

Exploiting Pareto-optimal Solutions for Knowledge-Augmented Design and Decision-Making

Ahmer Khan¹, Frank Haubold², Simon Xu², and Kalyanmoy Deb¹

¹Michigan State University, East Lansing, USA

{khanahm2, kdeb}@msu.edu

²General Motors, Detroit, USA

{frank.haubold, simon.xu}@gm.com

COIN Report 2026011

Abstract

Many industrial design and decision-making activities increasingly involve multiple conflicting objectives, such as simultaneous consideration of cost and quality. In recent years, numerous efficient multi-objective optimization and decision-making algorithms have been developed to identify sets of Pareto-optimal (PO) solutions for such problems. Beyond their optimality, PO solutions inherently encode trade-off information among conflicting objectives, thereby offering decision-makers a diverse set of efficient alternatives to choose from. However, it is often observed that certain regions of the Pareto front (PF) contain no solutions. These regions may correspond to solutions with desired trade-off, making it crucial to understand why solutions are absent in those areas. Prior studies suggest that PO solution sets frequently exhibit underlying variable relationships, arising from the satisfaction of unique optimality conditions specific to the problem. By analyzing these properties, the creation of new solutions in such regions can provide valuable insights into the reasons for their absence, revealing structural constraints and relationships embedded in the problem. To address this, in this paper, we propose a systematic procedure called Knowledge-Augmented Innovative Design Optimization (KAIDO). KAIDO explicitly extracts and utilizes variable relationships from the PO set to explain the absence of solutions in certain regions of the PF. In contrast, we also examine a recently-proposed machine learning (ML) approach, which implicitly captures variable patterns through network structures but does not reveal explicit relationships. The study compares both techniques in the context of understanding PF gaps on a number of problems including a real-world engineering design problem, discussing their respective strengths and limitations.

1 Introduction

Recently, many industrial design and manufacturing optimization problems consider multiple conflicting objectives [4, 25, 3, 29], such as cost and quality of the solution. Optimization of such problems results in a set of Pareto-optimal (PO) solutions presenting an optimal trade-off between the considered conflicting objectives [10]. Ideally, the extremes of the PO set constitute individual optimal solution of each objective, but the intermediate PO solutions compromise multiple objectives with trade-offs [8]. Thus, the PO set provides decision-makers (DMs) with several alternate solutions to choose from [10]. While the decision-maker (DM) choose a solution from the PO set, it is important for the PO set to be diverse and representative of all possible trade-off regions. We do not present any new decision-making procedure in this paper, but discuss two key post-optimal analysis tasks, which can provide additional and more comprehensive understanding of the PO solutions before a decision-making task can be performed or vital knowledge about properties of PO solutions can be obtained.

Knowledge-based design concepts are suggested in the literature [16, 18, 23, 20, 27]. But they are not generic and specific to certain esoteric design problems. In addition to providing expert knowledge in improving designs, knowledge can be deciphered from multi-objective PO solutions and systematically utilized to obtain a better understanding of possible optimal solutions. Despite a few Pareto solution

analysis procedures including identifying gaps in PF [7, 23, 20, 1, 19, 15], no systematic approach to provide reasons for gaps in PO sets for the purpose of design and decision-making purposes exist.

The first task in our proposed systematic knowledge-augmented design approach is to discover knowledge on key variable interactions, which can provide the users with precise information about variable clusters and intra-cluster variable relationships that constitute the high-performing PO solutions. Because the PO solutions are optimal, and not an set of solutions, they are expected to possess certain variable relationships that make them fall on the PF. Since these relationships are special only to PO solutions, once discovered, they can provide valuable knowledge about properties of optimal solutions. Although the concept of deciphering such knowledge common to PO solutions was termed as an act of ‘innovization’ – innovation through optimization in earlier studies [11, 9, 12], in this paper, we prescribe a systematic procedure of obtaining variable relationships by first performing a correlation analysis, leading to a clustering of variables and then creating polynomial relationships among intra-cluster variables. The derived relationships can be useful to users in many different ways. They can simply provide knowledge of variable interactions meaning how an increase in one of the variables must be negotiated with increase or decrease in another variables to stay Pareto-optimal. Such information may prove vital for a better understanding and in choosing a single preferred PO solution. The variable relationships can be used to create new non-dominated solutions in PF regions where no solutions were found by the multi-objective optimization run earlier - a topic which we describe next.

In practice, multi-objective optimization runs often end up discovering PO sets with gaps (regions having no solutions) in between. These gaps could represent trade-off regions of interest to DMs. Gaps in the PO set could be the result of one of the following reasons:

1. **Premature Convergence:** The optimization algorithm failed to discover these particular solutions
2. **Infeasibility:** The region is infeasible
3. **Domination:** The solutions in the region were dominated by other solutions in the PO set

In the event of gaps observed in a PO set, an appropriate reason for the gaps must be properly understood and, if possible, a follow-up post-optimal study must be executed to fill the gaps with new and non-dominated solutions. In this paper, we utilize the variable relationships identified from PO solutions through a machine learning approach to discover the reasons for observing a gap and also propose a way to fill the gaps with new solutions.

In this paper, we propose a systematic data analysis – knowledge-augmented innovative design optimization (KAIDO) – approach to automatically identify mathematical relationships among variables of the PO set. Once discovered, we use the information to formulate a reduced optimization problem utilizing the relationships and employ a focused EMO algorithm (such as, the reference-point based NSGA-II [14]) to find PO solutions in the gaps. We compare our proposed approach to an existing machine learning (ML) approach and present results on three different test problems and a real-life engineering problem. The advantages and disadvantages of both approaches are highlighted in this paper.

In the remainder of the paper, Section 2 details our proposed knowledge discovery approach in detail by classifying variables into four groups for establishing different types of variable relationships. Then, in Section 3, we detail different conditions with which gaps in a PF can appear from a run of an evolutionary multi-objective optimization algorithm. We then discuss how our proposed knowledge discovery approach can be used to construct a reduced focused optimization problem to discover solutions in apparent gaps in the EMO-obtained PF. Section 4 discusses briefly a previous alternate approach for producing solutions in gaps of a PF. Thereafter, Section 5 presents three test problems and a real-world engineering design problem used in this study and Section 6 discusses results from our proposed KAIDO approach and compares with the existing ML approach. Conclusions are drawn in Section 7.

2 Proposed Knowledge Discovery Approach from PO Solutions

We consider the following design optimization problem in which there are n design variables \mathbf{x} , M objectives (\mathbf{f}), J inequality constraints (\mathbf{g}):

$$\begin{aligned} & \text{Minimize}_{\mathbf{x}} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\}, \\ & \text{subject to} \quad g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, J, \\ & \quad \quad \quad x_i^{(L)} \leq x_i \leq x_i^{(U)}, \quad i = 1, 2, \dots, n. \end{aligned} \quad (1)$$

Given a set of Pareto Optimal (PO) solutions, obtained by an EMO algorithm, four different variable characteristics common to the PO set are extracted using an innovative data analysis procedure. We describe our approach in this section.

2.1 Identification of Fixed Variables

Among all design variables, it is likely that all PO solutions require that certain variables must be fixed to specific values within their specified lower and upper bounds. Such fixed variables can be treated as the most important variables by the designers, as the only way to create a PO solution is to make these fixed variables set to their specified set values, discovered by our approach. Often, the fixed variables get set to their lower or upper bounds, indicating the designers that they should try to relax the respective lower or upper bound, if possible, to possibly obtain a better PF. However, some variables may get fixed to an intermediate value within the lower and upper bounds. These variables are more interesting to designers, as they mean that any smaller or larger values than the extracted fixed values cause the solutions to deviate from the PO set, making the solutions dominated or infeasible.

We use the following approach to identify fixed variables. Given the variable vectors of a PF \mathbf{X}^* , we *normalize* the vector within the range $[0, 1]$, zero being the lower bound for each variable and one being the upper bound:

$$\hat{x}_i^* = \frac{x_i^* - x_i^{(L)}}{x_i^{(U)} - x_i^{(L)}}. \quad (2)$$

Once normalized, we calculate the spread (s_i) of each variable (i -th) within the range across the entire PF containing K trade-off solutions ($\hat{\mathbf{X}}^* = (\hat{\mathbf{x}}^{(1)}, \hat{\mathbf{x}}^{(2)}, \dots, \hat{\mathbf{x}}^{(K)})$):

$$s_i = \max_{k=1}^K \left(\hat{\mathbf{x}}_i^{(k)} \right) - \min_{k=1}^K \left(\hat{\mathbf{x}}_i^{(k)} \right), \quad i = 1, 2, \dots, n. \quad (3)$$

Variables having a spread below a threshold ($s_i \leq \delta$) are considered to be constant variables and are saved in \mathbf{X}^F . These variables are kept as constants for the subsequent knowledge-augmented design optimization (KAIDO) task. Where the constant value is assumed to be the average across the whole PF. Ideally, fixed variables would exhibit zero spread. However, because the underlying process is stochastic, a small practical tolerance is necessary. Moreover, since these variables are excluded from the reduced optimization task, the threshold must be strict enough to avoid convergence to non-optimal regions. Therefore, we choose $\delta = 0.02$ for this study. A trial-and-error sensitivity study showed similar performance even after doubling this parameter value.

