

An Efficient Approach for Solving Expensive Constrained Multiobjective Optimization Problems

Kamrul Hasan Rahi, *Graduate Student Member, IEEE*

Abstract—To solve real-world expensive constrained multi-objective optimization problems (ECMOPs), surrogate/approximation models are commonly incorporated in evolutionary algorithms to pre-select promising candidate solutions for evaluation. However, the performance of existing approaches are highly dependent on the relative position of unconstrained and constrained Pareto fronts (UPF and CPF, respectively). In addition, the uncertainty information of surrogate models is often ignored, which can misguide the search. To mitigate these key issues (among others), an efficient probabilistic selection based constrained multi-objective EA is proposed, referred to as PSCMOEA. It comprises novel elements such as (a) an adaptive search bound identification scheme based on the feasibility and convergence status of evaluated solutions (b) a probabilistic selection method backed by theoretical formulations of model mean and uncertainties to conduct search in the predicted space to identify promising solutions (c) an efficient single infill sampling approach to balance feasibility, convergence and diversity across different stages of the search and (d) an adaptive switch to unconstrained search based on certain search conditions. Numerical experiments are conducted on an extensive range of challenging constrained problems using low evaluation budgets to simulate ECMOPs. The performance of PSCMOEA is benchmarked against five competitive state-of-the-art algorithms, to demonstrate its competitive and consistent performance.

Index Terms—Surrogate assisted optimization, multi-objective algorithm, infill strategy, expensive constrained optimization.

I. INTRODUCTION

DECISION making in real-world optimization problems often requires trade-off considerations among conflicting objectives subject to satisfying various practical constraint(s). Formally this constitutes a constrained multi-objective optimization problem (CMOP) [1], represented as shown in Eq. 1.

$$\begin{aligned} \underset{\mathbf{x}}{\text{Minimize:}} \quad & \mathbf{F}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})\} \\ \text{subject to} \quad & \begin{cases} \mathbf{x} \in \Omega \\ g_j(\mathbf{x}) \leq 0, \quad j = 1, \dots, p \end{cases} \end{aligned} \quad (1)$$

Here, $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is an n -dimensional decision vector defining the decision/search space $\Omega = [x_i^L, x_i^U]^n \subseteq \mathbf{R}^n$ for real-valued continuous variables. The M objective functions are $f_1(\mathbf{x}), \dots, f_M(\mathbf{x})$, subject to p inequality constraints $g_{i=1:p}(\mathbf{x}) \leq 0$. Equality constraints, if present, are generally converted to inequality constraints by applying relaxation techniques [1], hence they are omitted for brevity. The constraint violation (CV) of a solution is computed as $CV(\mathbf{x}) = \sum_{j=1}^p \max(0, g_j(\mathbf{x}))$, which is 0 for any feasible solution (which satisfies all constraints), and a positive

quantity for an infeasible solution (which violates at least one constraint).

Between two feasible solutions \mathbf{x}_1 and \mathbf{x}_2 , solution \mathbf{x}_1 dominates \mathbf{x}_2 if $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2)$ for each $i \in \{1, \dots, M\}$ and $f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$ for at least one $j \in \{1, \dots, M\}$, which is denoted as $\mathbf{x}_1 \prec \mathbf{x}_2$. A feasible solution \mathbf{x}^* is said to be a Pareto optimal if no other feasible solution \mathbf{x} dominates \mathbf{x}^* . The set of all feasible Pareto optimal solutions are represented by a trade-off set known as constrained Pareto set (CPS), while its image in the objective space is termed as constrained Pareto front (CPF). Likewise, the Pareto set/front obtained by disregarding the constraints is referred to as unconstrained PS/PF (UPS/UPF)¹. Solving CMOPs is typically acknowledged to be more challenging than solving unconstrained MOPs (UMOPs) due to the need to account for feasibility in addition to convergence and diversity of the solutions. This challenge is further aggravated for expensive problems (ECMOPs), wherein evaluation of each candidate design requires significant expense, for example, time-consuming numerical simulations or physical experiments [2]. Evidently, there are stringent limitations on the number of calls to the true (expensive) evaluation that can be practically afforded when solving real-world problems of this nature. As an example, consider the Ford crash simulation discussed in [3], requiring 36-160 hours to assess a single design. Extrapolating from this, conducting 100 design evaluations (sequentially) would require approx 5 months to 1.8 years.

Multi-objective Evolutionary Algorithms (MOEAs) are a common choice to solve MOPs. This is primarily attributed to their ability to deal with highly non-linear/black-box functions and to evolve a set of solutions ('population') that naturally yields itself to a discrete approximation of the PF. Even so, they require evaluation of a large number of candidate solutions prior to convergence, and hence not suitable for solving EMOPs in their native form. To circumvent this challenge, surrogate-assisted MOEAs (SA-MOEAs) are often used, wherein approximations of the expensive functions are built to guide the search for the most part, with expensive evaluation called only sparingly for promising candidates. While there exist many different types of surrogate models, such as polynomial response surface modeling (RSM), radial basis function (RBF), support vector regression (SVR), one particular type of model, i.e, Kriging (Gaussian Process), has gained significant attention [4]. This is mainly attributed to the fact that unlike other models, which typically only provide a mean estimates of the response function, Kriging can also

¹Note that feasibility is a necessary condition for optimality, hence generally CPS/CPF are simply referred to as PS/PF. The use of the terms CPS/CPF here is to explicitly distinguish them from UPS/UPF.

provide uncertainty bounds, which can be used for a more informed selection of infill solutions [4]–[6]. While SA-MOEAAs have demonstrated significant progress in addressing EMOPs, majority of the studies focus on unconstrained EMOPs, as evident from (and highlighted in) a recent survey [7]. Only a limited number of investigations have delved into ECMOPs, although this important problem of practical significance is gaining traction in the research community recently [8]–[13].

Constraint-handling is a key element of solution approaches for solving CMOPs. It refers to strategies incorporated in one or more evolutionary operations, such as ranking, recombination or environmental selection, in order to drive the population towards feasibility and eventually cover the PF. The constraints may introduce several topological features in the search landscape that make it difficult for the search methods to navigate. These include, but are not limited to, presence of very constricted feasible regions, multiple disconnected regions and feasible regions with significant bias (e.g., some parts of a CPF are achieved well before others). These different constraint scenarios affect the relative positions of UPF and CPF and/or induce particular difficulties in achieving certain parts of the CPF, which in turn impacts the performance of the solution methods. A number of constraint handling techniques (CHTs) have been proposed for solving general CMOPs, some of which are also adapted to work for ECMOPs. However, limited evaluation budget constraint induces significant challenge to the algorithms for solving ECMOPs [14]. While prior approaches have demonstrated potential in addressing ECMOPs [1], [6], [15], several unresolved challenges and promising prospects require further attention which motivate this research. Firstly, the algorithm is required to be efficient in navigating potential feasible region(s) followed by identifying well converged and distributed solutions on the CPF within the allocated evaluation budget by ensuring proper balance among feasibility, convergence and diversity. Secondly, the algorithm should be capable of utilizing the uncertainty information of surrogate models to ensure reliable performance comparison of solutions in the predicted landscape in order to avoid any misguided search. Lastly, limited attention has been paid to the development of *steady-state* methods, where only a single candidate solution can be evaluated at a time [5]. There exist scarce approaches that cater to such scenarios, and are generally extensions of efficient global optimization (EGO) techniques, e.g. MultiObjectiveEGO [9]. This is likely because formulating a sampling criterion that can balance feasibility, convergence and diversity concurrently is challenging in its own right. Most existing approaches therefore operate on a generational model (multiple expensive evaluations done in each generation). However, recent studies have shown that steady-state approaches may be more advantageous for expensive UMOPs under limited evaluation budgets [5], especially where the evaluations are non-parallelizable, and it is worth investigating if similar benefits can be realized for ECMOPs.

Towards addressing the above research gaps, the following contributions are made in this study:

- 1) First, illustrative examples are provided to highlight the influence of the relative positions of UPF and CPF on some recent representative search methods. This helps

explain why some existing approaches might struggle, and provides a motivation to address these challenges in the algorithm design.

- 2) A new CMOEA based on *probabilistic selection* is proposed, referred to as PSCMOEA, to deal with ECMOPs. PSCMOEA is a steady-state algorithm, unlike most of its peers. Its mainstay in efficient search is the careful consideration of model uncertainties in its operation along with the status of the solutions in the archive in terms of feasibility, convergence and diversity. These considerations feed into various components, including (a) ranking/environmental selection using a theoretically derived probabilistic constrained dominance (*PCD*); (b) infill criteria that adapts itself based on the status of the evaluated solutions in the archive, catering for scenarios where all solutions are feasible, all are infeasible, or a mix of both.
- 3) PSCMOEA also contains other additional improvisations, such as normalization of the solutions in the presence of a mix of feasible/infeasible solutions, and adaptive switch to unconstrained search to improve convergence rate, triggered based on correlation between search directions of UPF and CPF.
- 4) Numerical experiments are presented on an extensive range of problems with challenging features to demonstrate the competence of PSCMOEA relative to five other state-of-the-art methods.

Following this introduction, Section II provides an overview of related work, highlighting the limitations of existing methods on specific problems to justify the motivation for this study. The proposed approach is detailed in Section III, followed by the numerical experiments in Section IV. Concluding remarks are presented in Section V.

II. RELATED WORK AND MOTIVATION

A. Conventional constraint handling techniques for CMOPs

CMOEAAs have received significant attention in the recent years [1]. Existing CMOEAAs can be roughly classified into four main categories based on their specific constraint handling techniques (CHTs). The first and perhaps the most used category separates the solutions into feasible and infeasible blocks based on the CV information, followed by preferring feasible solutions over infeasible ones [16], [17]. This is referred to as parameterless technique, constraint domination principle or feasibility-first principle. While it is a simple and an intuitive way of handling constraints, it may drive the population towards an easily identifiable portion of the CPF and miss other parts [18]. The second category involves converting a CMOP into an UMOP by adapting constraints as additional objectives or penalty functions [18]–[20]. However, ensuring the UMOP shares the same optimum as the CMOP requires extensive parameter tuning. Besides, addition of new objective(s) may induce further complexities, such as those encountered in *many-objective* optimization. The third and fourth categories can be considered as the representative of the most recent techniques. The third category utilizes multiple populations by setting different goals to optimize objectives

and constraints collaboratively [21]–[23]. This is achieved by evolving dedicated populations for each goal. Typically the primary goal is set to identify feasible region(s) by considering constraints while the secondary goal attempts to expedite convergence without considering constraints (i.e., search for UPF). Subsequently, these populations adopt a co-evolutionary approach by interacting with each other to enhance the effectiveness of evolutionary search and selection. However, the performance primarily depends on the relationship between these two goals, and disparities in optimal solution sets (for UPF vs CPF) may misguide the search. The fourth category divides the evolutionary process into stages favoring specific optimization tasks (e.g., with or without considering constraints) at different stages [24]–[26]. However, appropriate transition of stages and optimal distribution of computational resources are critical considerations that affect the performance of the approach.

B. Constraint handling techniques for ECMOPs

Although some of the past approaches incorporated simple mechanisms to handle constraints for ECMOPs [9], [27], [28], dedicated works to solve more challenging ECMOPs have only recently started emerging. The design of SA-CMOEAs typically involves three key elements: model management, optimization framework, and infill identification approach. Various recent SA-MOEAs (all proposed within last 3 years), such as SAMO-COBRA [29], IC-SA-NSGA-II [30], KTS [11], ASA-MOEA/D [12], and KMGSAEA [10], employ different surrogate models and optimization strategies for evolving candidate solutions. It is also seen that recent SA-CMOEAs adopt strategies inspired by parallel developments in the field of generic CMOEAs. Notably, KTS uses two inter-switchable surrogate-assisted EAs (with and without considering constraints) based on the correlation between the convergence direction and CV minimization direction, motivated by the concept of co-evolutionary approaches as discussed above. ASA-MOEA/D adaptively applies three types of local search (e.g., convergence driven, diversity driven and feasibility driven) to optimize the subpopulations along different reference vectors (RVs) within a decomposition based framework, demonstrating effectiveness of approximating CPFs of different shapes. KMGSAEA introduces a multi-granularity Kriging-assisted framework to mitigate the adverse effects of model errors. Furthermore, another category of SA-CMOEAs is evident in literature which involves constructing and optimizing effective acquisition functions based on the surrogate prediction of response functions and different performance indicators such as expected Hypervolume (HV) [31] improvement and expected improvement (EI) [32]. To address constraints in these approaches, the probability of feasibility (PoF) [33] of solutions is typically multiplied with these acquisition functions to balance objective optimization with constraint satisfaction. In [34], a constraint-oriented criterion is formulated by multiplying PoF with the probability of HV improvement followed by optimizing the criterion to identify the optimal solution in antenna design optimization, while in [35], an R2 indicator-based EI is multiplied with PoF. In [36],

three separate infill criteria is designed by multiplying PoF with computationally efficient EIM (expected improvement matrix), wherein non-dominated objective values are replaced with corresponding EI values. Besides, decomposition-based SA-CMOEAs [37], [38] enable parallel selection of multiple infill points by decomposing ECMOPs into constrained single-objective sub-problems where PoF is multiplied with unary indicators, contributing to improved diversity.

