN)	**3.0920e-1 (2.37e-1)**
MW6	9.5688e-1 (5.02e-3)-	8.5830e-1 (2.58e-1) =	8.2329e-1 (3.81e-1) =	8.6770e-1 (0.00e+0) =	8.1916e-1 (2.64e-1) =	8.8631e-1 (1.85e-1) =	NaN (NaN)	**7.2755e-1 (2.71e-1)**
MW7	1.1540e-1 (3.34e-2) -	7.9569e-2 (1.79e-2) -	**3.8148e-2 (8.70e-3) =**	6.4827e-1 (2.14e-1) -	1.5235e-1 (1.97e-1) -	8.9650e-2 (2.99e-2) -	NaN (NaN)	3.8656e-2 (1.65e-2)
MW8	8.7162e-1 (1.89e-1) -	5.3468e-1 (2.58e-1) -	4.9414e-1 (1.55e-1) -	1.0726e+0 (1.76e-2) -	5.7640e-1 (2.37e-1) -	6.2338e-1 (1.99e-1) -	NaN (NaN)	**2.1587e-1 (9.43e-2)**
MW9	NaN (NaN)	5.5454e-1 (3.40e-1) -	4.5913e-1 (2.16e-1) -	NaN (NaN)	6.9244e-1 (3.71e-1) -	NaN (NaN)	NaN (NaN)	**2.3107e-1 (2.20e-1)**
MW10	NaN (NaN)	4.6580e-1 (1.85e-1) =	NaN (NaN)	NaN (NaN)	NaN (NaN)	NaN (NaN)	NaN (NaN)	**3.2735e-1 (2.35e-1)**
MW11	7.3635e-1 (2.09e-1) -	2.5967e-1 (2.54e-1) =	**1.1501e-1 (2.19e-2) +**	9.7180e-1 (1.87e-1) -	8.3716e-1 (3.39e-1) -	5.8998e-1 (2.93e-1) -	NaN (NaN)	1.5771e-1 (7.21e-2)
MW12	8.8654e-1 (0.00e+0) -	8.6396e-1 (2.59e-1) -	7.2331e-1 (3.99e-1) -	NaN (NaN)	7.9611e-1 (3.00e-1) -	NaN (NaN)	NaN (NaN)	**2.2863e-1 (2.55e-1)**
MW13	4.4604e+0 (1.80e+0) -	1.6261e+0 (9.97e-1) -	1.5954e+0 (7.26e-1) -	5.4946e+0 (1.47e+0) -	3.1517e+0 (2.02e+0) -	4.1921e+0 (1.86e+0) -	4.9876e+0 (1.97e+0) -	**7.1978e-1 (5.35e-1)**
MW14	8.7063e-1 (4.11e-1) -	2.5561e+1 (1.29e+1) -	5.1675e-1 (5.04e-1) +	2.6162e+0 (3.01e-1) -	**4.2859e-1 (6.91e-2) +**	7.6709e-1 (2.24e-1) =	3.0452e+0 (3.88e-1) -	7.4283e-1 (3.82-1)
+/ − / =	0/28/0	2/21/5	3/18/7	0/23/5	2/24/2	0/23/5	0/26/2	

'NaN' represents no feasible solutions. The gray background represents the best result for each test instance.

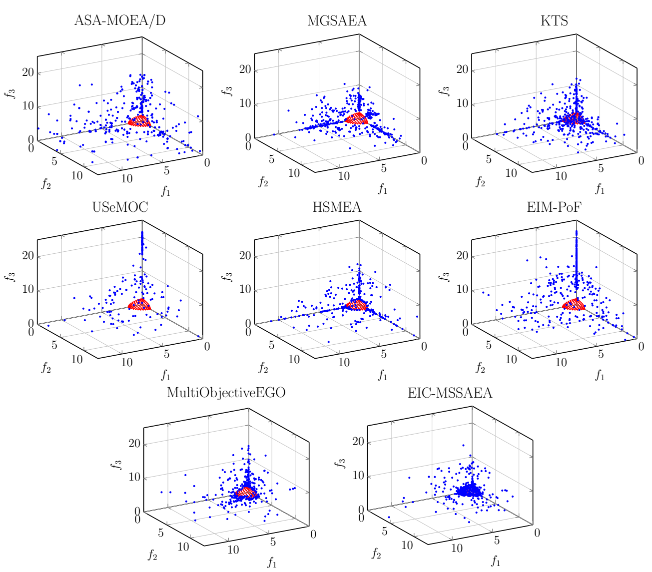


Fig. 6. In 20 experiments, the distribution of the solutions corresponding to the minimum value of IGD obtained by ASA-MOEA/D, MGSAEA, KTS, USeMOC, HSMEA, EIM-PoF, MultiObjectiveEGO, and EIC-MSSAEA on LIRCMOP13.