2.2 Identifying Variables with Relationships

We exclude already identified fixed variables from further classification. The remaining variables are analyzed for any possible relationships. For this purpose, we apply the following systematic procedure.

First, we create a correlation matrix revealing direct or inverse relationships that the remaining variables may possess. Variables having an absolute correlation value greater than a threshold θ ($= 0.9$ is used here) are chosen for further analysis. All variables which do not possess a high absolute correlation with any other remaining variable are called the *neutral* variables and are saved in \mathbf{X}^N . The remaining correlated variables are saved in \mathbf{X}^C . Thus, $\mathbf{X} = \mathbf{X}^F \cup \mathbf{X}^N \cup \mathbf{X}^C$. The interpretation of correlation strength follows established guidelines in the statistical literature, where coefficients with an absolute value of $\theta \geq 0.70$ are typically considered strong and those $\theta \geq 0.90$ very strong [17].

To achieve the mathematical relationships in a confident manner, we use this rather high value of correlation strength in this study.

Next, the correlated variables are further classified into independent variables \mathbf{X}^I and dependent variables \mathbf{X}^D , making $\mathbf{X}^C = \mathbf{X}^I \cup \mathbf{X}^D$. The purpose of this classification is to express dependent variables in terms of independent variables in user-tractable mathematical expressions. We start by creating a weighted graph, where nodes represent the variable set \mathbf{X}^C , edges represent the correlation between two variables, and the weight of the edge represents the absolute correlation value. Next, we divide the graph into communities using the Louvain clustering algorithm [2]. For each community, the variable range $[L, U]$ is normalized to $[0, 1]$ for all its associated variables. Next, we create an ideal uniform distribution of $|X^*|$ points. Then, for each variable x_i , the *IGD* [6] indicator from the ideal distribution is calculated. Finally, a *Centrality* metric is defined as a standard eigenvector centrality [22], quantifying the importance of a node in terms of its connection to other important nodes in the graph connecting the whole network. Then, we simply divide the Centrality by the *IGD* indicator to define a weight metric, as follows:

$$W_i = \frac{\text{Centrality}_i}{\text{IGD}_i}. \quad (4)$$

The variable having the largest weight W_i in each community is selected as the independent variable of the community, while the rest are classified as dependent variables. Once the independent variable for a community (variable cluster) is identified, each dependent variable can be expressed as a mathematical relationship

$$\hat{x}_j^D = \phi_j(\hat{x}_i^I), \quad \text{for each } j. \quad (5)$$

2.2.1 Identifying Relationships

The task of identifying a mathematical relationship between two variables can be challenging if no functional form is known a priori. However, for an easier interpretation purpose, we would like to have simpler and easier-to-interpret form. In this study, we use a polynomial regression approach with a maximum specified degree d : $\hat{x}_j^D = c_{j,d}(\hat{x}_i^I)^d + c_{j,d-1}(\hat{x}_i^I)^{d-1} + \dots + c_{j,1}\hat{x}_i^I + c_{j,0}$. To automate the process, we start with $d = 1$ and then regress with higher degrees one at a time. We accept a higher-degree relation only if the regression error improves more than η ($= 1e^{-3}$ is used in this study). Variables within a community that are not directly correlated with the independent variable \hat{x}_i^I are instead expressed as mathematical functions of another dependent variable \hat{x}_j^D . This establishes a chain of functional relationships that ultimately link each variable back to the independent variable.

3 Gaps in a PF

When a continuous Pareto front (PF) is represented by the finite set of Pareto-optimal (PO) solutions obtained from an optimization run, some spacing between neighboring points naturally arises from sampling. To formalize what constitutes a genuine gap in this study, we define a gap as an internal discontinuity enclosed within the discovered PF, rather than a spacing pattern attributable to sampling density or the open-ended sparsity at the extremes. In this experiment, we operationalize this definition using pseudo-weights derived from the PF: a contiguous set of neighboring PO solutions is identified as the boundary of a gap if its pairwise Euclidean distance in pseudo-weight space exceeds twice the standard deviation of the pairwise distances among all members of the discovered PF. We use pseudo-weights alone here to ensure a fair comparison with the parallel machine-learning approach. The general simplex-space algorithm—applicable to any number of objectives and capable of detecting enclosed gaps as well as incomplete convergence beyond the found extreme PF solutions—is detailed in Appendix A and produces results consistent with those reported for the two-objective problems examined in this work.

3.1 KAIDO Approach for Creating New PO Solutions in Gaps

Since our KAIDO approach has already established the variable relationships from all obtained PO solutions, we can use them to focus another EMO/EMaO run in the observed gaps alone. It is expected that the variable relationship will also extend to solutions in the gaps and focusing with the relationships may help fill the gaps. Variable relationships reduce the number of optimization

variables, hence the the additional focused optimization will be computationally quick. We can adopt the following variable reduction technique for the additional optimization run:

1. All fixed variables (\mathbf{X}^F) are set to their observed specific values. They are not treated as variables any more for the post-optimization run. The knowledge of fixed variables helps to reduce the number of decision variables for the post-optimization run.
2. Due to the availability of the mathematical relationships among correlated variables, all dependent variables can be derived from their associated independent variables. Therefore, dependent variables need not be used as variables in the post-optimization run, and the variable relationships can be used as equality constraints or used to directly obtain the dependent variables during the post-optimization run.
3. Since in most cases, variables may be found to vary in a smaller range compared to their original lower and upper bounds, the post-optimization task also becomes more focused and fast.

EMO literature contains specific focused optimization procedures, such as, reference-point based EMO (R-EMO) algorithms. For example, R-NSGA-II [14] and R-NSGA-III [28] finds PO solutions only in the designated preferred region, instead of on the entire PF. In both these approaches, an aspiration objective vector \mathbf{z} is specified to obtain the part of the PF which is closest (in the Euclidean sense in the normalized objective space) to the PF with a certain pre-specified spread. While this is ideal to create new PO solutions in a gap, the knowledge augmentation of variables can make the R-EMO application much faster. Since, the original multi-objective optimization problem can be rewritten as follows:

$$\begin{aligned}
& \text{Minimize}_{\mathbf{x}} \quad \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_M(\mathbf{x})\}, \\
& \text{subject to} \quad g_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, J, \\
& \quad \quad \quad x_i^F = x_i^{F^*}, \quad x_i \in \mathbf{X}^F, \\
& \quad \quad \quad \hat{x}_j^D = \phi_j(\hat{x}_i^I), \quad x_j \in \mathbf{X}^D, \quad x_i \in \mathbf{X}^I, \text{ and } (i, j) \text{ in same variable cluster,} \\
& \quad \quad \quad x_i^{(R,L)} \leq x_i \leq x_i^{(R,U)}, \quad i = 1, 2, \dots, n.
\end{aligned} \tag{6}$$

Bounds $x_i^{(R,L)}$ and $x_i^{(R,U)}$ are the reduced lower and upper bound observed in the PO set (\mathbf{X}^*). By eliminating the fixed and dependent variables from the original variable set, the reduced multi-objective optimization problem becomes as follows in which variable vector is constructed as $\mathbf{x}^R \in \mathbf{X}^N \cup \mathbf{X}^I$:

$$\begin{aligned}
& \text{Minimize}_{\mathbf{x}^R} \quad \{f_1(\mathbf{x}^R), f_2(\mathbf{x}^R), \dots, f_M(\mathbf{x}^R)\}, \\
& \text{subject to} \quad g_j(\mathbf{x}^R) \leq 0, \quad j = 1, 2, \dots, J, \\
& \quad \quad \quad x_i^{(R,L)} \leq x_i \leq x_i^{(R,U)}, \quad x_i \in \mathbf{x}^R.
\end{aligned} \tag{7}$$

Each $x_i^F = x_i^{F^*}$ and each dependent variable is computed as $\hat{x}_j^D = \phi_j(\hat{x}_i^I)$.

Hence, we run an R-EMO application using the reduced problem described to find solutions in the gaps of a PF, in which \mathbf{z} is created as the centroid of the boundary PO solutions of a gap.