C. Motivations of this study

Some of the challenges in CMOPs can be understood by observing relative locations of UPF and CPF, and the geometry of CPF itself. A few common instances as represented in [39] relevant to this study are schematically shown in Fig. 1. Fig. 1(a) represents a case where CPF and UPF are identical, making it easy for feasibility-first (FF) and/or co-evolutionary based strategies given their higher focus on feasibility. However, for the instances in Fig. 1(b) - 1(c), enhanced diversity during evolutionary search is required since the UPF and CPF share disconnected segments. Emphasis on the FF or unconstrained search might bias the search in these scenarios while, strategies that preserve some infeasible solutions [18] might be more suitable to navigate the disconnected feasible regions, followed by intensification of search close to the CPF. In Fig. 1(d), the CPF and UPF are completely separated by infeasible region(s) between them where balancing convergence and feasibility is a considerable challenge. Therefore, co-evolutionary approaches are less effective since ignoring constraints may waste computing resources by driving the search towards UPF, especially when the gap between UPF and CPF is large. Fig. 1(e) entails extremely relative small size of CPF and feasible regions, which is challenging for all CMOEAs because identification of the feasible region(s) is very difficult. In Fig. 1(f), convergence to the CPF can be restricted by the presence of disconnected feasible regions with local basins of attraction. Conventional CMOEAs may likely get stuck in such local fronts if the approach cannot navigate infeasible and local feasible regions effectively during evolution.

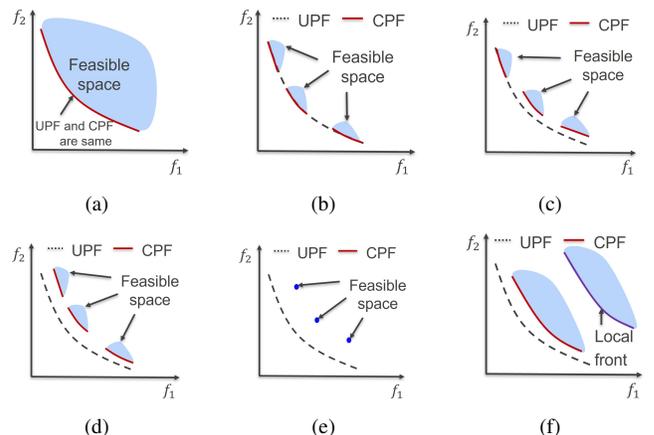


Fig. 1. Different relative positions of UPF and CPF.

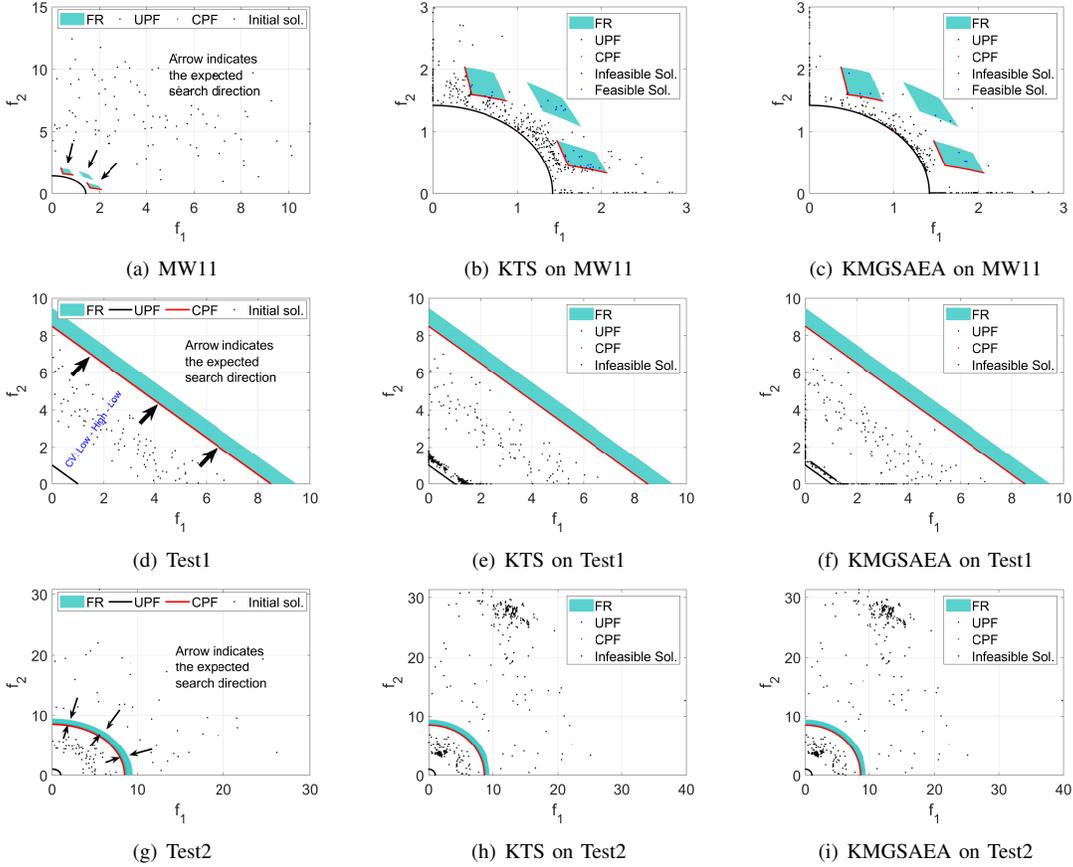


Fig. 2. Test cases to illustrate the influence of the relative position of UPF and CPF on search performance of existing approaches.

In order to solve CMOPs, it is critical to balance convergence, diversity, and feasibility of the solutions during the course of search. Although feasibility is important, discarding all infeasible individuals encountered during the search may significantly impair diversity, limiting exploration of promising regions. Conversely, retaining too many infeasible individuals risks guiding evolution away from the CPF. Besides these, ECMOPs have additional challenges due to limited the number of function evaluations. Although surrogates in SA-CMOEAs help to partially drive the solutions towards promising locations, the information they provide regarding feasibility, convergence and diversity of the candidate solutions is subject to their prediction uncertainties. If not carefully accounted for, the inaccurate information about the landscapes can adversely affect the search performance of the algorithms. The search difficulties encountered in some concrete instances of problems are demonstrated in Fig. 2, taking two most recently proposed SA-CMOEAs, namely KTS [11] and KMGSAAEA [10] as examples.

Scenario 1 shows the case of MW11 test problem [40], shown in Fig. 2(a). The CPF comprises two disconnected feasible regions detached from the UPF by an infeasible region along with a solitary point on the surface of UPF. It also includes one disconnected feasible region yielding no CPF solution. A set of 100 solutions generated using Latin Hypercube Sampling (LHS) is shown in Fig. 2(a), which spans

a large range in the infeasible region. The arrows indicate that both UPF and CPF lie in the same optimization direction (i.e., correlation between ranks based on objectives and CV minimization is positive). This implies that co-evolutionary based approaches will swiftly identify feasible region on the way to the UPF via unconstrained search. Therefore, both KTS and KMGSAAEA are able to identify a few feasible solutions at the end of computation budget (500 evaluations are performed in this study) as shown in Fig. 2(b) - 2(c). However, it is also clearly noticeable that the density of converged solutions is much higher in UPF compared to the CPF for both cases. Due to the partial focus on unconstrained search, a number of evaluations are wasted even after identifying the feasible regions. This situation becomes even more pronounced in scenarios 2 and 3 (Fig. 2(d) and 2(g) respectively). These two test problems, referred to as Test1 and Test2, are constructed based on FCP series [17] in which search is biased towards some local regions in the objective space. In Fig. 2(d), all the initially generated 100 solutions are located in between UPF and CPF which means the expected search direction to achieve CPF is opposite to that of achieving the UPF. The CV is high in the middle region which gradually decreases towards both UPF and CPF. Now, an interesting thing can be observed from Fig. 2(e) - 2(f) that the search is conducted totally opposite to the desired direction by both KTS and KMGSAAEA. Similar situation can be observed for scenario 3

in Fig. 2(g). In this problem, the initial solutions are generated on both sides of the CPF. Between UPF and CPF, there are a couple of infeasible regions with low CV values. As from the illustration in Fig. 2(h) - 2(i), the search is directed towards those infeasible regions by both KTS and KMGSAEA. All the scenarios demonstrate the significant impact that the relative locations of UPF and CPF have on the algorithm performance. Additionally, it is evident that discarding constraint information can sometimes lead to negative impact on the search while solving ECMOPs.

Some of these shortcomings are aimed to overcome through PSCMOEA, which is intended to solve a diverse range of ECMOPs efficiently.

III. PROPOSED APPROACH

The general framework of PSCMOEA² is shown in Fig. 3, which, in brief, comprises the following steps. In **step 1**, the initial solutions are generated and evaluated, followed by training of Kriging surrogate models in **step 2**. In **step 3**, normalization bounds are computed based on the status of the existing archive of evaluated solutions, to conduct the decomposition-based search in **step 4**. The **step 4** executes a search on the surrogate space in order to identify promising candidate solutions. In **step 5**, the infill solution for true evaluation is selected from these promising candidates. The infill solution is evaluated in **step 6**, added to the archive and the process loops back to **step 2**, until the prescribed number of evaluations are exhausted. More details of these steps, along with rationale behind their design, are discussed in the following subsections.

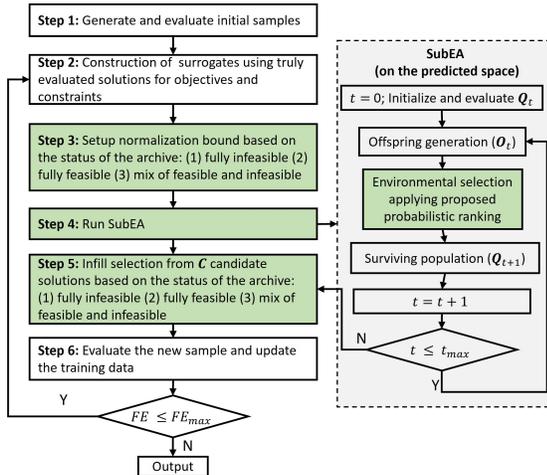


Fig. 3. A general overview of the PSCMOEA framework. The green shaded boxes indicate the steps where this study introduces new contributions.

A. Initialization and Kriging models

The initialization process is quite similar to most canonical SA-CMOEAs. N solutions are generated in the variable space

²For research purposes, Matlab implementation of PSCMOEA will be made available in its final form after manuscript review process is complete; to incorporate any suggested updates.

Algorithm 1 PSCMOEA baseline framework

Require: Maximum function evaluations FE_{max} , initial population size N , sparsity threshold (ϵ), SubEA parameters: population size (N_S), number of generations (G_S), crossover and mutation probability (P_c and P_m), crossover and mutation distribution index (η_c and η_m), infeasible ratio for initial ranking of SubEA (α), RVTagFlag: ‘ON’ or ‘OFF’, SearchFlag: ‘Constrained’ or ‘Unconstrained’.

Output: Constrained Pareto front (CPF) approximation.

```

1:  $\mathbf{X} \leftarrow$  Initialize  $N$  solutions. {Following LHS strategy}
2:  $[\mathbf{F}(\mathbf{x}), \mathbf{G}(\mathbf{x})]_{1:N} \leftarrow$  Evaluate  $(\mathbf{X})$ ;  $FE = |\mathbf{X}|$ .
3:  $\mathbf{A} \leftarrow$  Store initially evaluated database:  $[\mathbf{X}, \mathbf{F}(\mathbf{x})_{1:N}, \mathbf{G}(\mathbf{x})_{1:N}]$ .
4: Identify feasible ( $\mathbf{A}_{feas}$ ) and infeasible ( $\mathbf{A}_{infeas}$ ) solutions in  $\mathbf{A}$ .
5: Set RVTagFlag: ‘OFF’ and SearchFlag: ‘Constrained’.
6: if  $\mathbf{A}_{feas} \neq \emptyset$  then
7:   Initialize the shadow ND archive ( $\mathbf{A}^S$ ). {Section III-D}
8: else
9:    $\mathbf{A}^S \leftarrow \emptyset$ . {Initial shadow ND archive is set empty}
10: end if
11: while ( $FE \leq FE_{max}$ ) do
12:   Train the Kriging model for each objective and constraint.
13:   Set the normalization bounds  $\mathbf{Z}^I$  and  $\mathbf{Z}^N$ . {Algo. 2}
14:    $\mathbf{C} \leftarrow$  SubEA ( $\mathbf{A}$ , models,  $\mathbf{Z}^I$ ,  $\mathbf{Z}^N$ , Flags).
15:   New sample  $\leftarrow$  Infill Identification ( $\mathbf{C}$ ,  $\mathbf{A}$ ,  $\mathbf{A}^S$ , SearchFlag, RV-TagFlag). {Algo. 4}
16:   Evaluate the new sample with expensive functions.
17:   Update archive  $\mathbf{A}$ , shadow ND archive  $\mathbf{A}^S$  and  $FE = FE + 1$ .
18:   Update the RVTagFlag and SearchFlag
19: end while

```

based on LHS sampling and the corresponding objective and constraint functions are computed through true (expensive) evaluations. This dataset is then used to build surrogate models. In this study, Kriging is used, also known as Gaussian Process (GP) regression [4], as the underlying surrogate model for its ability to provide information about uncertainty along with mean predictions. Using the archive \mathbf{A} of all evaluated solutions, a Kriging model is built for each individual objective and constraint function. To ensure model stability, input data is pre-screened by removing solutions closer than $1e^{-4}$ in the normalized variable space. The process of modeling and choice of hyperparameters is consistent with a number of recent studies [5], [41] and the PlatEMO framework [42], hence the details are not replicated here for brevity.