Fig. S5 of the Supplementary material depicts the distribution of the solutions obtained by all algorithms on MW13, based on the minimum IGD. It is noteworthy that EIC-MSSAEA demonstrate many solutions positioned on the CPF. This observation serves to reinforce the effectiveness and superiority of our approach.

3) Results of C-DTLZ test suite: The C-DTLZ test suite retains the characteristics of the DTLZ while allowing for increased objective functions. The prefix of C-DTLZ, ”C”, indicates the type of constraint, such as ”C2”, which signifies that CPF is a part of UPF. The suffix of C-DTLZ corresponds to the characteristics of the DTLZ test suite, as detailed in Section IV-F. We compare EIC-MSSAEA against other algorithms using C-DTLZ test cases with 3 and 5 objectives. The IGD results are presented in Table SXI of the Supplemental material. The experimental results indicate that our algorithm outperforms ASA-MOEA/D, MGSAEA, KTS, USeMOC, HSMEA, EIM-PoF, and MultiObjectiveEGO in 13, 13, 9, 14, 5, 12, and 11 out of the 14 test cases, respectively. Our algorithm has the best overall performance. Fig. S6 of the Supplementary material shows the distribution of solutions obtained by all algorithms on C2-DTLZ2 with 3 objectives, based on the minimum IGD.

4) Results of the practical application: We validate the effectiveness of the algorithm on optimization of operating parameters of crude distillation units. Fourteen operational parameters need to be optimized for this problem, and two optimization objectives: oil product revenue and energy consumption cost. Additionally, 8 product quality constraints need to be met. The simulation model of the crude oil distillation unit is constructed using ASPEN HYSYS simulation software. The average HV results of 20 experiments are shown in Table SXIII of the Supplementary material, and the experimental results show that our algorithm has achieved competitive results. A more detailed description of the refining problem and further experimental results can be found in Section III-E of the Supplementary material.

5) Comparative results with CMOEAs: Table SXIV shows the IGD results obtained by EIC-MSSAEA and three CMOEAs, each with multiple stages, on the LIRCMOP, MW, and C-DTLZ benchmark suites. The results show that MCCMO, C3M, and MSCMO perform worse than

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

EIC-MSSAEA on all test cases. The main reason is that they cannot converge to the vicinity of CPF with only 400 FEs without using the surrogate models.

In our experiments, the maximum number of constraints is eight, and the proposed algorithm performs consistent well for different numbers of constraints. In principle, the proposed algorithm can solve ECMOPs with an arbitrary number of constraints. A more detailed discussion is provided in Section III-F of the Supplementary material.

F. Comparisons with Peers for Expensive Multi-Objective Optimization Problems

To assess the performance of $Stage1$, EIC-S1, within EIC-MSSAEA, we conduct a comparative analysis with state-of-the-art algorithms that utilize infill criteria, including ABSAEA, EIMEGO, and NSGA-III-EHVI. The experimental results for three- and five-objective test problems are provided in Table SXV and Table SXVI of the Supplementary material.

1) Results of DTLZ test suite: As shown in Table SXV, EIC-S1 outperforms ABSAEA, EIMEGO, and NSGA-III-EHVI in 6, 6, and 6 out of 7 cases, respectively, involving three objectives. Specifically, DTLZ1 and DTLZ3 exhibit multimodality, which poses challenges for achieving convergence. EIC-S1 excels in DTLZ1 and DTLZ3 due to the utilization of non-dominated sorting within EIC, emphasizing convergence as a priority. In contrast, DTLZ2 assesses algorithm diversity performance, where EIC-S1 outperforms other approaches due to multiple members within EIC accounting for diversity consideration. Additionally, DTLZ5 and DTLZ7 entail irregular Pareto fronts, necessitating strong diversity maintenance by the algorithm. Notably, EIMEGO achieves the best performance in cases involving Pareto front degradation. Furthermore, as indicated in Table SXVI of the Supplemental material, EIC-S1 surpasses ABSAEA, EIMEGO, and NSGA-III-EHVI in 3, 4, and 3 out of 7 cases, respectively, with five objectives. This further underscores the commendable performance of our algorithm in addressing these challenges.