To our understanding, a gap in a PF can happen due to following three reasons:

1. **Premature convergence of a run:** If an EMO or EMaO is terminated prematurely, either due to large computational time or premature convergence to local PFs or any other reasons for the algorithm's inability to converge to the true PF, gaps in the resulting PF may occur.
2. **Infeasible solutions in the gap:** There does not exist any feasible solutions in the gap, hence the original EMO or EMaO algorithm could not find any solution in the gap region of the PF.
3. **Dominated solutions in the gap:** There may exist solutions in the gap region, but the solutions are not non-dominated with the rest of the PF.

When a gap is observed in an EMO/EMaO-obtained PF, it is important for the user to know which of these reasons has caused the gap. This will provide user the confidence to focus on appropriate regions of the PF for choosing a preferred solution.

3.2 Gaps Due to Premature Convergence of an EMO Algorithm

If the re-optimization with reduced variables produces a better PF (feasible and dominating solutions) compared to the original EMO/EMaO run in the gap or any other region of the obtained original PF, it can be safely assumed that the original run did not converge to the true PO set. This is possible because the reduced-space search is able to concentrate better and produce a better (non-dominated) set of solutions, which theoretically should have been found by the original run. In this case, any gap that may have been observed in the original PF is an artificial matter, since the Pareto-optimality of the original PO solutions was questionable. This simply indicates that more iterations were needed or a more efficient optimization algorithm was needed to obtain a better set of solutions.

In any case, when this happens, new PO solutions can be analyzed to find revised variable relationships and a new reduced-space search can be initiated in the gaps in the new PO set. This can be repeated until no further improvement in the PO set is observed.

However, if new feasible and non-dominated points are found only at the gaps of the original PF, they can be accepted and variable relationships can be updated by considering original PO and new gap PO solutions together.

3.3 Infeasible or Dominated Gaps of a PF

If we did not discover any new feasible and non-dominated solutions in the gaps of the PF, we execute an additional optimization run with the following redefined reduced problem:

$$\text{Minimize}_{\mathbf{x}^R} \quad \{f_1(\mathbf{x}^R), f_2(\mathbf{x}^R), \dots, f_M(\mathbf{x}^R)\}, \\ x_i^{(R,L)} \leq x_i \leq x_i^{(R,U)}, \quad x_i \in \mathbf{x}^R. \quad (8)$$

Note that the problem constraints are ignored in the above definition in the hope of confirming if the gap is due to infeasibility of solutions in the gap. If we do discover solutions in the gaps, we can safely conclude that the gap was due to the presence of original problem constraints which made the gap region infeasible. On the other hand, if we still do not discover any new solutions in the gaps, we can conclude that other solutions in the PF dominate the solutions in the region.

Now, we are ready to provide our overall proposed knowledge-augmented innovative design optimization (KAIDO) procedure in step-by-step format:

- Step 1:** Find a set of PO solutions using an EMO or EMaO algorithm [8]. This will result in a set of non-dominated solutions \mathbf{X}^* , which can be normalized using variable bounds to produce $\hat{\mathbf{X}}^*$. The respective objective (\mathbf{F}^*) and constraint (\mathbf{G}^*) vectors are also available for these solutions.
- Step 2:** Classify all n design variables of the problem using obtained \mathbf{X}^* into at most four categories: fixed (\mathbf{X}^F), neutral (\mathbf{X}^N), correlated independent (\mathbf{X}^I), and correlated dependent (\mathbf{X}^D) variables. Additionally, the final two variable sets are identified with their own non-overlapping clusters.
- Step 3:** For each cluster of correlated variables, mathematical relationships $\phi()$ between each dependent and independent variable is obtained by a progressive regression procedure.
- Step 4:** Create a reduced optimization task and run an R-EMO/R-EMaO algorithm to generate points in the gaps.

1. If the resulting solutions are feasible and non-dominated relative to the original PO set, these gap solutions are accepted and a reduced EMO/EMaO run is performed. Should this run reveal the complete Pareto Front (PF), the process concludes with the finding of premature convergence localized around the gap region; otherwise, the procedure returns to Step 2 with the updated PF.
2. Alternatively, if the gap solutions are feasible and dominate some or all of the original PO solutions, the process proceeds by running a reduced EMO/EMaO task. If this produces a newer, smoother PF, the conclusion is that the overall EMO/EMaO algorithm has experienced premature convergence; if not, the process returns to the Step 2, now using the ND front of the combined PFs from previous runs.

Step 5: If no feasible and non-dominated solutions are found in Step 4, redefine the reduced problem without the constraints g_j and execute an additional R-EMO run to establish if the gap is due to unavailability of feasible solutions or presence of dominated solutions in the gaps.

The fixed nature of certain variables and relationships of other variables obtained in the above steps is a post-optimality task, which allows a designer to gather a plethora of knowledge about the nature and interactions of variables, which may not have been known to them before executing the KAIDO procedure. As it is intuitive, such knowledge about variables can be extremely useful for designers to have a more deeper understanding of their problems.

4 Existing Machine Learning based Gap Filling

An existing study used the EMO-obtained PO set \mathbf{X}^* to learn the variable associations using a machine learning (ML) approach [26]. To represent a PO solution, the study used an M -dimensional pseudo-weight vector \mathbf{w} , satisfying $\sum_{i=1}^M w_i = 1$:

$$w_i^{(k)} = \frac{(f_i^{\max} - f_i^{(k)}) / (f_i^{\max} - f_i^{\min})}{\sum_{j=1}^M (f_j^{\max} - f_j^{(k)}) / (f_j^{\max} - f_j^{\min})}, \quad i = 1, \dots, M. \quad (9)$$

Every PO solution in the objective space is represented by a unique and systematically defined \mathbf{w} -vector. For two-objective problems, the extreme (ideal f_1 and nadir f_2) PO solution has $\mathbf{w} = (1, 0)$, indicating the solution's 100% importance to f_1 . Similarly, the other extreme (nadir f_1 and ideal f_2) PO solution has $\mathbf{w} = (0, 1)$. A PO solution in the middle of the PF has $\mathbf{w} = (0.5, 0.5)$. Thus, a gap in the PF can be represented by a set of contiguous \mathbf{w} -vectors obtained from the edges of the gap.

It is then a matter of learning an ML model to map \mathbf{w} - \mathbf{x} combinations (with M inputs and n output parameters) from the obtained PF (\mathbf{X}^*). Once the ML model is trained well (with a small error), it can be suitably used to predict \mathbf{x} -vector for any given \mathbf{w} -vector. This approach does not need to have mathematical relationships among variables of the PO set, rather a learning of variable relationships implicitly through a ML model is enough. Although this approach cannot provide explicit variable relationships, it can predict an appropriate PO solution from the supplied pseudo-weight identifier. Once the \mathbf{x} -vector is predicted, it can be used to compute objective vector \mathbf{F} and constraint vector \mathbf{G} . Then, the solution's feasibility and non-domination level can be checked with respect to the EMO-obtained PF to determine the reason for the gap.

5 Test and Engineering Problems

We tested the proposed method on three different test problems, two from the literature and a newly created numerical problem with known variable characteristics and a real-world engineering design problem. The problems from the literature are a two-bar truss design problem studied in [5, 21] and the ZDT3 test problem. The description of each problem is provided in the subsequent sections.

5.1 Two-bar Truss Design Problem

The PF characteristics for the two-bar truss design problem were discovered in [13]. The problem setting is shown in Figure 1. The optimization setting is as follows:

$$\begin{aligned} \text{Minimize}_{\mathbf{x}} \quad & f_1(\mathbf{x}) = x_1 \sqrt{16 + x_3^2} + x_2 \sqrt{1 + x_3^2}, \\ \text{Minimize}_{\mathbf{x}} \quad & f_2(\mathbf{x}) = \max(\sigma_{AC}(\mathbf{x}), \sigma_{BC}(\mathbf{x})), \\ \text{Subject to} \quad & \max(\sigma_{AC}(\mathbf{x}), \sigma_{BC}(\mathbf{x})) \leq S_{\max}, \\ & 0 \leq x_1, x_2 \leq A, \\ & 1 \leq x_3 \leq 3. \end{aligned}$$

Here, x_1 is the cross-sectional area of AC and x_2 is the cross-sectional area of BC in m^2 . x_3 is the vertical distance between B (or A), indicated as y in the figure, All variables are represented in m . The two conflicting objectives are: (i) minimize the total volume of truss members and (ii) minimize

the maximum stress developed in both members (AC and BC) due to the application of the 100 *kN* load.

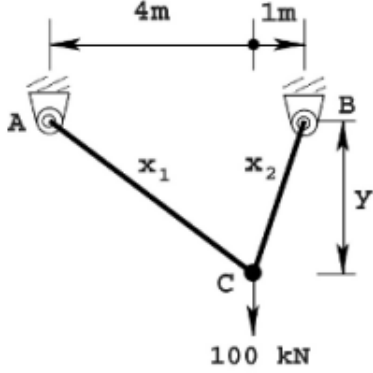


Figure 1: Two-bar truss problem.