B. Setting the normalization bounds

Normalization plays a crucial role in MOEAs by ensuring a fair basis for comparison when dealing with objectives that cover significantly different numerical ranges [43]. Normalization is particularly important in decomposition-based MOEAs, so that the uniformly distributed reference vectors (RVs) could map to uniform objectives irrespective of their ranges. However, setting suitable bounds for normalization is challenging when true ideal (\mathbf{Z}^I) and Nadir (\mathbf{Z}^N) points are unknown. On the other hand, setting bounds by only considering the feasible ND solutions in the archive \mathbf{A} could make the search overly conservative, diminishing the likelihood of identifying competitive solutions beyond the current range [5], [44]. Another open question is what bounds to use when there is no feasible solution in the archive? In this context, a new and adaptive way of setting the normalization bounds is introduced based on the feasibility status of \mathbf{A} at the beginning of each generation. The bounds will be utilized during search on the surrogate space (will be detailed in the next subsection). The

process of identifying the normalization bounds is outlined in Algo. 2 and further illustrated through Fig. 4.

Algorithm 2 Setting normalization bound

Require: Archive \mathbf{A} .

Output: Normalization bounds, \mathbf{Z}^I and \mathbf{Z}^N .

- 1: Identify feasible (\mathbf{A}_{feas}) and infeasible ($\mathbf{A}_{\text{infeas}}$) solutions in \mathbf{A} and take their corresponding objective values \mathbf{F}_{feas} and $\mathbf{F}_{\text{infeas}}$ respectively.
 - 2: **if** $\mathbf{A}_{\text{feas}} == \emptyset$ **then**
 - 3: $\mathbf{Z}^I = \min\{\mathbf{F}_{\text{infeas}}^i\}^M$; $\mathbf{Z}^N = \max\{\mathbf{F}_{\text{infeas}}^i\}^M$. $\{i = 1, \dots, M\}$
 - 4: **else if** $\mathbf{A}_{\text{infeas}} == \emptyset$ **then**
 - 5: **Identify** $\mathbf{Fnd}_{\text{feas}}$ from \mathbf{A}_{feas} . {Feasible ND set in \mathbf{A}_{feas} }
 - 6: $\mathbf{Z}^I = \min\{\mathbf{Fnd}_{\text{feas}}^i\}^M$; $\mathbf{Z}^N = \max\{\mathbf{Fnd}_{\text{feas}}^i\}^M$. $\{i = 1, \dots, M\}$
 - 7: **Extend** \mathbf{Z}^N following it: $\mathbf{Z}^N = \mathbf{Z}^N + ((\mathbf{Z}^N - \mathbf{Z}^I) \times 0.1)$.
 - 8: **else**
 - 9: **Identify** $\mathbf{Fnd}_{\text{feas}}$ from \mathbf{A}_{feas} .
 - 10: **Identify and list** the infeasible solution(s) which are ND w.r.t any $\mathbf{Fnd}_{\text{feas}}$, presented as $\mathbf{F}_{\text{infeas}}^L$.
 - 11: **Combine** $\mathbf{F}_{\text{comb}} \leftarrow \{\mathbf{Fnd}_{\text{feas}}, \mathbf{F}_{\text{infeas}}^L\}$.
 - 12: $\mathbf{Z}^I = \min\{\mathbf{F}_{\text{comb}}^i\}^M$; $\mathbf{Z}^N = \max\{\mathbf{F}_{\text{comb}}^i\}^M$. $\{i = 1, \dots, M\}$
 - 13: **Extend** \mathbf{Z}^N as: $\mathbf{Z}^N = \mathbf{Z}^N + ((\mathbf{Z}^N - \mathbf{Z}^I) \times 0.1)$.
 - 14: **end if**
-

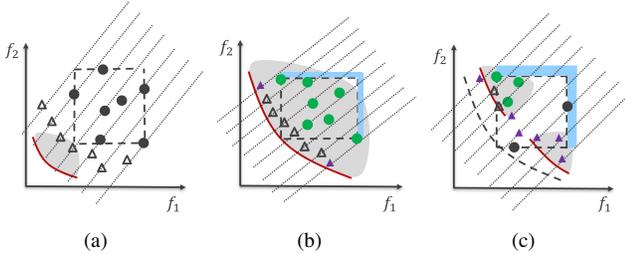


Fig. 4. Normalization bound when archive is (a) fully infeasible, (b) fully feasible and (c) mix of feasible and infeasible. The gray shaded region denotes feasible region. The bluish shaded region in (b) and (c) denotes the extended bounds. The green and black circular dots represents true feasible and infeasible solutions respectively. The triangular points represent the converged predicted candidate set along the RVs.

From Section II-C, it could be inferred that search bounds should adapt in different stages of the search. For example, in the beginning when no feasible solution has been identified, priority has to be given to the exploration when searching for a new candidate solution in the predicted fitness landscape (normalized in terms of current \mathbf{Z}^I and \mathbf{Z}^N). To ensure this, the complete available original objective space is used for searching by calculating the minimum (\mathbf{Z}^I) and maximum (\mathbf{Z}^N) numerical values of each objective in \mathbf{A} (Algo. 2, line 3). As shown in Fig. 4(a), although the initial true solutions are still outside the feasible region, the RVs generated based on the current \mathbf{Z}^I and \mathbf{Z}^N cover the available feasible region and intersect the CPF uniformly, leading to solutions with a good spread in predicted space (denoted with triangles). The second type includes the situation when the full \mathbf{A} is feasible (Algo. 2, line 4 - 7). In this case, \mathbf{Z}^I and \mathbf{Z}^N are calculated by taking the minimum and maximum of feasible non-dominated (ND) objective values respectively. After that, \mathbf{Z}^N is modified by adopting a boundary extension technique introduced in [5]. It has shown efficacy to identify well distributed solutions on the PF regardless of the shape (regular/irregular) [5]. As illustrated in Fig. 4(b), the rectangle with dashed line indicates the current bound which is extended following equation in Algo. 2, line 7,

shown by the blue shaded region. Such a simple extension promotes diversity which increases the likelihood of covering the whole CPF. For example, the two candidates along two extreme RVs represented by magenta colored triangle can be obtained only through this extension. The third case denotes a more frequent scenario encountered by the SA-CMOEAs in which \mathbf{A} contains both feasible and infeasible solutions. In this case, search has to be directed in a way to explore more feasible regions (if present) and at the same time, to sample well distributed solutions on the CPF of the already identified feasible region. Setting the normalization bound at this stage only based on the infeasible solutions (as case 1) would result in misdirected evaluations by searching in an unreasonably wide area. On the other hand, considering the bounds based only on feasible ND solutions might bias the search towards one of multiple disconnected regions shown in Fig. 4(c). Therefore, a novel strategy is proposed by maintaining the trade-off between convergence and diversity in which the high quality infeasible solutions in terms of the objective values are considered alongside the feasible ND solutions to construct the bounds (Algo. 2, line 9 - 13). The list of infeasible solutions which are ND to any feasible ND solution in terms of objectives are added to the feasible ND set to construct the combined set. After that, the minimum and maximum values of each objective in this combined set are considered as the \mathbf{Z}^I and \mathbf{Z}^N respectively, followed by the extension of the \mathbf{Z}^N . The benefit of the proposed strategy can be clearly noticed from Fig. 4(c). By considering the high quality (ND) infeasible solutions, the search can cover both the feasible regions. The magenta colored solutions are only obtained (in the predicted space) using the proposed normalization bound. Using these bounds, the predicted value (μ) and the associated uncertainty (σ) of the solution are scaled using Eq. 2 for the subsequent steps.

$$\mu'_i(\mathbf{x}) = \frac{\mu_i(\mathbf{x}) - \mathbf{Z}_i^I}{\mathbf{Z}_i^N - \mathbf{Z}_i^I}; \sigma'_i(\mathbf{x}) = \frac{\sigma_i(\mathbf{x})}{(\mathbf{Z}_i^N - \mathbf{Z}_i^I)^2}; i = 1, \dots, M \quad (2)$$

C. SubEA to identify promising candidate solutions

As discussed in the overview and outlined in Algo. 1, line 14, a SubEA is used to search the surrogate to identify a promising candidate pool for infill identification in the subsequent step. A decomposition-based evolutionary approach is employed for this purpose, with *mirrored* reference vectors that have been shown to perform well for irregular shaped Pareto fronts [5], [45]. An illustration of mirrored RVs can be seen in Fig. 4, represented by black dotted lines. These are formed by connecting the uniformly generated reference points on the unit simplex using Normal Boundary Intersection (NBI) method with their corresponding mirrored points (which will lie in third quadrant for Fig. 4). The details of SubEA are given below, of which the environmental selection using probabilistic constraint dominance is where this work makes the key contribution, and hence is discussed in more detail. Note that all fitness values used in SubEA undergo normalization using the bounds identified previously (Algo. 2). Furthermore, as SubEA operates on the surrogate landscape, it doesn't involve any expensive evaluations.

1) *Initialization and evaluation*: In this step, a predefined number of samples (population size, N_S) are required as the initial population for surrogate assisted search. For enhanced seeding [46], part of the truly evaluated solutions in \mathbf{A} are inherited directly into this population. For this, the solutions in $|\mathbf{A}|$ are ordered using infeasibility-driven ranking strategy [18] by maintaining recommended 20% marginally infeasible solutions on top of the feasible ones. The top N_S solutions are kept if $|\mathbf{A}| > N_S$. Otherwise, all solutions of \mathbf{A} are kept followed by the remaining ones generated using LHS sampling. The performance of any newly generated solution(s) is estimated using the surrogate models.

2) *Offspring generation*: Offspring solutions are evolved from the parent population using simulated binary crossover (SBX) and polynomial mutation (PM) [47]. If any of these offspring solutions turn out identical to either the parent population and/or any truly evaluated solution in the archive, the offspring is substituted with a new randomly generated solution. Two solutions are deemed identical if the Euclidean distance computed between them in the normalized variable space is less than a tolerance of $1e^{-4}$.

3) *Environmental selection*: The SubEA follows a population based approach, wherein N_S new samples are evolved and evaluated (using surrogates) in each generation. Out of the $2N_S$ (parent+offspring) solutions, denoted as \mathbf{Q} , N_S are retained for the next generation through environmental selection. The selection process discussed below is developed through careful considerations of the surrogate predictions and associated uncertainties of the objectives and constraints, and forms one of the key contributions of this study. The pseudo-code of the selection scheme is outlined in Algo. 3.

The mean and uncertainty of the response functions for the $2N_S$ solutions are denoted as $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$, respectively, estimated from Kriging models. Given the set \mathbf{W} of RVs, each of these $2N_S$ solutions are assigned to its closest RV based on minimum angle (denoted as \angle) assignment [48] in the predicted objective space. For this, the objective distributions of $2N_S$ solutions are scaled using Eq. 2. This assignment may lead to one or more solutions being associated with a single RV (referred to as active RV), while some RVs may not have any solution associated with them (inactive RV).

For comparing solutions assigned to each (active) RV ($\hat{\mathbf{W}}_j$; $\{j = 1, \dots, N_W (= \text{No. of RVs})\}$) an integrated performance criterion is proposed, denoted *probability of constrained domination (PCD)* shown in Eq. 3, which quantitatively estimates the probability of a solution \mathbf{x} being better than solution \mathbf{y} . PCD comprises of three terms, capturing three scenarios respectively: (a) \mathbf{x} is feasible and \mathbf{y} is not; (b) both are feasible and \mathbf{x} dominates \mathbf{y} in terms of objective values; and (c) both are infeasible and \mathbf{x} has a lower *CV* than \mathbf{y} .

$$PCD = \mathbf{P}(\mathbf{x} \prec \mathbf{y}) = \underbrace{\text{PoF}_{\mathbf{x}}(1 - \text{PoF}_{\mathbf{y}})}_{\mathbf{x} \text{ feasible, } \mathbf{y} \text{ infeasible}} + \underbrace{\text{PoF}_{\mathbf{x}}\text{PoF}_{\mathbf{y}}\mathbf{P}(\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{y}))}_{\mathbf{x} \text{ feasible, } \mathbf{y} \text{ feasible}} + \underbrace{(1 - \text{PoF}_{\mathbf{x}})(1 - \text{PoF}_{\mathbf{y}})\mathbf{P}(CV(\mathbf{x}) < CV(\mathbf{y}))}_{\mathbf{x} \text{ infeasible, } \mathbf{y} \text{ infeasible}} \quad (3)$$

In the above equations, the probability of feasibility (PoF) estimates the likelihood that a solution (e.g. \mathbf{x}) falls into

Algorithm 3 Environmental selection method for SubEA

Require: Population to order \mathbf{Q} (of size $\approx 2N_S$), \mathbf{Z}^I , \mathbf{Z}^N , SearchFlag.
Output: A set of solutions N_W , one assigned to each reference vector.