2) Results of WFG test suite: From Table SXV, EIC-S1 outperforms ABSAEA, EIMEGO, and NSGA-III-EHVI in 5, 6, and 4 out of 9 cases, respectively. Specifically, EIC-S1 achieves the best performance on WFG2, WFG3, and WFG7, where WFG2 and WFG3 have discontinuous Pareto fronts, and WFG7 exhibits separability. In the case of WFG5, a deceptive problem, EIMEGO achieves the best results. On the other hand, NSGA-III-EHVI attains the best results on the inseparable problem of WFG6. Furthermore, as shown in Table SXVI of the Supplemental material, EIC-S1 surpasses ABSAEA, EIMEGO, and NSGA-III-EHVI in 6, 5, and 3 out of 9 cases with five objectives. The above statistical results further emphasize the remarkable performance of EIC-S1 in effectively scaling up optimization for many objectives.

V. Conclusions

This paper proposes an EIC-MSSAEA for addressing ECMOPs. The algorithm integrates the GP-assisted multi-stage optimization and ensemble infill criterion to improve the performance while maintaining computational efficiency. The multi-stage optimization process aims to quickly approximate the CPF by overcoming infeasible obstacles for problems with a discontinuous CPF and a small feasible region. In each stage, the ensemble infill criterion selects the best solution for evaluation based on feasibility, convergence, diversity, exploration, and exploitation. We evaluate its performance on constrained benchmark problems (e.g., LIRCMOP, MW, C-DTLZ) and a practical application problem, as well as unconstrained benchmark problems (e.g., DTLZ, WFG). EIC-MSSAEA is shown to be competitive compared with state-of-the-art algorithms, highlighting its effectiveness in handling complex ECMOPs and EMOPs.

However, it should be noted that in the optimization process assisted by GP, we did not consider the potential impact of model errors. Our approach solely relies on the predicted objective values to guide the search. To address this problem, we plan to incorporate the uncertainty information the GP provides to enhance the optimization process in our future work. Another important consideration is the dimensionality of the decision space. High-dimensional optimization problems pose challenges due to the computational complexity of the GP and the requirement to model each objective and constraint individually. Consequently, EIC-MSSAEA may not be suited for solving high-dimensional ECMOPs. To address this limitation, we plan to explore alternative machine-learning models that can serve as more effective surrogate models.

References

[1] A. R. Yildiz, H. Abderazek, and S. Mirjalili, “A comparative study of recent non-traditional methods for mechanical design optimization,” *Arch. Comput. Methods Eng.*, vol. 27, pp. 1031–1048, 2020.

[2] G. P. Rangaiah, *Multi-objective optimization: techniques and applications in chemical engineering*. World Scientific, 2016, vol. 5.

[3] R. G. Regis, “Stochastic radial basis function algorithms for large-scale optimization involving expensive black-box objective and constraint functions,” *Comput. Oper. Res.*, vol. 38, no. 5, pp. 837–853, May. 2011.

[4] L. V. Santana-Quintero, A. A. Montano, and C. A. C. Coello, “A review of techniques for handling expensive functions in evolutionary multi-objective optimization,” *Computational intelligence in expensive optimization problems*, pp. 29–59, 2010.

[5] K. Deb, P. C. Roy, and R. Hussein, “Surrogate modeling approaches for multiobjective optimization: methods, taxonomy, and results,” *Math. Comput. Appl.*, vol. 26, no. 1, p. 5, 2020.

[6] J. Liang, X. Ban, K. Yu, B. Qu, K. Qiao, C. Yue, K. Chen, and K. C. Tan, “A survey on evolutionary constrained multiobjective optimization,” *IEEE Trans. Evol. Comput.*, vol. 27, no. 2, pp. 201–221, Mar. 2022.