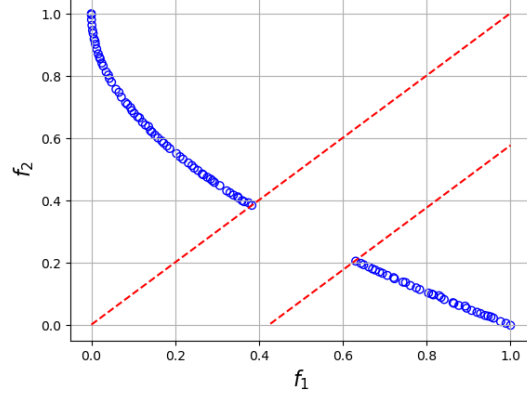


Figure 2: PF and constraint boundary for the numerical problem.

5.2 A Numerical Problem

We also design a new test problem with eight ($n = 8$) variables to demonstrate the strength of the proposed method for useful knowledge discovery and its use in filling gaps in the PF. The problem is defined, as follow:

$$\begin{aligned}
 &\text{Minimize}_{\mathbf{x}} \quad f_1(\mathbf{x}) = x_1 \\
 &\text{Minimize}_{\mathbf{x}} \quad f_2(\mathbf{x}) = (1 - \sqrt{x_1}) \times h(\mathbf{x}) \\
 &\text{Subject to:} \quad g(\mathbf{x}) = \left(\frac{-(f_1(\mathbf{x}) - 0.51) + (f_2(\mathbf{x}) - 0.30)}{\sqrt{2}} \right)^2 - 0.15^2 \leq 0 \\
 &\text{Where:} \quad h(\mathbf{x}) = 1 + 100 \left[(x_3 - (1 - 0.5x_1))^2 + (x_4 - (0.1 - 0.1x_1 + x_1^2))^2 \right. \\
 &\quad \left. + (x_5 - 0.3)^2 + (x_6 - 0.5)^2 + 15(x_7 - 0.5)^4 + 4(x_8 - 0.6)^4 + (x_2 - 0.7)^2 \right], \\
 &\quad 0 \leq x_i \leq 1 \quad \text{for } i = 1, 2, \dots, 8.
 \end{aligned} \tag{10}$$

Figure 2 shows the theoretical PF and the constraint boundary for the problem. The blue curve is the unconstrained PF, and the red lines represents the constraint boundary where everything on the boundary and outside the strip is feasible. A little thought will reveal that for the entire blue part of the PF, every term of $h(\mathbf{x})$ must be zero, thereby making the following variable relationships of the unconstrained PO set:

$$x_2 = 0.7, \tag{11}$$

$$x_3 = 1 - 0.5x_1, \tag{12}$$

$$x_4 = 0.1 - 0.1x_1 + x_1^2, \tag{13}$$

$$x_5 = 0.3, \tag{14}$$

$$x_6 = 0.5, \tag{15}$$

$$x_7 = 0.5, \tag{16}$$

$$x_8 = 0.6. \tag{17}$$

Two variables (x_7 and x_8) involving a power of four indicate that the respective terms $(x_7 - 0.5)^4$ and $(x_8 - 0.6)^4$ will take a near-zero value with a larger band of x_7 and x_8 values near 0.5 and 0.6, respectively, compared to those for other three variables x_2 , x_5 and x_6 having constant values. Thus,

if our approach works well to find a well-converged PO set, following classification of variables is expected:

$$\begin{aligned}\mathbf{X}^F &= \{x_2, x_5, x_6\}, \\ \mathbf{X}^N &= \{x_7, x_8\}, \\ \mathbf{X}^C &= \{x_1, x_3, x_4\}.\end{aligned}$$

Among \mathbf{X}^C , one of them is expected to be independent and two (due to two equations) are expected to be dependent variables. We believe that this test problem will provide a comprehensive test to our proposed approach in dealing with all four variable classes. It is important to note that if our approach does not classify x_1 as an independent variable, rather it does with another \mathbf{X}^C variable, a different relationship among correlated variables will be obtained.

Another reason for creating this problem is that the PF has a gap in the middle. This problem will allow a validation of our step-by-step KAIDO procedure.

5.3 ZDT3 Problem

We also use the well-known ZDT3 test problem with 10 ($n = 10$) variables from the ZDT test suite [30]. The PF for the problem is shown in Figure 3a. The problem is defined as follows:

$$\begin{aligned}\text{Minimize}_{\mathbf{x}} \quad & f_1(x) = x_1, \\ \text{Minimize}_{\mathbf{x}} \quad & f_2(x) = g(x) h(f_1(x), g(x)), \\ \text{Where:} \quad & g(x) = 1 + \frac{9}{n-1} \sum_{i=2}^n x_i, \\ & h(f_1, g) = 1 - \sqrt{\frac{f_1}{g} - \frac{f_1}{g}} \sin(10\pi f_1), \\ & 0 \leq x_i \leq 1 \quad \text{for } i = 1, \dots, n.\end{aligned}\tag{18}$$

The reason for choosing this problem is the natural gaps in its PF. The optimum characteristics for the problem variables are as follows:

$$\begin{aligned}0 &\leq x_1^* \leq 0.0830, \\ 0.1822 &\leq x_1^* \leq 0.2577, \\ 0.4093 &\leq x_1^* \leq 0.4538, \\ 0.6183 &\leq x_1^* \leq 0.6525, \\ 0.8233 &\leq x_1^* \leq 0.8518, \\ x_i^* &= 0 \text{ for } i = 2, \dots, n,\end{aligned}$$

having clearly visible gaps in between these piece-wise continuous ranges. The $g(\mathbf{x})$ expression reveals that all other variables except x_1 will take a value zero for the ideal PF. Additionally, the gaps observed in the PF are because of dominated solutions in between different parts of the PF, as shown in Figure 3b.

5.4 A Real-world Engineering Design

We also test our proposed approach on a real-world engineering design problem with 67 decision variables, 69 constraints, and two objectives to be minimized. This problem is obtained from an automobile industry. Due to confidentiality reasons, we represent variables with x_i , constraints with g_j and objectives with f_i . Two objective functions have an order of magnitude difference in values. Large number of variables and constraints provide a stiff test to our proposed approach.

6 Results

For each problem, we first find NSGA-II solutions and then present the variable relationships obtained by our proposed KAIDO approach. Then, depending on the existence of a gap, we compare the gap identification results from both KAIDO and ML approaches.

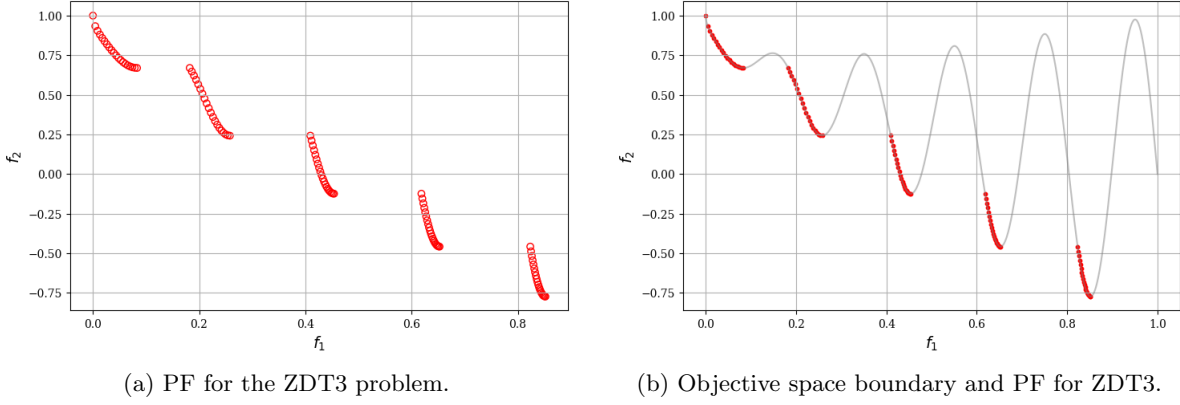


Figure 3: Theoretical PF and fitness landscape in between gaps for ZDT3

6.1 Two-bar Truss Design Problem

This problem produces a continuous PF, but to investigate our gap determination approach, we artificially create a gap in the original PF, by removing a small continuous part near minimum f_1 -values of the PF, as shown in Figure 4a. We use this modified PF to learn variable relationships, if any. Figure 4b shows the spread of variable values with the PO set. It is clear that none of the variables is fixed, while x_1 and x_3 have a smaller range of values for the PO set.