- 1: Normalize the objective vectors in \mathbf{Q} using the \mathbf{Z}^I and \mathbf{Z}^N .
 - 2: Generate reference vectors (RVs) \mathbf{W} and set $n_W = N_W = |\mathbf{W}|$.
 - 3: **while** $n_W \neq \emptyset$ **do**
 - 4: Assign each solution to its closest RV using angle as the proximity metric: $\angle_i(\boldsymbol{\mu}_i, \mathbf{W})$; $i = 1, \dots, \mathbf{Q}$.
 - 5: Project the assigned solutions along the corresponding RV.
 - 6: **if** SearchFlag == ‘Constrained’ **then**
 - 7: Along each RV, compare each solution pairwise with all others according to Eq. 3. The average of the pairwise scores yields the final score of the corresponding solution.
 - 8: **else if** SearchFlag == ‘Unconstrained’ **then**
 - 9: Along each RV, compare each solution pairwise with all others according to Eq. 6. The average of the pairwise scores yields the final score of the corresponding solution.
 - 10: **end if**
 - 11: Select the solution with the highest score along each RV.
 - 12: Remove the selected solutions and the RVs (which had at least one solution assigned) from \mathbf{Q} and \mathbf{W} respectively.
 - 13: $n_W = |\mathbf{W}|$
 - 14: **end while**
-

the feasible region, computed by integrating the joint prior distributions of all constraints as shown in Eq. 4.

$$\text{PoF}_{\mathbf{x}} = \prod_{i=1}^p \Phi(g_i(\mathbf{x}) \leq 0) = \prod_{i=1}^p \Phi\left(-\frac{\mu_{g_i}(\mathbf{x})}{\sigma_{g_i}(\mathbf{x})}\right) \quad (4)$$

To estimate the dominance as per the second component of Eq. 3, the distributions of the candidates along each objective $(\mu_{f_i}(\mathbf{x}), \sigma_{f_i}(\mathbf{x}))$ are projected along the corresponding $\hat{\mathbf{W}}_j$. The projection of the distributions on the RV, using Eq. 5, yields single $\mu_f(\mathbf{x})$ and $\sigma_f(\mathbf{x})$ for a solution, \mathbf{x} .

$$\mu_f(\mathbf{x}) = \hat{\mathbf{W}}_j \boldsymbol{\mu}^T; \quad \sigma_f(\mathbf{x}) = \hat{\mathbf{W}}_j \boldsymbol{\Psi} \hat{\mathbf{W}}_j^T \quad (5)$$

Here, $\boldsymbol{\Psi}$ represents the covariance matrix $\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$ and $\mu_f(\mathbf{x})$, $\sigma_f(\mathbf{x})$ are the mean and standard deviation, respectively, projected along the RV. The probability that \mathbf{x} dominates \mathbf{y} in terms of the M original objectives is estimated by applying the concept of probabilistic dominance (PDF) [49], as shown in Eq. 6.

$$\mathbf{P}(\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{y})) = \prod_{i=1}^M \left(\frac{1}{2} + \frac{1}{2} \text{erf} \left(\frac{\mu_{f_i}(\mathbf{y}) - \mu_{f_i}(\mathbf{x})}{\sqrt{2(\sigma_{f_i}(\mathbf{x})^2 + \sigma_{f_i}(\mathbf{y})^2)}} \right) \right) \quad (6)$$

where, error function ‘erf’ is defined as:

$$\text{erf}(\mathbf{x}) = \frac{2}{\sqrt{\pi}} \int_0^{\mathbf{x}} e^{-t^2} dt \quad (7)$$

For the third component, the probability that $CV(\mathbf{x}) < CV(\mathbf{y})$ is needed to be estimated. However, from the individual Kriging models, we only get the individual constraint distributions of each solution. Computing $P(CV(\mathbf{x}) < CV(\mathbf{y}))$ directly based on these distributions is not appropriate since a highly feasible distribution in the predicted landscape can bias the estimation, therefore resulting in incorrect classification. To overcome this, The concept of rectified Gaussian distribution has been adopted to determine the corresponding CV

distributions. In probability theory, it represents a modification of the Gaussian distribution wherein its negative elements are reset to 0, analogous to an electronic rectifier. Essentially, it is a combination of a discrete distribution (constant 0) and a continuous distribution (a truncated Gaussian distribution with the interval $(0, \infty)$ due to censoring [50]).

For a normal distribution $\mathcal{N}(\mu, \sigma^2)$, the mean of the rectified distribution (μ_R) yields a higher value than μ since some probability density is shifted to a higher value (since negative values are transformed to 0). Conversely, the rectified standard deviation (σ_R) is smaller than σ . Following the method proposed in [51], the interval between $a = 0$ and $b = \mu + 6\sigma$ (to allow 6σ limit) is set first to be acting on a standard normal distribution: $c = \frac{a-\mu}{\sigma}$, $d = \frac{b-\mu}{\sigma}$. Using the transformed constraints, μ_R and σ_R^2 can be expressed as shown in Eq. 8.

$$\begin{aligned} \mu_t &= \frac{1}{\sqrt{2\pi}} \left(e^{-\frac{c^2}{2}} - e^{-\frac{d^2}{2}} \right) + \frac{c}{2} \left(1 + \operatorname{erf} \left(\frac{c}{\sqrt{2}} \right) \right) \\ &\quad + \frac{d}{2} \left(1 - \operatorname{erf} \left(\frac{d}{\sqrt{2}} \right) \right), \\ \sigma_t^2 &= \frac{\mu_t^2 + 1}{2} \left(\operatorname{erf} \left(\frac{d}{\sqrt{2}} \right) - \operatorname{erf} \left(\frac{c}{\sqrt{2}} \right) \right) \\ &\quad - \frac{1}{\sqrt{2\pi}} \left((d - 2\mu_t) e^{-\frac{d^2}{2}} - (c - 2\mu_t) e^{-\frac{c^2}{2}} \right) \\ &\quad + \frac{(c - \mu_t)^2}{2} \left(1 + \operatorname{erf} \left(\frac{c}{\sqrt{2}} \right) \right) + \frac{(d - \mu_t)^2}{2} \left(1 - \operatorname{erf} \left(\frac{d}{\sqrt{2}} \right) \right), \\ \mu_R &= \mu + \sigma\mu_t, \sigma_R^2 = \sigma^2\sigma_t^2 \end{aligned} \quad (8)$$

Once the CV distribution of each individual constraint is available, and assuming that the individual CV distributions are mutually independent, the estimated total (sum of) CV of a solution, \mathbf{x} can be mathematically expressed as: $\mathcal{N} \left(\sum_{i=1}^p \mu_{g_i}, \sum_{i=1}^p \sigma_{g_i}^2 \right)$.

For a given solution \mathbf{x} , the above pairwise comparison is done with all solutions associated with the same RV, and the corresponding *PCD* values are averaged to yield its overall fitness score. This is repeated for each solution along the given RV, and solution with the highest score is selected as the best performing solution along the RV. A proof-of-concept of ranking along a given RV is shown in Fig. 5 for DASCOP1 problem [52] based on 100 samples. The samples span both feasible and infeasible regions of the search space. The colormap of the solutions denotes the score (higher the better) in a normalized scale. It can be seen that the selected candidate is competitive, as it is feasible and dominates all other feasible solutions. The scores of all other candidates also reflect their quality well in terms of convergence.

The above procedure is repeated for all active RVs. Subsequently, all the active RVs and their corresponding best solutions are selected from \mathbf{W} and \mathbf{Q} , respectively. Using the remaining set of RVs and solutions, the process (involving assignment, projection, and selection) is repeated, until no empty RV remains. Thus, in the end, each RV is paired with an associated best solution (totaling N_W solutions). In the scenario where $N_S > N_W$, the remaining $N_S - N_W$ solutions are randomly chosen to form the surviving set of N_S solutions for the subsequent generation.

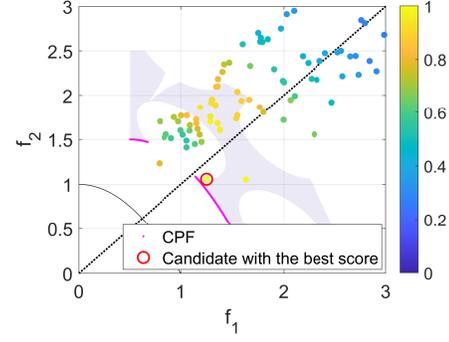


Fig. 5. Best candidate selection along the RV according to Eq. 3.

D. Infill sample identification

In this stage one solution out of the population (\mathbf{C}) delivered by the SubEA needs to be identified for expensive evaluation. To maintain dynamic balance among feasibility, convergence and diversity, the sampling criteria is set based on the status of archive \mathbf{A} : (a) \mathbf{A} is fully infeasible – in this case feasibility and diversity are targeted; (b) \mathbf{A} is fully feasible – convergence and diversity are targeted; and (c) \mathbf{A} is a mix of feasible and infeasible solutions – feasibility is targeted, followed by convergence and diversity. The proposed sampling process is outlined in Algo. 4, noting that any duplicate solutions in \mathbf{C} are removed a priori. Each of the above three cases are discussed in more detail below.

Algorithm 4 Infill identification for expensive evaluation

Require: Archive \mathbf{A} , candidates \mathbf{C} , \mathbf{Z}^I , \mathbf{Z}^N , SearchFlag, RVTagFlag.

Output: Infill sample, \mathbf{C}^{In} for expensive evaluation.

- 1: Identify feasible (\mathbf{A}_{feas}) and infeasible ($\mathbf{A}_{\text{infeas}}$) solutions in \mathbf{A} .
- 2: **if** $\mathbf{A}_{\text{feas}} == \emptyset$ **then**
- 3: **for** $i = 1 : |\mathbf{C}|$ **do**
- 4: **for** $j = 1 : |\mathbf{C}|$ **do**
- 5: Compute $\mathbf{P}(C_i < C_j)$. {According to PCD (Eq. 3) if SearchFlag is ‘Constrained’; otherwise based on PDF (Eq. 6)}
- 6: **end for**
- 7: Score _{i} \leftarrow mean{ $\mathbf{P}(C_i < C_{j=1:|\mathbf{C}|})$ }.
- 8: **end for**
- 9: SC \leftarrow {Score₁, ..., Score _{$|\mathbf{C}|$} }. {Final score list of all \mathbf{C} }
- 10: Arrange \mathbf{C} in descending order of SC and their corresponding RVs are tagged accordingly.
- 11: **if** RVTagFlag == ‘ON’ **then**
- 12: Select the top scored candidate along a unique RV as the new infill, \mathbf{C}^{In} .
- 13: Store the selected RV for subsequent stage.
- 14: **end if**
- 15: Select the top scored candidate as the new infill, \mathbf{C}^{In} and store the corresponding RV for subsequent stage.
- 16: Return RVTagFlag = ‘ON’.
- 17: **else if** $\mathbf{A}_{\text{infeas}} == \emptyset$ **then**
- 18: Identify the reference set of solutions, \mathbf{A}_{ref} where $\mathbf{A}_{\text{ref}} = \mathbf{A}_{\text{ND}}$.
- 19: $\mathbf{C}^{In} \leftarrow$ Infill(\mathbf{A} , \mathbf{C} , \mathbf{Z}^I , \mathbf{Z}^N , \mathbf{A}_{ref}). [5]
- 20: **else**
- 21: Identify $\mathbf{Fnd}_{\text{feas}}$ from \mathbf{A}_{feas} .
- 22: Identify and list the infeasible solution(s) which are ND w.r.t any $\mathbf{Fnd}_{\text{feas}}$, presented as $\mathbf{F}_{\text{infeas}}^L$.
- 23: Combine $\mathbf{F}_{\text{comb}} \leftarrow$ { $\mathbf{Fnd}_{\text{feas}}$, $\mathbf{F}_{\text{infeas}}^L$ } and treat them as \mathbf{A}_{ref} .
- 24: $\mathbf{C}^{In} \leftarrow$ Infill(\mathbf{A} , \mathbf{C} , \mathbf{Z}^I , \mathbf{Z}^N , \mathbf{A}_{ref}). [5]
- 25: **end if**

Until no feasible solution has been identified in \mathbf{A} , the preferred infill solution is the one that is most likely of being

feasible along a new RV, to enhance feasibility and diversity. For this, the average score of each solution in \mathbf{C} is computed by comparing it with all other solutions in the set using Eq. 3, and the one with highest score is chosen for evaluation. The selection process is illustrated in Fig. 6. The triangular markers represent candidate set \mathbf{C} , along the RVs in the predicted landscape. The one with highest score is highlighted in magenta and selected for true evaluation at iteration t as shown in Fig. 6(a). The green circular marker shows the solution in the original objective space after evaluation. The corresponding RV (blue solid line) is also stored in \mathbf{A} , to promote diversity in the subsequent selection process. Fig. 6(b) represents the selection stage of iteration $(t + 1)$, where the search bound is still unchanged due to not achieving improved performance from iteration (t) . The selection of blue RV again (same RV as the previous iteration) indicates that the corresponding solution based on the prediction still represents the best performing one. In such a scenario, the selection strategy avoids the blue RV sample and instead selects the next best performing candidate (along yellow RV). The same process is repeated as long as the search bound in the original objective space stays unchanged and no feasible solution is identified in \mathbf{A} . This is maintained by the ‘RVTagFlag’ as mentioned in Algo. 4, line 11 - 16. This simple diversity enhancing approach encourages search for feasible regions from different directions resulting in enhanced performance where the problem has multiple disconnected feasible regions.