[7] J. Zou, R. Sun, Y. Liu, Y. Hu, S. Yang, J. Zheng, and K. Li, “A multi-population evolutionary algorithm using new cooperative mechanism for solving multi-objective problems with multi-constraint,” *IEEE Trans. Evol. Comput.*, Mar. 2023.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

[9] R. Sun, J. Zou, Y. Liu, S. Yang, and J. Zheng, "A multistage algorithm for solving multiobjective optimization problems with multiconstraints," *IEEE Trans. Evol. Comput.*, vol. 27, no. 5, pp. 1207–1219, Nov. 2022.

[10] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, "A surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 3, no. 2, pp. 106–116, Oct. 2018.

[11] Y. Jin, H. Wang, T. Chugh, D. Guo, and K. Miettinen, "Data-driven evolutionary optimization: An overview and case studies," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 442–458, Jun. 2018.

[12] Y. Jin, H. Wang, and C. Sun, *Data-driven evolutionary optimization*. Springer, 2021.

[13] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Comput.*, vol. 9, no. 1, pp. 3–12, May. 2005.

[14] K. Li and R. Chen, "Batched data-driven evolutionary multiobjective optimization based on manifold interpolation," *IEEE Trans. Evol. Comput.*, vol. 27, no. 1, pp. 126–140, Mar. 2022.

[15] M. Seeger, "Gaussian processes for machine learning," *Int. J. Neural Syst.*, vol. 14, no. 02, pp. 69–106, 2004.

[16] G. Li, Q. Zhang, Q. Lin, and W. Gao, "A three-level radial basis function method for expensive optimization," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 5720–5731, Mar. 2021.

[17] Y. Liu, J. Liu, Y. Jin, F. Li, and T. Zheng, "A surrogate-assisted two-stage differential evolution for expensive constrained optimization," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 715–730, 2023.

[18] J. Luo, L. Chen, X. Li, and Q. Zhang, "Novel multitask conditional neural-network surrogate models for expensive optimization," *IEEE Trans. Cybern.*, vol. 52, no. 5, pp. 3984–3997, Sep. 2020.

[19] T. Chugh, Y. Jin, K. Miettinen, J. Hakanen, and K. Sindhya, "A surrogate-assisted reference vector guided evolutionary algorithm for computationally expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 129–142, Oct. 2016.

[20] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018.

[21] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. De Freitas, "Taking the human out of the loop: A review of bayesian optimization," *Proc. IEEE*, vol. 104, no. 1, pp. 148–175, 2015.

[22] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, no. 4, p. 455, Dec. 1998.

[23] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm Evol. Comput.*, vol. 1, no. 2, pp. 61–70, Feb. 2011.

[24] D. Guo, Y. Jin, J. Ding, and T. Chai, "Heterogeneous ensemble-based infill criterion for evolutionary multiobjective optimization of expensive problems," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 1012–1025, May. 2018.

[25] X. Wu, Q. Lin, J. Li, K. C. Tan, and V. C. Leung, "An ensemble surrogate-based coevolutionary algorithm for solving large-scale expensive optimization problems," *IEEE Trans. Cybern.*, vol. 53, no. 9, Sep. 2022.

[26] H. Ma, H. Wei, Y. Tian, R. Cheng, and X. Zhang, "A multi-stage evolutionary algorithm for multi-objective optimization with complex constraints," *Inf. Sci.*, vol. 560, pp. 68–91, Jun. 2021.

[27] M. Sohst, F. Afonso, and A. Suleman, "Surrogate-based optimization based on the probability of feasibility," *Struct. Multidiscip. Optim.*, vol. 65, no. 1, p. 10, 2022.

[28] R. Hussein and K. Deb, "A generative kriging surrogate model for constrained and unconstrained multi-objective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, 2016, pp. 573–580.

[29] D. Zhan, Y. Cheng, and J. Liu, "Expected improvement matrix-based infill criteria for expensive multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 6, pp. 956–975, Apr. 2017.

[30] P. Singh, I. Couckuyt, F. Ferranti, and T. Dhaene, "A constrained multi-objective surrogate-based optimization algorithm," in *2014 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2014, pp. 3080–3087.

[31] R. de Winter, B. van Stein, and T. Bäck, "Samo-cobra: A fast surrogate assisted constrained multi-objective optimization algorithm," in *Evolutionary Multi-Criterion Optimization: 11th International Conference, EMO 2021, Shenzhen, China, March 28–31, 2021, Proceedings 11*. Springer, 2021, pp. 270–282.