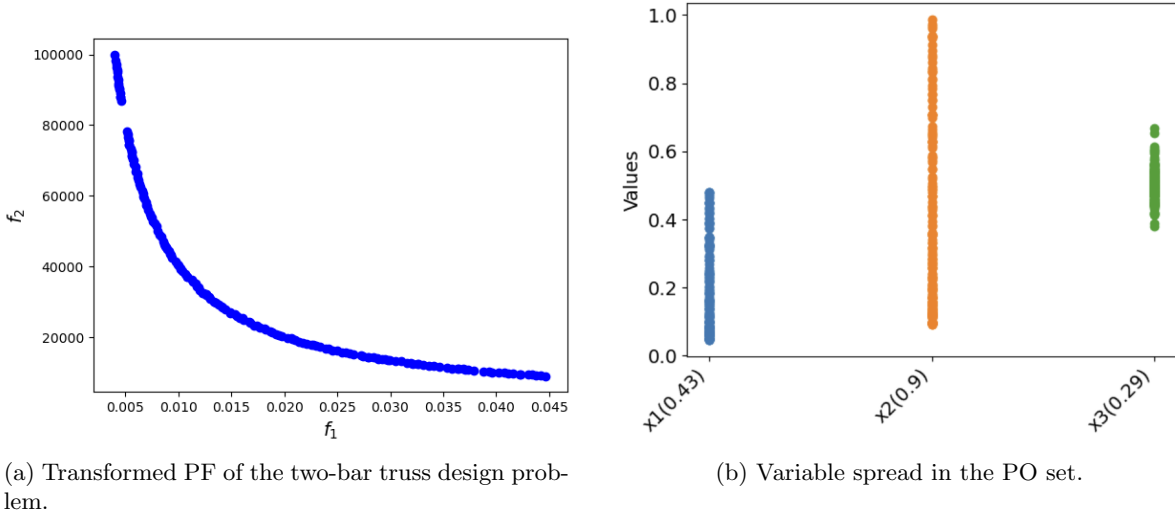


Figure 4: PF and variable spread for the two-bar truss design problem.

6.1.1 Variable Relationships Using KAIDO Approach

We start by classifying variables. To locate any fixed variable, we use $\delta = 0.02$. By this setting, no fixed variable exists for this particular PO set as seen in Figure 4b, thus, $\mathbf{X}^F = \emptyset$. Next, we move on to discover any variable relationship present in the PO set. Figure 5 shows the correlation heatmaps for all variables and selected strongly correlated variables (with correlation threshold $\theta = 0.9$). From the figure, it can be seen that x_1 and x_2 are highly correlated ($\mathbf{X}^C = \{x_1, x_2\}$) and there is no correlation of any of these two variables with x_3 . Thus, $\mathbf{X}^N = \{x_3\}$. Next, using \hat{X}^* we calculate the weight for both the variables. Since there are only two variables and a single cluster the *Centrality* of both the variables would be the same. Between the two variables, x_2 is selected as the independent variable since it has a larger weight, intuitively obtained from the higher spread in the search range mimicking the ideal distribution. Thus, the obtained relationship between normalized \hat{x}_1 and \hat{x}_2 variables is:

$$\hat{x}_1 = 1.02\hat{x}_2. \quad (19)$$

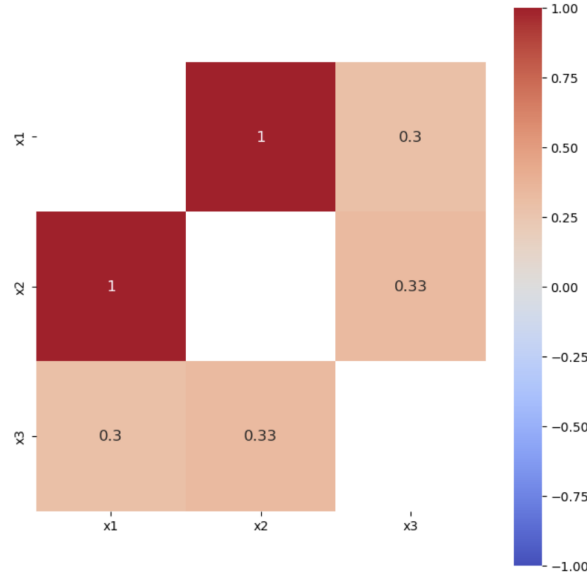
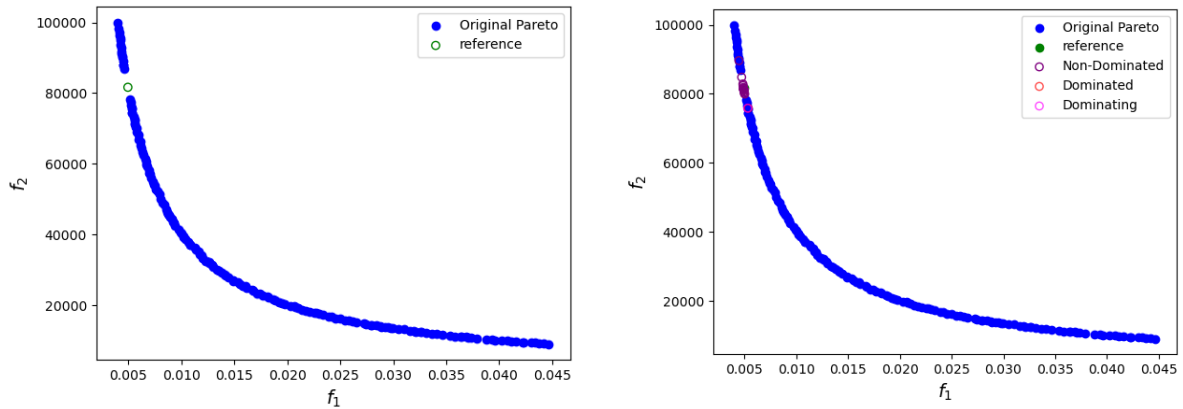


Figure 5: Correlation heatmaps for the variables of two-bar truss design problem from the PO set indicates x_1 and x_2 are highly correlated.

When these two variables are de-normalized to their original values, following relationship is obtained:

$$\begin{aligned} x_1 &= 0.492x_2 + 1.714 \times 10^{-5}, \\ &\approx 0.492x_2. \end{aligned} \quad (20)$$

A previous study obtained the relationship $x_1 = 0.5x_2$ between the two variables from the PO set by a manual graph-plotting process [11]. Our automatically-obtained relationship is close to the manually-obtained past result. Knowing variable x_1 will be almost half of variable x_2 may be a significant information to the designers. Our approach is capable of finding such vital information involving design variables to produce high-performing (near-Pareto-optimal) solutions.



(a) Placement of a reference point for R-NSGA-II.

(b) Solutions obtained within the gap close to the reference point by R-NSGA-II on two-bar truss design problem.

Figure 6: Gap-filling study with KAIDO and R-NSGA-II on two-bar truss problem.

6.1.2 Gap Analysis Using KAIDO's Variable Relationships

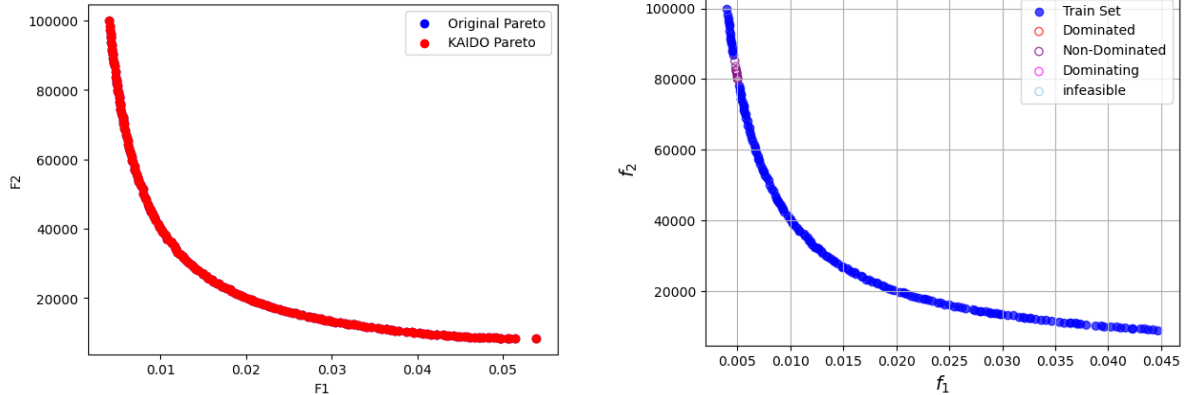
Next, we demonstrate the working of our gap-filling strategy with the obtained variable classification and relationships. We use R-NSGA-II to find solutions in the gap. A reduced optimization problem

with only x_2 and x_3 as the decision variables is now formulated. When evaluating a solution with the found values of the two variables, x_1 is predicted from x_2 using the relationship discovered in Equation 20. A reference point is created at the center of the gap, as shown in Figure 6a, for R-NSGA-II to find points close to the reference point. Figure 6b shows the solutions found by R-NSGA-II, and as seen, the new solutions fill the gap with non-dominated solutions, as they were in the original PF. This validates our approach in confirming that PO solutions exist in the gap, and shows premature convergence of the original algorithm as intended.