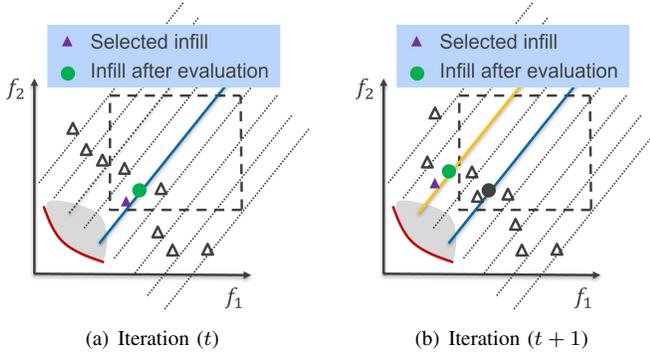


Fig. 6. Infill selection when the full evaluated archive \mathbf{A} is infeasible.

On the other hand, when some or all solutions in \mathbf{A} are feasible, the intent is to promote convergence and diversity during infill sample selection aiming for well distributed solutions on the CPF, outlined in Algo. 4, line 17 - 25. This is done through an adapted version of two-state infill identification, which as demonstrated in [5] for unconstrained problems. In the first stage, the subset of \mathbf{C} that is ND w.r.t. a reference set \mathbf{A}_{ref} is identified. In case no such solution exists in \mathbf{C} , the ND set of \mathbf{C} is chosen instead. In the second stage, a distance-based selection (DSS) approach [53] is applied on the subset to select the most diverse candidate as the final infill solution. The key element here is the use of Mahalanobis distance (MD) [5] for computing distance instead of commonly used Euclidean distance (ED). This is because MD considers both the mean and uncertainties in the computation of distances (in this instance, characterized by predicted mean and uncertainty of

the Kriging model).

The notable difference in this study from [5] is the construction of \mathbf{A}_{ref} . In [5], \mathbf{A}_{ref} is simply the ND set of archived solutions (\mathbf{A}_{ND}), since there are no infeasible solutions to deal with in unconstrained problems. However, here, when the archive includes a mix of feasible and infeasible solutions, \mathbf{A}_{ref} is constructed by assembling feasible ND solution(s) ($\mathbf{Fnd}_{\text{feas}}$) with the set of infeasible solution(s) which are ND w.r.t any $\mathbf{Fnd}_{\text{feas}}$ in terms of the original objectives (Algo. 4, line 21 - 23). This is consistent with the setting of normalization bound as discussed in Section III-B. The general idea behind preserving the infeasible solution(s) with good objective performance in \mathbf{A}_{ref} is to explore diverse areas for sampling if any feasible region still remains undiscovered. This will particularly benefit the search in problems with multiple disconnected feasible regions.

Finally, the concept of ‘shadow ND archive’ (\mathbf{A}^{S}) is also adopted from [5] with slight modification. It contains the set of infill solutions that ended up being dominated after evaluation, or whose closest solution in \mathbf{A} is dominated. This set is included into \mathbf{A}_{ref} while applying DSS to reduce the possibility of selecting an infill in a dominated region. Unlike [5], in the presence of constraints, apart from the newly evaluated solution being dominated, there is a chance of it being infeasible as well. When such a solution is encountered, it is also assigned to \mathbf{A}^{S} . Note that \mathbf{A}^{S} is only activated and updated progressively from the point of identifying a feasible solution in \mathbf{A} . If feasible solution(s) exists in the initial archive, \mathbf{A}^{S} is constructed at the beginning as outlined in Algo. 1, line 6 - 10. Initially, the infeasible solutions which are ND to any feasible solution, are added to the \mathbf{A}^{S} .

After the MD computations based on the above sets, the candidate solution in the subset of \mathbf{C} with the highest merit is selected as the final infill solution, employing the same principle as in DSS. The solution then undergoes true (expensive) evaluation.

E. Search switching mechanism

To expedite feasible region identification and convergence, an adaptive search switching mechanism is additionally embedded within PSCMOEA framework which is only triggered based on certain conditions. The ‘SearchFlag’ essentially controls the environmental selection step in SubEA (as discussed in Section III-C3) and infill selection (when \mathbf{A} is fully infeasible) by setting ‘Unconstrained’ or ‘Constrained’ based probabilistic ranking criteria. Note that ‘Constrained’ flag is the default whereas the ‘Unconstrained’ flag is only activated based on two specific conditions, which are (a) no feasible solution is identified so far in the \mathbf{A} and (b) the Kendall-tau rank correlation [54] between objective optimization and CV minimization is at least moderately positive (defined by a value ≥ 0.27 in this study, based on approximate ranges suggested in statistics literature). This general idea is to bias the search towards the UPF for identifying potential feasible location(s), when the UPF and CPF are along the same search direction. This offers the strategy to adapt based on the problem characteristics and is effective in expediting the

convergence. For determining the rank correlation, referred to as τ , two ranks are derived for every solution. The first denotes the rank of the solution based on its CV (in ascending order) and the second rank is based on their corresponding front by front rank calculated by non-dominated sorting approach. The correlation between these two is denoted by τ . Once/if the ‘Unconstrained’ search is triggered, it is continued to the subsequent iteration only if the latest evaluated solution yields the minimum CV value, otherwise it returns back to regular ‘Constrained’ search. The switching mechanism is also terminated once the first feasible solution is identified.

IV. NUMERICAL EXPERIMENTS

The proposed PSCMOEA is assessed on multiple well-known benchmark problem suites to cover a wide range of problem characteristics. The performance is compared with five state-of-the-art SA-CMOEAs proposed in recent years. Additionally, an ablation study is conducted to highlight the functionality and significance of some the algorithm’s key components.

A. Test problems and experimental settings

As detailed in Section II-C, two test cases, namely Test1 and Test2, are formulated in this study to highlight some challenging scenarios, the detailed formulations of which are in the supplementary online material (SOM Section I). Besides this, numerical experiments are conducted on three distinct set of popular benchmark test problems including MW series [40] (14 instances), LIRCOP series [55] (14 instances) and DASCOP series [52] (9 instances). While most of the problems involves bi-objective formulations, few of them such as MW4, MW8, MW14, LIRCOP13 - LIRCOP14 and DASCOP7 - DASCOP9 contain 3 objectives. For this study, variable dimension (D) is consistently set to as 10. These test suites collectively cover a wide range of challenges, such as low feasibility, disconnected CPFs, local fronts, and irregular CPF shapes.

For PSCMOEA, the crossover and mutation probability is set to as 0.9 and 0.1, respectively with distribution index as 10 and 20, respectively. Regarding SubEA, a population of 100 solutions undergoes evolution over 100 generations. The spacing parameter (H) for generating RVs is set to 99 for 2-objective instances and 12 for 3-objective instances. The five representative state-of-the-art algorithms considered are KMGSAEA [10], ASA-MOEA/D [12], KTS [11], MultiObjectiveEGO [9], and SILE [13]. Four of them have been published in last two years (2022-23) and evaluate more than one solution per generation, while one of them, MultiObjectiveEGO, is a steady-state method that evaluates one solution per generation. The global parameter settings remain consistent across all algorithms. For KMGSAEA, MultiObjectiveEGO and SILE including PSCMOEA, the initial sample size is determined as $11D - 1$, where D represents the number of variables. Note that KTS is an exception, with a fixed sample size of 100 for all cases due to algorithmic requirements. Similarly, ASA-MOEA/D uses 50 and 91 initial samples for 2- and 3-objective instances respectively. The initial samples for each

independent run are generated using the same seed for all algorithms, ensuring a fair comparison without initialization bias. The experiments are conducted with maximum evaluations $FE_{max} = 500$ for all algorithms and all problems. Other algorithm specific parameters are set to their defaults recommended in the respective publications. Results of 31 runs were used for statistical performance comparison.

B. Performance measurement

To measure the quality of the obtained CPF approximations in terms of convergence and diversity, three widely recognized metrics are utilized: IGD [56], IGD⁺ [57] and HV [58]. The reference point for HV calculation is set to the commonly used $(1.1, \dots, 1.1)$. For IGD and IGD⁺, reference CPF sets with 1000 points were generated using PlatEMO for MW, LIRCOP, and DASCOP series. For Test1 and Test2 problems, same reference sets are adopted as FCP1 and FCP3 test problems [17] as they share identical CPFs. All metrics are computed in the normalized objective space using true ideal and nadir points. The computation is performed on the feasible points obtained by an algorithm in each run for a given problem. If an algorithm is unable to identify any feasible solution in a given run, the following process is followed. The worst metric value in terms of IGD, IGD⁺ or HV is determined across all the considered algorithms for a particular problem, and is deteriorated slightly further by adding for (IGD, IGD⁺) or subtracting (HV) an additional small quantity 0.1.

To measure how efficiently the algorithm finds feasible solutions, the number of evaluations required for the algorithm to identify the first feasible solution (FFE) is tracked. In case no feasible solution is identified in a particular run, FFE is assigned as the $FE_{max} = 500$. Additionally, to establish the consistency in performance of an algorithm, the number of successful trials (ST) is counted, where the algorithm successfully identified at least one feasible solution. Notably, among the test problems, some are ‘‘feasibility-hard’’ for which it is difficult to locate a feasible solution within a limited budget, such as MW series, LIRCOP1 - LIRCOP4 and DASCOP5 - DASCOP9. On the other hand, some problems such as LIRCOP5 - LIRCOP14 and DASCOP1 - DASCOP3, are feasibility-easy, where feasible solutions may often exist in the initial randomly generated samples itself.

To establish statistical significance, Wilcoxon ranksum test with 95% confidence interval is utilized. In the pairwise comparisons, a statistically significantly better, worse and equivalent performance in terms of a given metric is classified as ‘win’ (\uparrow), ‘loss’ (\approx) and ‘tie’ (\downarrow) respectively.

Lastly, to visually summarize the cumulative performance of the algorithms across multiple problems, *Performance profiles* [59] based on mean IGD and mean FFE are utilized. In these plots, the x-axis indicates the ratio (λ) of the performance of a given algorithm compared to the best performing algorithm for the instance (lower the better). The y-axis ($\rho_s(\lambda)$) denotes the proportion of total instances solved by the algorithm for a given λ . A higher area under the curve signifies a better cumulative performance.

C. Performance comparison in terms of FFE and ST

The statistics obtained FFE and ST across 31 runs are shown in Table I. From the ST values, it is clearly noticeable that PSCMOEA is overall the most successful and consistent method in identifying feasible regions for the feasibility hard problems such as Test1, Test2, MW1, MW10, LIRCMOP1, LIRCMOP2, DASCMP9 etc. For DASCMP9, PSCMOEA identifies feasible solution in all runs, no other peer algorithm could identify any feasible solution in any run. Similar trends can be noticed from Test1, Test2, MW10 problems as well. There are only a few exceptions where PSCMOEA is not the best but is competitive, such as MW8, MW11, DASCMP5, DASCMP6 (inferior to KMGSAEA) and LIRCMOP3 (inferior to KTS). The overall performance establishes the credibility of the proposed probabilistic selection strategy to drive the search in the predicted landscape. The validation of correct search behavior can be confirmed from the illustration in Fig. 7 where its clear that PSCMOEA could guide the search towards the CPF for Test1/Test2 problems even though UPF is not located in same search direction as CPF. As for the FFE statistics, PSCMOEA once again achieves superior performance in most of the feasibility-hard problems compared to its counterparts. This means that PSCMOEA is able to identify feasible solution swiftly. Thus, fewer function evaluations are spent on the infeasible space, leaving more of the budget towards identifying a well distributed set of CPF solutions. Additionally, the cumulative results of the pairwise significance test ($\uparrow/\approx/\downarrow$ of compared method vs PSCMOEA) at the bottom of Table I confirms the clear advantage of PSCMOEA over it's peers in terms of FFE. This is also reflected strongly in the performance profile in terms of mean FFE presented in Fig. 8(a) for the feasibility-hard problems. PSCMOEA obtains the best results in close to 70% of the instances, and competitive performance in remaining problems with low τ values relative to the peers.

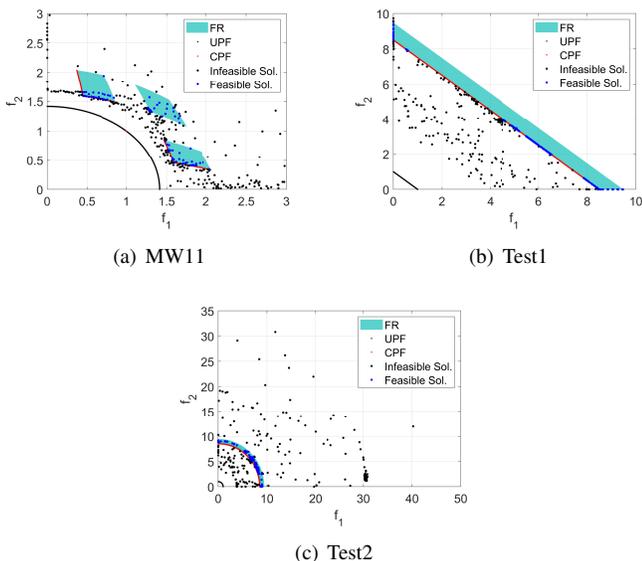


Fig. 7. Distribution of the evaluated solutions for the median run (out of 31) in terms of IGD by PSCMOEA.