[32] G. Sun, L. Li, J. Fang, and Q. Li, "On lower confidence bound improvement matrix-based approaches for multiobjective bayesian optimization and its applications to thin-walled structures," *Thin-Walled Structures*, vol. 161, p. 107248, 2021.

[33] S. Belakaria, A. Deshwal, and J. R. Doppa, "Max-value entropy search for multi-objective bayesian optimization with constraints," *arXiv preprint arXiv:2009.01721*, 2020.

[34] E. C. Garrido-Merchán and D. Hernández-Lobato, "Parallel predictive entropy search for multi-objective bayesian optimization with constraints," *arXiv preprint arXiv:2004.00601*, 2020.

[35] S. Belakaria, A. Deshwal, and J. R. Doppa, "Uncertainty aware search framework for multi-objective bayesian optimization with constraints," *arXiv preprint arXiv:2008.07029*, 2020.

[36] Y. Pang, Y. Wang, S. Zhang, X. Lai, W. Sun, and X. Song, "An expensive many-objective optimization algorithm based on efficient expected hypervolume improvement," *IEEE Trans. Evol. Comput.*, Dec. 2022.

[37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, 2013.

[38] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "An adaptive bayesian approach to surrogate-assisted evolutionary multi-objective optimization," *Inf. Sci.*, vol. 519, pp. 317–331, Mar. 2020.

[39] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.

[40] Z. Song, H. Wang, C. He, and Y. Jin, "A kriging-assisted two-archive evolutionary algorithm for expensive many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1013–1027, Apr. 2021.

[41] Z. Yang, H. Qiu, L. Gao, L. Chen, and J. Liu, "Surrogate-assisted moea/d for expensive constrained multi-objective optimization," *Inf. Sci.*, p. 119016, Aug. 2023.

[42] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[43] Z. Song, H. Wang, B. Xue, M. Zhang, and Y. Jin, "Balancing objective optimization and constraint satisfaction in expensive constrained evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, Jul. 2023.

[44] Y. Zhang, H. Jiang, Y. Tian, H. Ma, and X. Zhang, "Multigranularity surrogate modeling for evolutionary multiobjective optimization with expensive constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 3, pp. 2956–2968, Aug. 2023.

[45] E. Zitzler, M. Laumanns, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," *TIK report*, vol. 103, 2001.

[46] C. Durantin, J. Marzat, and M. Balesdent, "Analysis of multi-objective kriging-based methods for constrained global optimization," *Comput. Optim. Appl.*, vol. 63, pp. 903–926, 2016.

[47] S. Zhang, F. Yang, C. Yan, D. Zhou, and X. Zeng, "An efficient batch-constrained bayesian optimization approach for analog circuit synthesis via multiobjective acquisition ensemble," *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.*, vol. 41, no. 1, pp. 1–14, Jan. 2021.

[48] Y. Li, J. Shen, Z. Cai, Y. Wu, and S. Wang, "A kriging-assisted multi-objective constrained global optimization method for expensive black-box functions," *Mathematics*, vol. 9, no. 2, p. 149, Jan. 2021.

[49] A. Habib, H. K. Singh, T. Chugh, T. Ray, and K. Miettinen, "A multiple surrogate assisted decomposition-based evolutionary algorithm for expensive multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 1000–1014, Feb. 2019.

[50] J. C. Helton and F. J. Davis, "Latin hypercube sampling and the propagation of uncertainty in analyses of complex systems," *Reliab. Eng. Syst. Saf.*, vol. 81, no. 1, pp. 23–69, 2003.

[51] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Sep. 2013.

[52] Z.-Z. Liu and Y. Wang, "Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 870–884, Jan. 2019.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Evolutionary Computation. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2024.3400832

[53] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm Evol. Comput.*, vol. 44, pp. 665–679, Feb. 2019.

[54] H. Ge, M. Zhao, L. Sun, Z. Wang, G. Tan, Q. Zhang, and C. P. Chen, "A many-objective evolutionary algorithm with two interacting processes: Cascade clustering and reference point incremental learning," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 572–586, Oct. 2018.

[55] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in moea/d for cmops with large infeasible regions," *Soft Comput.*, vol. 23, pp. 12 491–12 510, Feb. 2019.

[56] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Trans. Evol. Comput.*, vol. 23, no. 6, pp. 972–986, Feb. 2019.