According to Step 4 of the KAIDO approach, we accept the new non-dominated solutions and make an additional NSGA-II run for the reduced problem defined in Equation 7 and plot the PF discovered as shown in Figure 7a. As seen reduced problem PF exactly overlays the original PF hence we conclude the analysis as a premature convergence around the gap region.

6.1.3 Gap Analysis Using ML Approach

Next, we apply the existing ML approach (Gaussian process regression (GPR) [24]) on the same PF with the artificially imposed gap to investigate the ML approach. We start by mapping the PF from the objective space to the pseudo-weight space. The ML model is trained using the weight vectors as input and respective variable vectors as output of all PO solutions except those in the gap. Once trained, pseudo-weight vectors are created at the gap and trained GPR model is used to predict the \mathbf{x} -vectors. They are evaluated for objective values and the predicted objective vectors are shown in Figure 7b. As seen, the predicted points are all non-dominated to the remaining PO set and they nicely fill the gap.



(a) PF obtained using KAIDO by NSGA-II on two-bar truss design problem.

(b) Results from the ML approach on two-bar truss problem.

Figure 7: Gap-filling with KAIDO and ML approaches on the two-bar truss problem.

6.2 Numerical Problem

The PF for this problem, obtained using NSGA-II, is shown in Figure 8a. It is clear that a part of the PF comes from the unconstrained PF. There is a distinct gap, arising from the constraint violations.

6.2.1 Variable Relationships Using KAIDO Approach

We start by classifying variables. Figure 8b shows the spread of different variables within the search space in the PO set. We use $\delta = 0.02$ as before and find three variables to be fixed $x_2 = 0.699$, $x_5 = 0.299$, and $x_6 = 0.500$, thus, $\mathbf{X}^F = \{x_2, x_5, x_6\}$. Note these fixed values are almost the same as expected and mentioned in Section 5.1. Removing these fixed variables, we now calculate the correlation of different variables as shown in Figure 9a and filter highly correlated variables ($\theta = 0.9$).

From the correlation heatmap, we obtain $\mathbf{X}^C = \{x_1, x_3, x_4\}$, and $\mathbf{X}^N = \{x_7, x_8\}$. Figure 9b shows a single cluster of related variables, where the nodes are the variables and the edges are the correlations.

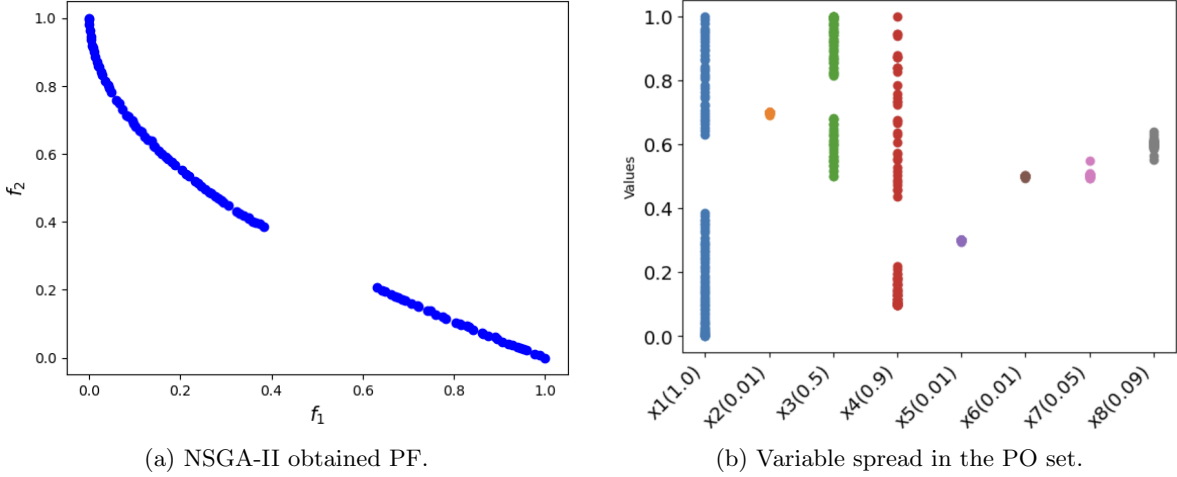


Figure 8: PF with a natural gap and observed variable spread for the numerical problem.

A single cluster indicates that there is a single independent variable which can be used to express other dependent variables of the cluster. Positive correlation is represented as a red edge, where the negative correlation is shown with a blue edge. The independent variable for the cluster is the red colored node x_1 , obtained by our procedure. Relationships found for normalized variables are as follows:

$$\hat{x}_3 = 1.00 - 1.00\hat{x}_1, \quad (21)$$

$$\hat{x}_4 = -0.01 - 0.03\hat{x}_1 + 1.02\hat{x}_1^2. \quad (22)$$

The de-normalised forms of the above relations are given as follows:

$$x_3 = 1.0 - 0.5x_1, \quad (23)$$

$$x_4 = 0.091 - 0.027x_1 + 0.918x_1^2. \quad (24)$$

These two obtained relationships from the reduced PF and using our systematic approach are very similar to theoretical relationships presented in Equations 12 and 13.

6.2.2 Gap Analysis Using KAIDO Approach

Having identified fixed variables and uncovered structural relationships among the remaining decision variables, we reformulate the optimization problem by reducing the search space to include only x_1 , x_7 , and x_8 . Their respective bounds are restricted to those observed within the Pareto-optimal (PO) set. Figure 10a illustrates the placement of a reference point within the observed gap, while Figure 10b presents the solution set obtained using R-NSGA-II with parameters $N = 30$, $T_{max} = 100$, and $\epsilon = 0.005$. Notably, all discovered solutions lie within the feasible region of the Pareto front (PF), and none occupy the gap itself. The absence of solutions in the gap across both the original NSGA-II run and the targeted R-NSGA-II run suggests that the gap is not attributable to premature convergence of the EMO run. Thus, we move onto the next step by redefining the reduced problem without constraints (Equation 8).

The resulting solution set, shown in Figure 11a, now includes points located precisely within the gap. Upon evaluating these solutions for constraint violation, we find that they are indeed infeasible as they are within the boundary marked by the dashed lines in the figure. This confirms that the gap in the original PF arises from an infeasible region and KAIDO successfully identifies this underlying cause. While no new solutions were discovered, the process revealed critical variable characteristics that deepen our understanding of the PF's structure and its connection to the physical and dynamic properties of the problem.

6.2.3 Gap Analysis Using ML Approach

Next, we apply the ML approach to fill the same gap on the PF. We generate pseudo-weight vectors in the gap and predict variable vectors using the forward pass of the model. The predicted vectors

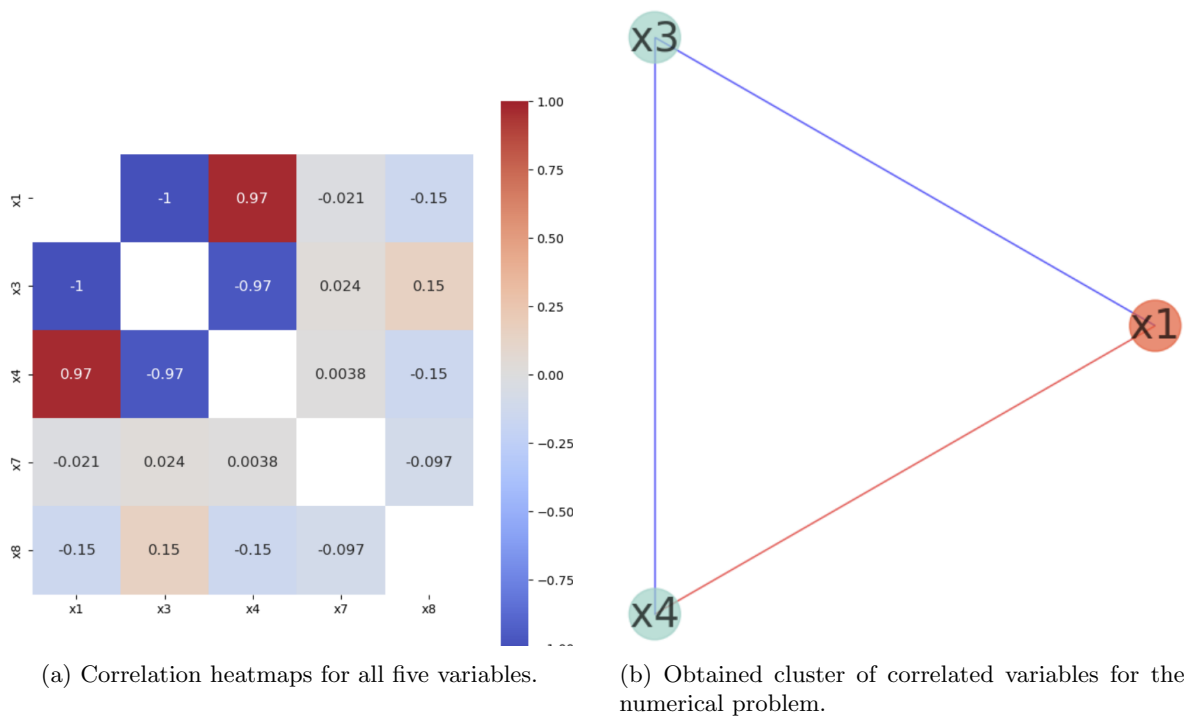


Figure 9: Correlation heat-map and correlated variable cluster for the numerical problem.