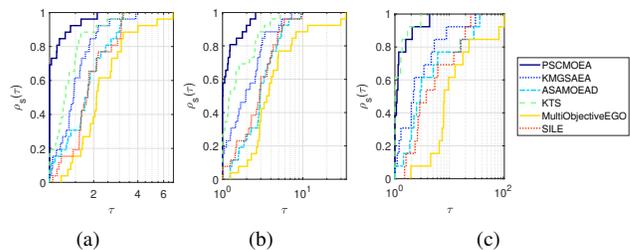


Fig. 8. Performance profiles based on (a) mean FFE and (b) mean IGD for feasibility-hard problems (Test1-2, MW1-14, LIRCMOP1-4, DASCMP4-9), (c) mean IGD for feasibility-easy problems (LIRCMOP5-14 and DASCMP1-3).

D. Performance comparison in terms of IGD, IGD⁺ and HV

The IGD statistics based on 31 runs are presented in Table II, where PSCMOEA shows the best performance in 18 (out of 39) instances followed by KTS in 13 instances. However, KTS performs the best on LIRCMOP2 - LIRCMOP4 where the feasible region essentially is a very narrow strip. From our observation, both PSCMOEA and KTS could identify only a few feasible solutions (maximum 2 or 3). The larger values in the metric is only due to one additional feasible solution. PSCMOEA performs the best for most of the feasibility-hard problems since it could identify the feasible solutions faster and more consistently, leaving remaining computational budget for obtaining high quality CPF approximation. The pairwise significance tests based on IGD ($\uparrow/\approx/\downarrow$ of the compared algorithms against PSCMOEA) are presented at the bottom of Table II. It can be seen that peer algorithms obtain significantly worse performance metrics than PSCMOEA. KTS performs best among them but shows worse performance than PSCMOEA in 18 instances, while being better in 7 instances including LIRCMOP2 - LIRCMOP4 as discussed above. For reference, the total $\uparrow/\approx/\downarrow$ statistics in terms of IGD⁺ and HV is also added at the bottom which also signifies the similar performance trend. More detailed tables of IGD⁺ and HV for MW, LIRCMOP and DASCMP series are included in the SOM II, III and IV respectively. The plots of the obtained solutions by all compared algorithms on MW, LIRCMOP and DASCMP series problems are also included as supplementary data for the interested readers in SOM Sections II, III and IV respectively, due to space constraints.

The performance profiles, based on mean IGD, are presented in Fig. 8 to summarize cumulative performance of the algorithms. Two profiles are drawn, corresponding to feasibility-hard and feasibility-easy problems (Fig. 8(b) and easy (Fig. 8(c), respectively). For the former category, PSCMOEA clearly outperforms the other algorithm by obtaining the best results in almost 60% of the total instances, and being competitive on the remaining instances. For the feasibility-easy problems, KTS shows marginally better performance than PSCMOEA, since it could use total evaluation budget to search for a good approximation of the CPF without wasting evaluations searching for UPF in such cases. PSCMOEA displays very similar profile to (and better than others), highlighting its ability to identify well distributed set of solutions on CPF.

TABLE I

MEAN FFE (STD.) AND ST ACROSS 31 TRIALS OBTAINED BY PSCMOEA AND PEER-ALGORITHMS. $\uparrow/\approx/\downarrow$ SYMBOLS DENOTE WHETHER THE CORRESPONDING ALGORITHM IS SIGNIFICANTLY BETTER/ EQUIVALENT/ WORSE THAN PSCMOEA. THE GRAY SHADE HIGHLIGHTS THE BEST MEAN PERFORMANCE.

Problems	M	PSCMOEA		KMGSAAE		ASA-MOEA/D		KTS		MultiObjectiveEGO		SILE	
		FFE	ST	FFE	ST	FFE	ST	FFE	ST	FFE	ST	FFE	ST
Test1	2	129 (50)	31	500 (0) ↓	0	168 (183) ↑	24	421 (149) ↓	7	336 (147) ↓	25	220 (146) ↓	25
Test2	2	138 (116)	29	354 (216) ↓	10	150 (148) ≈	28	341 (218) ↓	11	299 (210) ↓	17	296 (209) ↓	16
MW1	2	296 (123)	27	433 (74) ↓	20	485 (43) ↓	5	484 (47) ↓	6	500 (0) ↓	0	484 (43) ↓	5
MW2	2	147 (28)	31	218 (68) ↓	30	196 (86) ↓	31	166 (69) ≈	31	211 (50) ↓	31	235 (60) ↓	31
MW3	2	118 (8)	31	119 (4) ≈	31	72 (22) ↑	31	107 (11) ↑	31	499 (3) ↓	1	131 (17) ↓	31
MW4	3	263 (81)	30	387 (101) ↓	18	496 (12) ↓	4	396 (88) ↓	22	500 (0) ↓	0	487 (43) ↓	3
MW5	2	252 (68)	31	344 (66) ↓	29	437 (90) ↓	12	373 (118) ↓	20	500 (0) ↓	0	467 (76) ↓	6
MW6	2	157 (29)	31	292 (118) ↓	28	379 (125) ↓	19	194 (60) ↓	31	320 (96) ↓	28	330 (100) ↓	27
MW7	2	125 (9)	31	122 (11) ≈	31	90 (16) ↑	31	111 (8) ↑	31	487 (73) ↓	1	143 (18) ↓	31
MW8	3	229 (106)	30	243 (48) ≈	31	376 (115) ↓	22	205 (92) ≈	29	428 (115) ↓	12	339 (118) ↓	23
MW9	2	306 (76)	31	321 (73) ≈	30	494 (30) ↓	2	394 (92) ↓	23	500 (0) ↓	0	500 (0) ↓	0
MW10	2	416 (118)	14	463 (70) ≈	11	500 (0) ↓	0	483 (48) ↓	4	500 (0) ↓	0	500 (0) ↓	0
MW11	2	258 (91)	30	122 (15) ↑	31	273 (85) ≈	31	147 (56) ↑	31	489 (57) ↓	2	374 (149) ↓	17
MW12	2	233 (71)	31	374 (52) ↓	29	441 (105) ↓	10	317 (98) ↓	29	500 (0) ↓	0	429 (84) ↓	19
MW13	2	119 (50)	31	109 (37) ≈	31	108 (55) ≈	31	106 (38) ≈	31	281 (194) ↓	24	119 (45) ≈	31
MW14	3	142 (28)	31	179 (61) ↓	31	240 (95) ↓	31	138 (50) ≈	31	184 (54) ↓	31	145 (53) ≈	31
LIRCMOP1	2	342 (90)	30	480 (39) ↓	10	478 (79) ↓	3	376 (99) ≈	23	477 (94) ↓	2	395 (100) ↓	25
LIRCMOP2	2	279 (60)	31	490 (26) ↓	4	461 (126) ↓	3	325 (94) ≈	27	467 (93) ↓	4	404 (93) ↓	24
LIRCMOP3	2	309 (98)	28	500 (2) ↓	1	483 (85) ↓	2	265 (72) ↑	31	499 (4) ↓	1	423 (96) ↓	18
LIRCMOP4	2	298 (82)	31	500 (0) ↓	0	485 (84) ↓	1	233 (63) ↑	31	489 (61) ↓	1	415 (101) ↓	16
LIRCMOP5-12	2	1 (0)	31	1 (0) ≈	31	1 (0) ≈	31	1 (0) ≈	31	1 (0) ≈	31	1 (0) ≈	31
LIRCMOP13-14	3	1 (0)	31	1 (0) ≈	31	1 (0) ≈	31	1 (0) ≈	31	1 (0) ≈	31	1 (0) ≈	31
DASCMP1	2	7 (5)	31	7 (5) ≈	31	6 (4) ≈	31	4 (5) ≈	31	5 (4) ≈	31	7 (5) ≈	31
DASCMP2	2	7 (5)	31	7 (5) ≈	31	6 (4) ≈	31	7 (7) ≈	31	6 (5) ≈	31	7 (5) ≈	31
DASCMP3	2	7 (5)	31	7 (5) ≈	31	6 (4) ≈	31	6 (4) ≈	31	6 (5) ≈	31	7 (5) ≈	31
DASCMP4	2	162 (38)	31	220 (74) ↓	31	499 (6) ↓	2	249 (128) ↓	25	500 (0) ↓	0	500 (0) ↓	0
DASCMP5	2	172 (68)	30	211 (66) ↓	31	492 (29) ↓	5	272 (141) ↓	23	500 (0) ↓	0	500 (0) ↓	0
DASCMP6	2	189 (104)	29	198 (67) ↓	31	497 (11) ↓	2	244 (131) ↓	25	500 (0) ↓	0	500 (0) ↓	0
DASCMP7	3	157 (39)	31	254 (112) ↓	30	492 (27) ↓	4	197 (84) ↓	29	500 (0) ↓	0	500 (0) ↓	0
DASCMP8	3	162 (71)	30	235 (105) ↓	30	488 (49) ↓	4	218 (113) ↓	27	500 (0) ↓	0	500 (0) ↓	0
DASCMP9	3	240 (66)	31	498 (9) ↓	1	500 (0) ↓	0	500 (0) ↓	0	500 (0) ↓	0	500 (0) ↓	0
FFE ($\uparrow/\approx/\downarrow$)				1/19/19		3/16/20		5/22/15		0/13/26		0/15/22	

E. Component-wise performance assessment of PSCMOEA

To systematically observe the benefits offered by the key internal components of PSCMOEA, three variants of PSCMOEA are constructed. Of these, the first two are constructed by modifying the probabilistic selection ranking strategy. The third variant is constructed to show the effectiveness of the search switching mechanism. The brief outline of the variants is presented below.

- V1: In the first variant, instead of ranking the solutions along the RVs for ‘Constrained’ search according to Eq. 3, a lexicographic ranking strategy is undertaken which essentially follows a feasibility-first (FF) principle. According to it, the solution cluster along a RV is divided into feasible and infeasible set based on their corresponding PoF score as represented in Eq. 4. The threshold value of PoF for each individual constraint distribution is set as ≥ 0.99 to treat a solution as feasible. Note that each constraint has to satisfy the threshold PoF value. Among the feasible block, the solutions are sorted based on their PD_F score, as represented in Eq. 6. For the infeasible block, the solutions are first sorted based on their PoF product score (higher the better). If there are solutions with the same PoF product score, then they are further sorted based on their number of satisfied (NS) constraints (higher, the better). If there remain solutions with the same NS counts, they are sorted based on their mean CV predictions (smaller, the better). Finally the

sorted feasible block of solutions are placed above the sorted infeasible ones. Note that the same ranking strategy is followed during the infill identification stage as well when the archive is fully infeasible. All other components are kept as same as PSCMOEA.

- V2: The second variant contains a simpler ranking strategy. For a block of candidate solutions along a RV, numerical score of PoF and PD_F is calculated for each individual candidate. For each solution, the product of PoF and PD_F scores ($PoF \times PD_F$) is considered as the overall score. The scores for all solutions along the RV is calculated, and they are sorted in descending order. Same principle is undertaken during the infill identification stage by keeping all the remaining components as same as PSCMOEA.
- V3: This variant is constructed simply by taking off the search switching mechanism (Sec. III-E) from PSCMOEA while keeping all other components the same.

All variants (including PSCMOEA), are run on each test problem 31 times. The performance is compared pairwise with each other variant in terms of IGD, IGD^+ , HV and FFE, and the results are summarized in Table III. For example, MW contains a total of 14 problems. This means total 42 ($= 14 \times 3$) comparisons are performed for PSCMOEA against the three variants. More detailed statistics including mean IGD, IGD^+ , HV, FFE, ST are included in the SOM Section V.

As mentioned above, the first two variants (V1 and V2) are constructed to observe the effectiveness of the proposed

TABLE II

MEAN IGD (STD.) ACROSS 31 RUNS OBTAINED BY PSCMOEA AND PEER-ALGORITHMS. $\uparrow/\approx/\downarrow$ SYMBOLS DENOTE IF THE COMPARED ALGORITHM IS SIGNIFICANTLY BETTER/ EQUIVALENT/ WORSE THAN PSCMOEA. THE GRAY SHADE HIGHLIGHTS THE BEST MEAN PERFORMANCE. THE CUMULATIVE $\uparrow/\approx/\downarrow$ STATISTICS OF EACH ALGORITHM AGAINST PSCMOEA IN TERMS OF IGD, IGD⁺ AND HV ARE PROVIDED AT THE BOTTOM.