[57] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.

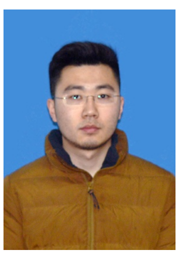
[58] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Comput. Intell. Mag.*, vol. 12, no. 4, pp. 73–87, Oct. 2017.

[59] L. Ma, M. Huang, S. Yang, R. Wang, and X. Wang, "An adaptive localized decision variable analysis approach to large-scale multiobjective and many-objective optimization," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6684–6696, Jan. 2021.

[60] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

[61] D. A. Van Veldhuizen, *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. Air Force Institute of Technology, 1999.

[62] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Trans. Cybernet.*, vol. 47, no. 6, pp. 1510–1522, May. 2016.



Haofeng Wu recived the M.S. degree in control engineering from the Shenyang University of Technology, Shenyang, in 2017. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, China.

His current research interests include surrogate-assisted evolutionary optimization and its applications in industrial processes.

Qingda Chen (Member, IEEE) received the B.S. degree from Yantai University, Yantai, China, in 2013, and the Ph.D. degree in control theory and control engineering from the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University, Shenyang, China, in Apr. 2020.

He is currently a Lecturer with the State Key Laboratory of Synthetical Automation for Process Industry, Northeastern University, Shenyang, China. His current research interests include computational intelligence and its application in industrial processes.

Jiaxin Chen received an M.S. degree in control theory and control engineering from the State Key Laboratory of Synthetical Automation for Process Industries at Northeastern University, Shenyang, China, in 2019. She is currently pursuing a Ph.D. degree in control science and engineering at the same institution.

Her current research interests include multi-objective bilevel optimization, hyper-parameter optimization, and automated machine learning.

Yaochu Jin (Fellow, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996, respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Germany, in 2001. He joined the Westlake University, Hangzhou, China in October 2023 as a Chair Professor of AI, leading the Trustworthy and General AI Lab. His main research interests include multi-objective and data-driven evolutionary optimization, evolutionary multi-objective learning, trustworthy AI, and evolutionary developmental AI.

Prof. Jin is presently the President-Elect of the IEEE Computational Intelligence Society and the Editor-in-Chief of Complex & Intelligent Systems. He is the recipient of the 2018, 2021 and 2023 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, and the 2015, 2017, and 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award. He was named by the Clarivate as "a Highly Cited Researcher" from 2019 to 2022 consecutively. He is a Member of Academia Europaea.

Jinliang Ding (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in control theory and control engineering from Northeastern University, Shenyang, China, in 2001, 2004 and 2012, respectively.

He is currently a Professor with the State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University. He has authored or coauthored over 200 refereed journal and international conference papers, and has invented or coinvented more than 50 patents. His research interests include modeling, plant-wide control and optimization for the complex industrial systems, machine learning, industrial artificial intelligence, computational intelligence, and its application.

Dr. Ding has received numerous awards, including the Young Scholars Science and Technology Award of China (2016), the National Science Fund for Distinguished Young Scholars (2015), the National Technological Invention Award (2013). He also serves as an Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II: EXPRESS BRIEFS.

Xingyi Zhang (Senior Member, IEEE) received the B.Sc. degree from Fuyang Normal College, Fuyang, China, in 2003, and the M.Sc. and Ph.D. degrees from Huazhong University of Science and Technology, Wuhan, China, in 2006 and 2009, respectively.

He is currently a Professor with the School of Computer Science and Technology, Anhui University, Hefei, China. His current research interests include unconventional models and algorithms of computation, multi-objective optimization, and membrane computing. He is the recipient of the 2018, 2021, and 2024 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, and the 2020 IEEE Computational Intelligence Magazine Outstanding Paper Award.

Tianyou Chai (Life Fellow, IEEE) received the Ph.D.degree in control theory and engineering from Northeastern University, Shenyang, China, in 1985.

He became a Professor with Northeastern University in 1988 and a Chair Professor in 2004. His current research interests include adaptive control, intelligent decoupling control, integrated plant control and systems, and the development of control technologies with applications to various industrial processes.

Prof. Chai is a member of the Chinese Academy of Engineering, an Academician of the International Eurasian Academy of Sciences, and an IFAC Fellow.

Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on April 06,2025 at 14:05:56 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.