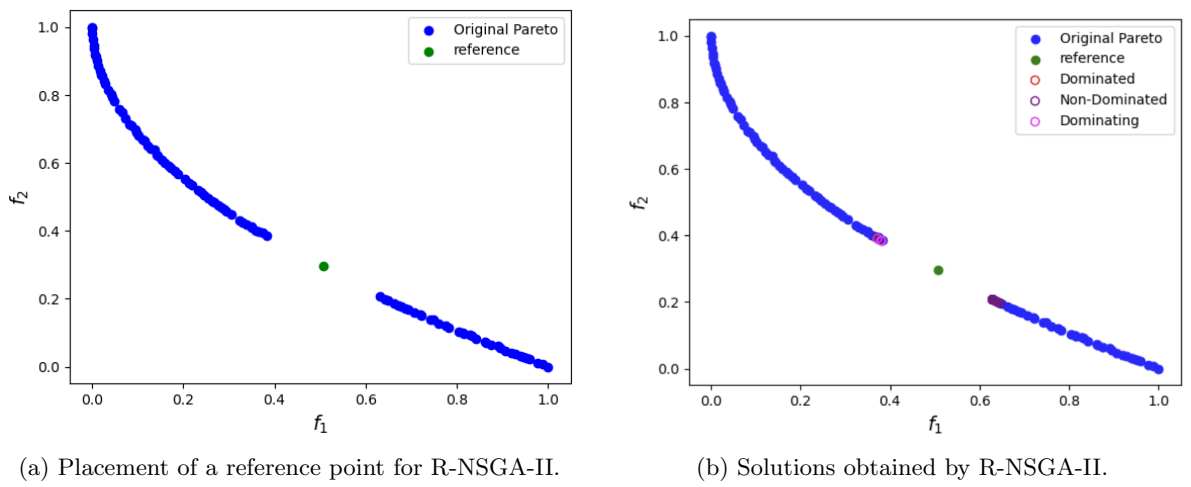
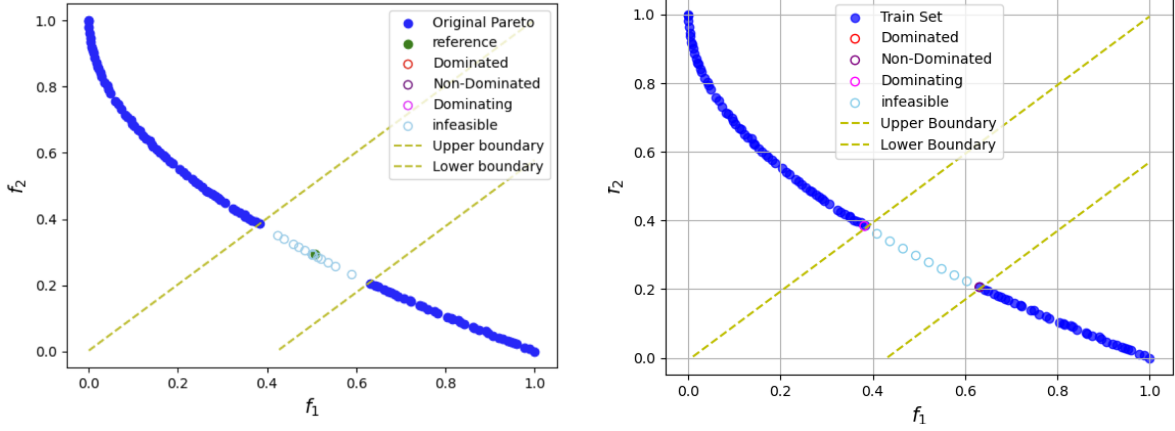


Figure 10: KAIDO approach for the numerical problem.



(a) Solutions obtained with the reduced problem without constraints using RNSGA-II.

(b) Results from the ML approach on the numerical problem.

Figure 11: Gap-filling with ML and KAIDO knowledge authenticity on numerical problem. ML model produces infeasible solutions in the gap from the get go, while the KAIDO approach needs an additional optimization run without the constraint to produce similar results, since optimization algorithms never produce infeasible results in the presence of problem constraints.

are evaluated for their objective and constraint values, and plotted in the objective space with their feasibility characteristic and dominance relationship, as seen in Figure 11b.

The ML method yields similar results, successfully predicting new solutions within the observed gap of the PF. Upon evaluation, it is observed all predicted solutions in the gap are infeasible. While this confirms that the gap corresponds to an infeasible region, the ML approach offers no insight into the underlying structural or physical dynamics of the problem. As a black-box model, it lacks interpretability and does not reveal the variable relationships or constraint mechanisms responsible for the gap. Although the method may require fewer evaluations and can efficiently approximate solution distributions, it falls short in providing meaningful and interpretable knowledge about the problem’s internal behavior — an essential aspect for informed decision-making and a deeper understanding of the Pareto front topology.

6.3 ZDT3 Problem

The PF for this problem, obtained using NSGA-II, is shown in Figure 12a. As seen, there are several gaps between different regions of the PF.

6.3.1 Variable Relationships Using KAIDO Approach

We start by classifying variables, Figure 12b shows the spread of different variables within the search space in the PO set. We use $\delta = 0.02$ as before and find nine variables to be fixed $x_i^* = 0$ for $i = 2, \dots, 10$, thus, $\mathbf{X}^F = \{\mathbf{x}_2, \dots, \mathbf{x}_{10}\}$, $\mathbf{X}^N = \{\mathbf{x}_1\}$, and $\mathbf{X}^C = \emptyset$. This reduced the problem to a single variable and no further analysis is required.

6.3.2 Gap Analysis Using KAIDO

Having identified fixed variables, we reformulate the optimization problem by reducing the search space to include only x_1 . The respective bounds are restricted to those observed within the Pareto-optimal (PO) set. Figure 13a illustrates the placement of a reference points within the observed gaps, while Figure 13b presents the solution set obtained using R-NSGA-II with parameters $N = 30$, $T_{max} = 100$, and $\epsilon = 0.005$. Since there are no constraints for the problem and both RNSGA-II and NSGA-II did not find any points in the gaps, we can safely conclude that the gaps are because of dominated regions. This can be seen in Figure 14a, where we plot the landscape of ZDT3 within the gaps with the PF and KAIDO points.

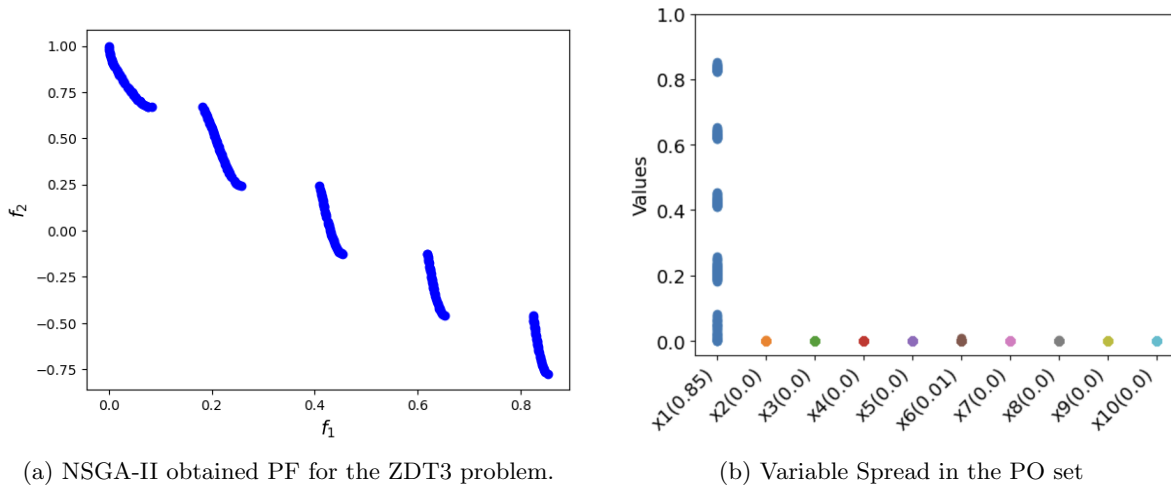


Figure 12: Gaped PF and variable spread for ZDT3 indicate nine out of 10 variables stay fixed while a disjointed variation of one variable causes the entire disjointed PF.