Prob.	M	Algorithms					
		PSCMOEA	KMGSAEA	ASA-MOEA/D	KTS	MultiObjectiveEGO	SILE
Test1	2	2.08e-01 (1.10e-01)	8.51e-01 (0.00e+00) ↓	2.08e-01 (3.53e-01) ↑	6.59e-01 (3.60e-01) ↓	5.96e-01 (2.15e-01) ↓	1.68e-01 (3.40e-01) ↑
Test2	2	1.21e-01 (1.93e-01)	6.58e-01 (2.48e-01) ↓	1.59e-01 (2.35e-01) ≈	6.34e-01 (2.42e-01) ↓	5.88e-01 (2.24e-01) ↓	4.74e-01 (3.57e-01) ↓
MW1	2	2.50e-01 (3.80e-01)	6.80e-01 (2.89e-01) ↓	9.11e-01 (1.27e-01) ↓	8.32e-01 (2.98e-01) ↓	9.61e-01 (4.51e-16) ↓	9.09e-01 (1.37e-01) ↓
MW2	2	1.58e-01 (1.30e-01)	4.96e-01 (3.74e-01) ↓	5.45e-01 (2.29e-01) ↓	1.82e-01 (2.16e-01) ≈	4.20e-01 (1.74e-01) ↓	2.31e-01 (1.49e-01) ↓
MW3	2	2.05e-02 (7.83e-03)	4.53e-02 (1.02e-02) ↓	9.64e-02 (3.41e-02) ↓	1.74e-02 (2.50e-03) ≈	5.97e-01 (1.80e-02) ↓	8.42e-02 (8.39e-02) ↓
MW4	3	4.99e-01 (3.98e-01)	7.74e-01 (6.46e-01) ≈	1.46e+00 (1.57e-01) ↓	7.08e-01 (5.53e-01) ↓	1.52e+00 (6.77e-16) ↓	1.44e+00 (2.44e-01) ↓
MW5	2	2.67e-01 (1.48e-01)	4.50e-01 (2.51e-01) ↓	7.97e-01 (3.05e-01) ↓	6.82e-01 (3.30e-01) ↓	1.02e+00 (4.51e-16) ↓	9.60e-01 (1.28e-01) ↓
MW6	2	5.44e-01 (2.17e-01)	7.92e-01 (3.30e-01) ↓	1.01e+00 (3.58e-01) ↓	6.83e-01 (3.07e-01) ↓	8.60e-01 (3.03e-01) ↓	8.12e-01 (3.05e-01) ↓
MW7	2	2.87e-02 (1.15e-02)	4.83e-02 (1.24e-02) ↓	3.97e-02 (1.31e-02) ↓	2.47e-02 (5.88e-03) ≈	7.38e-01 (1.80e-02) ↓	5.23e-02 (7.04e-02) ↓
MW8	3	3.01e-01 (4.06e-01)	4.82e-01 (2.34e-01) ↓	1.62e+00 (2.74e-01) ↓	4.43e-01 (4.21e-01) ↓	1.49e+00 (6.10e-01) ↓	9.00e-01 (6.71e-01) ↓
MW9	2	2.87e-01 (2.36e-01)	6.30e-01 (3.32e-01) ↓	1.19e+00 (1.63e-01) ↓	6.19e-01 (4.34e-01) ↓	1.23e+00 (2.26e-16) ↓	1.23e+00 (2.26e-16) ↓
MW10	2	6.46e-01 (2.11e-01)	6.79e-01 (2.13e-01) ≈	6.79e-01 (2.13e-01) ≈	8.07e-01 (1.13e-16) ↓	8.07e-01 (1.13e-16) ↓	8.07e-01 (1.13e-16) ↓
MW11	2	2.86e-01 (2.02e-01)	1.38e-01 (1.30e-01) ↑	1.74e-01 (1.38e-01) ≈	1.11e-01 (1.52e-01) ↑	7.88e-01 (5.56e-02) ↓	6.07e-01 (2.14e-01) ↓
MW12	2	1.80e-01 (1.99e-01)	4.60e-01 (4.00e-01) ↓	1.21e+00 (4.38e-01) ↓	4.65e-01 (4.10e-01) ↓	1.49e+00 (2.26e-16) ↓	1.06e+00 (3.66e-01) ↓
MW13	2	4.88e-01 (2.75e-01)	4.85e-01 (5.21e-01) ≈	2.39e+00 (1.15e+00) ↓	7.78e-01 (7.34e-01) ↓	3.58e+00 (1.33e+00) ↓	1.28e+00 (1.23e+00) ↓
MW14	3	7.74e-02 (2.62e-02)	5.63e-01 (1.81e-01) ↓	7.34e-01 (2.19e-01) ↓	1.91e-01 (1.47e-01) ↓	8.95e-01 (1.93e-01) ↓	7.48e-01 (1.72e-01) ↓
LIRCMOP1	2	5.41e-01 (1.59e-01)	8.55e-01 (1.50e-01) ↓	8.86e-01 (1.55e-01) ↓	4.69e-01 (3.25e-01) ≈	9.01e-01 (1.35e-01) ↓	4.40e-01 (2.77e-01) ↑
LIRCMOP2	2	5.96e-01 (1.64e-01)	9.10e-01 (7.19e-02) ↓	8.75e-01 (1.83e-01) ↓	3.00e-01 (2.79e-01) ↑	8.59e-01 (1.97e-01) ↓	4.30e-01 (2.82e-01) ↑
LIRCMOP3	2	5.52e-01 (1.75e-01)	9.15e-01 (1.94e-02) ↓	8.92e-01 (1.07e-01) ↓	3.12e-01 (1.52e-01) ↑	9.03e-01 (9.08e-02) ↓	6.07e-01 (2.73e-01) ≈
LIRCMOP4	2	5.98e-01 (1.61e-01)	9.09e-01 (3.39e-16) ↓	8.91e-01 (9.89e-02) ↓	2.70e-01 (1.28e-01) ↑	8.91e-01 (9.94e-02) ↓	6.11e-01 (2.95e-01) ≈
LIRCMOP5	2	1.21e-01 (2.23e-01)	3.05e-02 (6.83e-03) ↑	1.02e+00 (4.65e-01) ↓	2.77e-02 (9.07e-03) ↑	2.86e+00 (4.09e-01) ↓	6.94e-01 (2.21e-01) ↓
LIRCMOP6	2	3.94e-02 (3.93e-02)	3.47e-02 (4.57e-03) ≈	9.89e-01 (5.04e-01) ↓	4.40e-02 (1.93e-02) ↓	2.88e+00 (5.51e-01) ↓	4.23e-01 (8.17e-02) ↓
LIRCMOP7	2	2.26e-01 (1.09e-01)	2.47e-01 (8.30e-02) ≈	4.17e-01 (3.81e-01) ↓	2.07e-01 (4.56e-02) ≈	1.68e+00 (4.68e-01) ↓	3.12e-01 (1.65e-01) ↓
LIRCMOP8	2	1.86e-01 (7.25e-02)	2.41e-01 (1.79e-01) ≈	4.17e-01 (3.81e-01) ↓	1.89e-01 (5.56e-02) ≈	1.82e+00 (4.43e-01) ↓	4.37e-01 (1.87e-01) ↓
LIRCMOP9	2	1.36e-01 (5.49e-02)	2.98e-01 (6.36e-02) ↓	3.40e-01 (1.01e-01) ↓	1.74e-01 (6.09e-02) ↓	1.07e+00 (2.24e-01) ↓	3.63e-01 (5.14e-02) ↓
LIRCMOP10	2	1.07e-01 (7.34e-02)	3.17e-01 (8.69e-02) ↓	3.46e-01 (1.04e-01) ↓	6.78e-02 (2.59e-02) ↑	8.72e-01 (1.43e-01) ↓	3.68e-01 (4.54e-02) ↓
LIRCMOP11	2	1.09e-01 (5.63e-02)	2.19e-01 (6.39e-02) ↓	3.20e-01 (9.23e-02) ↓	1.10e-01 (6.02e-02) ≈	8.14e-01 (1.05e-01) ↓	3.13e-01 (4.81e-02) ↓
LIRCMOP12	2	1.12e-01 (3.02e-02)	2.25e-01 (7.52e-02) ↓	2.11e-01 (7.62e-02) ↓	1.52e-01 (4.18e-02) ↓	8.90e-01 (1.64e-01) ↓	2.93e-01 (4.94e-02) ↓
LIRCMOP13	3	1.41e-01 (1.96e-01)	6.05e-01 (1.93e-01) ↓	1.09e+00 (2.02e-01) ↓	6.79e-02 (1.16e-02) ≈	1.22e+00 (1.15e-01) ↓	1.12e+00 (1.17e-01) ↓
LIRCMOP14	3	2.05e-01 (2.43e-01)	5.98e-01 (1.50e-01) ↓	1.07e+00 (1.50e-01) ↓	1.75e-01 (3.75e-02) ↓	1.15e+00 (1.18e-01) ↓	1.08e+00 (1.33e-01) ↓
DASCMP1	2	2.06e-02 (1.07e-02)	6.28e-01 (1.40e-01) ↓	6.36e-02 (2.14e-02) ↓	6.14e-02 (7.43e-02) ≈	4.77e-01 (1.18e-01) ↓	4.07e-01 (1.59e-01) ↓
DASCMP2	2	1.06e-01 (2.09e-02)	4.96e-01 (6.06e-02) ↓	1.34e-01 (2.12e-02) ↓	9.30e-02 (1.03e-02) ↑	4.24e-01 (5.79e-02) ↓	3.51e-01 (6.48e-02) ↓
DASCMP3	2	2.77e-01 (6.89e-02)	5.95e-01 (8.33e-02) ↓	2.61e-01 (5.83e-02) ≈	4.44e-01 (7.89e-02) ↓	5.12e-01 (5.91e-02) ↓	4.18e-01 (8.24e-02) ↓
DASCMP4	2	4.43e-01 (3.39e-01)	4.05e-01 (2.60e-01) ≈	1.20e+00 (8.16e-02) ↓	4.79e-01 (3.97e-01) ≈	1.22e+00 (0.00e+00) ↓	1.22e+00 (0.00e+00) ↓
DASCMP5	2	4.78e-01 (2.93e-01)	4.49e-01 (1.81e-01) ≈	1.26e+00 (1.31e-01) ↓	6.63e-01 (4.23e-01) ≈	1.30e+00 (6.77e-16) ↓	1.30e+00 (6.77e-16) ↓
DASCMP6	2	5.88e-01 (2.81e-01)	5.84e-01 (1.76e-01) ≈	1.32e+00 (6.69e-02) ↓	7.01e-01 (3.52e-01) ≈	1.34e+00 (9.03e-16) ↓	1.34e+00 (9.03e-16) ↓
DASCMP7	3	3.85e-01 (3.58e-01)	3.29e-01 (2.44e-01) ≈	1.21e+00 (1.38e-01) ↓	2.58e-01 (3.42e-01) ≈	1.26e+00 (4.51e-16) ↓	1.26e+00 (4.51e-16) ↓
DASCMP8	3	3.67e-01 (3.79e-01)	4.10e-01 (2.76e-01) ≈	1.44e+00 (1.78e-01) ↓	3.96e-01 (4.77e-01) ≈	1.50e+00 (6.77e-16) ↓	1.50e+00 (6.77e-16) ↓
DASCMP9	3	4.60e-01 (1.61e-01)	1.26e+00 (1.80e-02) ↓	1.26e+00 (4.51e-16) ↓	1.26e+00 (4.51e-16) ↓	1.26e+00 (4.51e-16) ↓	1.26e+00 (4.51e-16) ↓
IGD ($\uparrow/\approx/\downarrow$)			2/11/26	1/3/35	7/14/18	0/0/39	3/2/34
IGD ⁺ ($\uparrow/\approx/\downarrow$)			2/11/26	1/2/36	7/13/19	0/0/39	3/2/34
HV ($\uparrow/\approx/\downarrow$)			1/9/29	1/4/34	5/17/17	0/0/39	4/3/32

integrated ranking method using PoF and PD metrics following Eq. 3. From the given statistics in Table III, one can clearly observe that both $V1$ and $V2$ perform worse than PSCMOEA in terms of all metrics. This is because the ranking approach used in $V1$ is biased more on PoF metric than PDF_F , where PDF_F is only applied for ranking the solutions with high likelihood of feasibility (by considering $PoF \geq 0.99$ as feasible). For the feasibility-hard problems, such an approach drives the search very slowly in the original objective space as it ignores the dominance relationship among the candidates in the predicted fitness landscape. On the other hand, the second variant $V2$ equally weighs them by simply multiplying PoF and PDF_F to order the candidate solutions. For feasibility-hard problems, such approach can expedite the convergence rate for the instances where a good positive correlation exists between objective optimization and CV minimization direction. This is because PDF_F can drive the search towards optimal objective location when PoF is very low. For this case, the win statistics improves for MW test suites compared to $V1$ as indicated in Table III. However, the weakness of such approach can be observed on the problems where the correlation is negative between

objective optimization and CV minimization direction, for example on Test1 where it could not identify any feasible solution across 31 runs. Moreover, there is no means to fully rank the candidates with 0 PoF. In addition, the information regarding the CV distribution with low PoF is ignored in the approach, hence resulting in performance degradation for such scenarios. From these results, it could be inferred that PSCMOEA ranking strategy can more efficiently deal with a range of scenarios.

As for $V3$, it is constructed to observe the influence of the search switching mechanism on the overall performance of PSCMOEA. From Table III, some interesting performance trends can be observed. The switching mechanism clearly improves the performance significantly in terms of all metrics on the MW test suite while a slight performance degradation can be noticed on DASCMP series. This is because most of the problems in MW test suite have a positive correlation between UPF and CPF search direction, therefore switching to unconstrained search occasionally expedites the search. The opposite situation happens in DASCMP series for not having good correlation. The switching behavior can be observed

TABLE III
THE CUMULATIVE RESULTS OF PAIRWISE SIGNIFICANCE TESTS ($\uparrow/\approx/\downarrow$)
BETWEEN ALL PAIRS OF VARIANTS, INCLUDING PSCMOEA.

Metric	Problems	M	Algorithms			
			PSCMOEA	V1	V2	V3
IGD	Test	2	3/3/0	1/3/2	0/1/5	3/3/0
	MW (2 & 3)	(2 & 3)	23/16/3	7/11/24	13/13/16	11/20/11
	DASCMP (2 & 3)	(2 & 3)	16/9/2	2/7/18	2/7/18	18/9/0
IGD*	Test	2	3/3/0	1/3/2	0/1/5	3/3/0
	MW (2 & 3)	(2 & 3)	24/16/2	6/10/26	13/13/16	13/17/12
	DASCMP (2 & 3)	(2 & 3)	16/8/3	4/5/18	2/7/18	18/8/1
HV	Test	2	3/3/0	1/3/2	0/1/5	3/3/0
	MW (2 & 3)	(2 & 3)	23/17/2	6/11/25	13/13/16	11/21/10
	DASCMP (2 & 3)	(2 & 3)	16/9/2	4/5/18	2/8/17	16/10/1
FFE	Test	2	1/4/1	1/4/1	3/0/3	1/4/1
	MW (2 & 3)	(2 & 3)	16/22/4	7/12/23	14/18/10	10/22/10
	DASCMP (2 & 3)	(2 & 3)	14/13/0	0/14/13	1/14/12	12/13/2

from the bar chart in Fig. 9 which illustrates the number of times PSCMOEA switched to unconstrained search during the median-run (out of 31 runs in terms of IGD metric) for the feasibility-hard problems. From the figure, it is clear that there are higher number of iterations conducted in unconstrained mode for MW series than those from other test suites. In particular for MW10, the number is highest because the feasibility rate is very low and the relationship between CPF and UPF remains positive during the search. On the other hand, in Test1 and Test2, no unconstrained search was initiated due to the consistent negative correlation. LIRCMOP and DASCMP encounter some interactions in unconstrained mode, however, it did not improve the solutions, PSCMOEA did not waste further resources on unconstrained search. Overall the number of iterations that switch to unconstrained search are typically very low, indicating that PSCMOEA is able to swiftly identify if the unconstrained search is of any benefit, and switch back to the default constrained search if not.

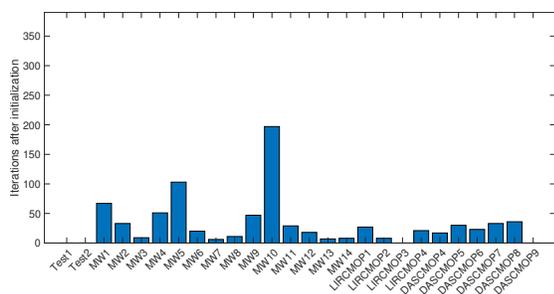


Fig. 9. Number of times PSCMOEA switches to unconstrained search in the median-run (out of 31 trials) in terms of IGD metric on (a) MW series, Test1 - Test2, LIRCMOP1 - LIRCMOP4 and DASCMP4 - DASCMP9.

V. CONCLUDING REMARKS

In this study, some of the existing works have been reviewed that tackle ECMOPs, and highlighted how the relative position of UPF and CPF and model uncertainties can impact their performance. These observations along with the lack of steady-state methods to solve ECMOPs, formed the key motivations of the study. To address the identified gaps, the paper develops an efficient approach based on probabilistic selection, referred to as PSCMOEA, for solving ECMOPs on a low evaluation

budget. The key differentiating feature of PSCMOEA is its careful consideration of model uncertainties and the status of evaluated solutions in terms of feasibility, convergence, and diversity in its components. These considerations feed into multiple components, such as environmental selection using probabilistic constrained dominance (*PCD*), adaptive setting of normalization bounds to effectively explore the predicted space and an adaptive search switching mechanism to expedite the convergence when the UPF and CPF search directions are similar. Numerical experiments and benchmarking with five state-of-the-art algorithms were conducted on an extensive set of problems to demonstrate the efficacy of PSCMOEA. Moreover, ablation studies were conducted to highlight and verify the effectiveness of some of the key components.

Some of the potential directions for future work include extension to many-objective problems (i.e., those with more than 3 objectives) and development of a generational version of the proposed approach.

REFERENCES

- [1] 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, 2022.
- [2] R. G. Regis, "A two-phase surrogate approach for high-dimensional constrained discrete multi-objective optimization," in *GECCO Companion*, 2021, pp. 1870–1878.
- [3] G. G. Wang and S. Shan, "Review of metamodeling techniques in support of engineering design optimization," in *IDETC-CIE*, vol. 4255, 2006, pp. 415–426.
- [4] J. Martínez-Frutos and D. Herrero-Pérez, "Kriging-based infill sampling criterion for constraint handling in multi-objective optimization," *J. Glob. Optim.*, vol. 64, pp. 97–115, 2016.
- [5] K. H. Rahi, H. K. Singh, and T. Ray, "A steady-state algorithm for solving expensive multiobjective optimization problems with nonparallelizable evaluations," *IEEE Trans. Evol. Comput.*, vol. 27, no. 5, pp. 1544–1558, 2023.
- [6] A. Mazumdar, T. Chugh, J. Hakanen, and K. Miettinen, "Probabilistic selection approaches in decomposition-based evolutionary algorithms for offline data-driven multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1182–1191, 2022.
- [7] T. Chugh, K. Sindhya, J. Hakanen, and K. Miettinen, "A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms," *Soft Comput.*, vol. 23, pp. 3137–3166, 2019.
- [8] K. Deb, R. Hussein, P. C. Roy, and G. Toscano-Pulido, "A taxonomy for metamodeling frameworks for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 104–116, 2018.
- [9] R. Hussein and K. Deb, "A generative Kriging surrogate model for constrained and unconstrained multi-objective optimization," in *GECCO*, 2016, pp. 573–580.
- [10] 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.*, 2023.
- [11] 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.*, 2023.
- [12] Z. Yang, H. Qiu, L. Gao, L. Chen, and J. Liu, "Surrogate-assisted MOEA/D for expensive constrained multi-objective optimization," *Inf. Sci.*, vol. 639, p. 119016, 2023.
- [13] S. Yin, R. Wang, J. Zhang, X. Liu, and Y. Wang, "Fast surrogate-assisted constrained multiobjective optimization for analog circuit sizing via self-adaptive incremental learning," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 42, no. 7, pp. 2080–2093, 2023.
- [14] H. Wu, Q. Chen, Y. Jin, J. Ding, and T. Chai, "A surrogate-assisted expensive constrained multi-objective optimization algorithm based on adaptive switching of acquisition functions," *IEEE Trans. Emerg. Top. Comput. Intell.*, 2024.
- [15] C. He, Y. Zhang, D. Gong, and X. Ji, "A review of surrogate-assisted evolutionary algorithms for expensive optimization problems," *Expert Syst. Appl.*, p. 119495, 2023.

- [16] J. Zhou, Y. Zhang, J. Zheng, and M. Li, "Domination-based selection and shift-based density estimation for constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 27, no. 4, pp. 993–1004, 2023.
- [17] J. Yuan, H.-L. Liu, Y.-S. Ong, and Z. He, "Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 2, pp. 379–391, 2021.
- [18] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," *Constr.-handling Evol. Optim.*, pp. 145–165, 2009.
- [19] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with an archive and gradient-based mutation," in *CEC*. IEEE, 2010, pp. 1–9.
- [20] Z.-Z. Liu, Y. Qin, W. Song, J. Zhang, and K. Li, "Multiobjective-based constraint-handling technique for evolutionary constrained multi-objective optimization: A new perspective," *IEEE Trans. Evol. Comput.*, vol. 27, no. 5, pp. 1370–1384, 2023.
- [21] K. Qiao, J. Liang, K. Yu, M. Wang, B. Qu, C. Yue, and Y. Guo, "A self-adaptive evolutionary multi-task based constrained multi-objective evolutionary algorithm," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 7, no. 4, pp. 1098–1112, 2023.
- [22] Z.-Z. Liu, F. Wu, J. Liu, Y. Qin, and K. Li, "Constrained multiobjective optimization with escape and expansion forces," *IEEE Trans. Evol. Comput.*, 2023.
- [23] J. Zou, R. Sun, Y. Liu, Y. Hu, S. Yang, J. Zheng, and K. Li, "A multipopulation evolutionary algorithm using new cooperative mechanism for solving multiobjective problems with multiconstraint," *IEEE Trans. Evol. Comput.*, vol. 28, no. 1, pp. 267–280, 2024.
- [24] M. Ming, R. Wang, H. Ishibuchi, and T. Zhang, "A novel dual-stage dual-population evolutionary algorithm for constrained multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 26, no. 5, pp. 1129–1143, 2021.
- [25] M. Zuo, D. Gong, Y. Wang, X. Ye, B. Zeng, and F. Meng, "Process knowledge-guided autonomous evolutionary optimization for constrained multiobjective problems," *IEEE Trans. Evol. Comput.*, vol. 28, no. 1, pp. 193–207, 2024.
- [26] 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, 2023.
- [27] K. S. Bhattacharjee, H. K. Singh, and T. Ray, "Multi-objective optimization with multiple spatially distributed surrogates," *J. Mech. Des.*, vol. 138, no. 9, p. 091401, 2016.
- [28] 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, 2019.
- [29] R. de Winter, P. Bronkhorst, B. van Stein, and T. Bäck, "Constrained multi-objective optimization with a limited budget of function evaluations," *Memet. Comput.*, vol. 14, no. 2, pp. 151–164, 2022.
- [30] J. Blank and K. Deb, "Constrained bi-objective surrogate-assisted optimization of problems with heterogeneous evaluation times: Expensive objectives and inexpensive constraints," in *EMO*. Springer, 2021, pp. 257–269.
- [31] K. Shang, H. Ishibuchi, L. He, and L. M. Pang, "A survey on the hypervolume indicator in evolutionary multiobjective optimization," *IEEE Trans. on Evol. Comput.*, vol. 25, no. 1, pp. 1–20, 2020.
- [32] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Glob. Optim.*, vol. 13, pp. 455–492, 1998.
- [33] A. Forrester, A. Sobester, and A. Keane, *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008.
- [34] P. Singh, M. Rossi, I. Couckuyt, D. Deschrijver, H. Rogier, and T. Dhaene, "Constrained multi-objective antenna design optimization using surrogates," *Int. J. Numer. Model.: Electron. Netw. Devices Fields*, vol. 30, no. 6, p. e2248, 2017.
- [35] D. Han and J. Zheng, "A Kriging model-based expensive multiobjective optimization algorithm using R2 indicator of expectation improvement," *Math. Prob. Engg.*, vol. 2020, 2020.
- [36] 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, 2017.
- [37] Z. Han, F. Liu, C. Xu, K. Zhang, and Q. Zhang, "Efficient multi-objective evolutionary algorithm for constrained global optimization of expensive functions," in *CEC*. IEEE, 2019, pp. 2026–2033.
- [38] A. Tran, M. Eldred, S. McCann, and Y. Wang, "srMO-BO-3GP: A sequential regularized multi-objective Bayesian optimization for constrained design applications using an uncertain Pareto classifier," *J. Mech. Des.*, vol. 144, no. 3, p. 031705, 2022.
- [39] S. Liu, Z. Wang, Q. Lin, J. Li, and K. C. Tan, "Learning-aided evolutionary search and selection for scaling-up constrained multiobjective optimization," *IEEE Trans. on Evol. Comput.*, 2024.
- [40] 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, 2019.
- [41] K. H. Rahi, H. K. Singh, and T. Ray, "Partial evaluation strategies for expensive evolutionary constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1103–1117, 2021.
- [42] 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, 2017.
- [43] L. He, H. Ishibuchi, A. Trivedi, H. Wang, Y. Nan, and D. Srinivasan, "A survey of normalization methods in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 25, no. 6, pp. 1028–1048, 2021.
- [44] B. Wang, H. K. Singh, and T. Ray, "Investigating normalization bounds for hypervolume-based infill criterion for expensive multiobjective optimization," in *EMO*. Springer, 2021, pp. 519–530.
- [45] M. Elarbi, S. Bechikh, C. A. C. Coello, M. Makhlof, and L. B. Said, "Approximating complex Pareto fronts with predefined normal-boundary intersection directions," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 809–823, 2019.
- [46] C. Gong, Y. Nan, L. M. Pang, Q. Zhang, and H. Ishibuchi, "Effects of including optimal solutions into initial population on evolutionary multiobjective optimization," in *GECCO*, 2023, pp. 661–669.
- [47] 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, 2002.
- [48] 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.
- [49] E. J. Hughes, "Evolutionary multi-objective ranking with uncertainty and noise," in *EMO*. Springer, 2001, pp. 329–343.
- [50] J. Sun, M. Kong, and S. Pal, "The modified-half-normal distribution: Properties and an efficient sampling scheme," *Commun. Stat. - Theory Methods*, vol. 52, no. 5, pp. 1591–1613, 2023.
- [51] A. W. Palmer, A. J. Hill, and S. J. Scheding, "Methods for stochastic collection and replenishment (SCAR) optimisation for persistent autonomy," *Robot. Auton. Syst.*, vol. 87, pp. 51–65, 2017.
- [52] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty adjustable and scalable constrained multiobjective test problem toolkit," *Evol. comput.*, vol. 28, no. 3, pp. 339–378, 2020.
- [53] H. K. Singh, K. S. Bhattacharjee, and T. Ray, "Distance-based subset selection for benchmarking in evolutionary multi/many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 904–912, 2018.
- [54] M. G. Kendall, "Rank correlation methods." 1948.
- [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. 12491–12510, 2019.
- [56] C. A. C. Coello, G. B. Lamont, D. A. Van Veldhuizen *et al.*, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, vol. 5.
- [57] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Modified distance calculation in generational distance and inverted generational distance," in *EMO*. Springer, 2015, pp. 110–125.
- [58] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.
- [59] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Math. Prog.*, vol. 91, no. 2, pp. 201–213, 2002.