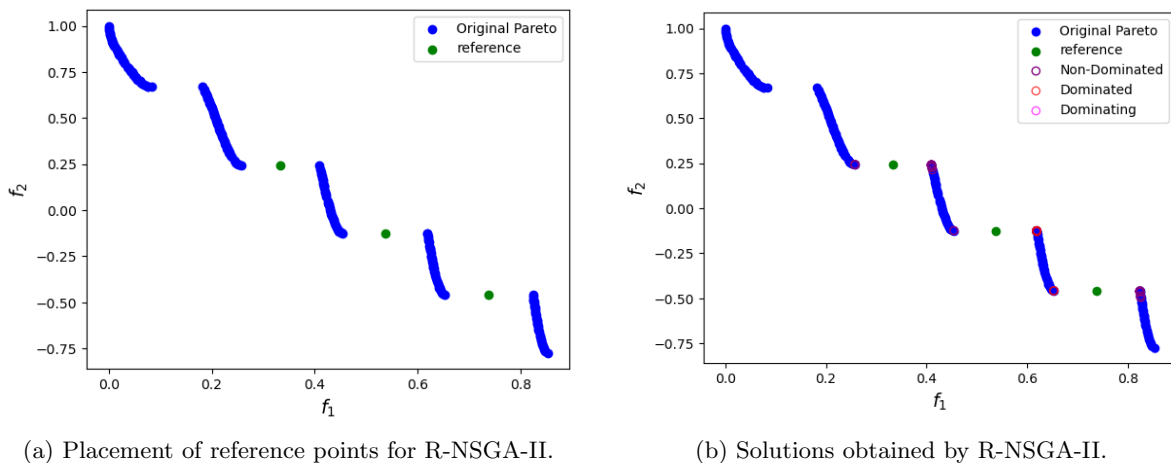
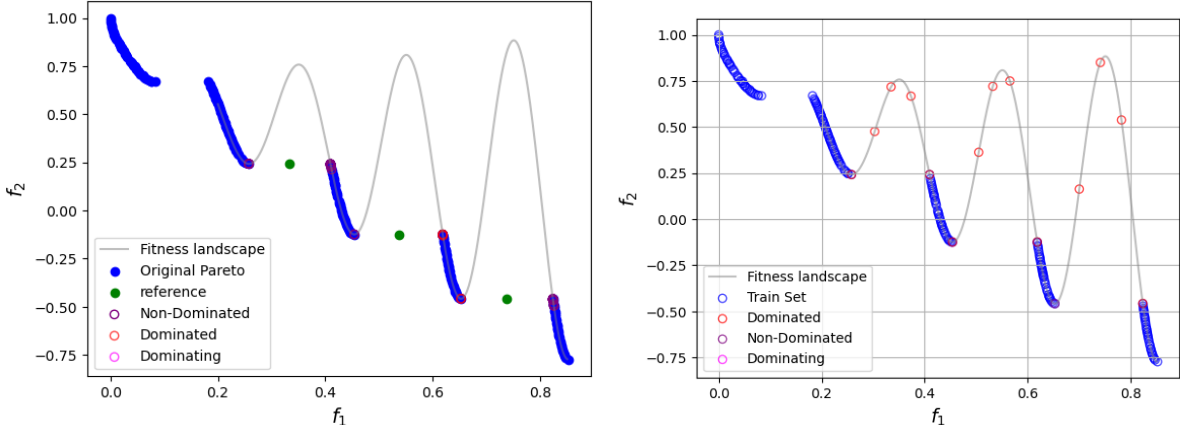


Figure 13: KAIDO approach for ZDT3 with the original problem fitness landscape within the gaps.

6.3.3 Gap Analysis Using ML Approach

To investigate the origin of the observed gap, we employ a machine learning (ML) approach. As in the previous analysis, the ML model is trained using weight vectors associated with the Pareto front (PF), thereby learning a mapping from the weight space to the decision variable space. We then generate arbitrary weight vectors located within the gap region and use the trained ML model to predict the corresponding \mathbf{x} -vectors. These predicted solutions are subsequently evaluated. The resulting solutions are illustrated in Figure 14b, which depicts the ZDT3 fitness landscape across the gap region. Notably, the ML predictions trace the fitness landscape between the gaps, yielding dominated solutions. This outcome demonstrates that the gap arises precisely because solutions in this region are dominated, confirming the structural nature of the discontinuity in the PF.



(a) ZDT3 fitness landscape and the found PF with KAIDO points.

(b) Solutions obtained by the ML approach.

Figure 14: Solutions found by both approaches for gap identification of the ZDT3 test problem with the original fitness landscape in between gaps plotted as well.

6.4 Real-world Engineering Design Problem

Having shown the three different cases: (i) existence of feasible and non-dominated solutions in the gap of a PF, (ii) non-existence of feasible solutions in the gap of a PF, and (iii) non-existence of non-dominated solutions in the gap of a PF, on simplistic test and engineering problems, we now consider a real-world problem from an industry involving 67 variables, 69 constraints and two objectives.

First, the NSGA-II non-dominated solutions of this problem are shown in Figure 15. It is clear that the obtained PF has a few apparent gaps. In addition to discovering interpretable variable relationships, we use both the KAIDO and the ML approaches to attempt to find potential solutions in the gaps and also identify the reason for the existence of the gaps in the front.

6.4.1 Variable Relationships Using KAIDO Approach

Using the PF shown in Figure 15, we discover different variable characteristics of the problem using $\delta = 0.02$, and $\theta = 0.9$. Figure 16 shows the spread of all decision variables of the PO set within the original search space. With $\delta = 0.02$ we find 10 fixed variables, $x_{11} = 0.69$, $x_{15} = 1.27$, $x_{16} = 4.66$, $x_{20} = 9.93$, $x_{21} = 8.70$, $x_{22} = 0.86$, $x_{23} = 0.42$, $x_{24} = 0.35$, $x_{25} = 6.98$, $x_{26} = 0.54$. Figure 17b shows the 21 highly correlated variables above the threshold and Figure 18 shows the variable clusters with independent variables (colored in red) for each cluster. The observed relationships between different normalized variables within a cluster are given, as follows:

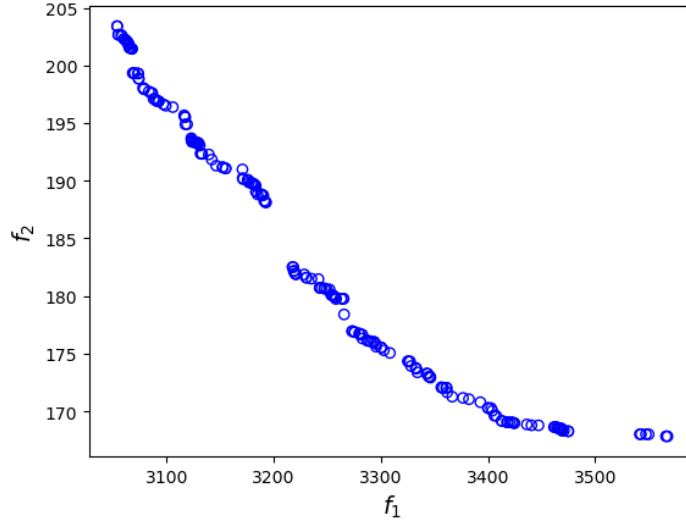


Figure 15: NSGA-II obtained PF for the real-world 67-variable engineering design problem.

Cluster1 : (\hat{x}_{62})	$\hat{x}_{31} = 0.99 - 0.56 \hat{x}_{62} - 0.24 \hat{x}_{62}^2$ $\hat{x}_{32} = 0.82 - 1.04 \hat{x}_{62} + 0.34 \hat{x}_{62}^2$ $\hat{x}_{33} = 0.83 - 1.70 \hat{x}_{62} + 0.91 \hat{x}_{62}^2$ $\hat{x}_5 = 0.10 + 0.88 \hat{x}_{33}$ $\hat{x}_{44} = 1.01 - 0.11 \hat{x}_5 - 0.72 \hat{x}_5^2$
Cluster2 : (\hat{x}_{43})	$\hat{x}_6 = 0.71 - 1.07 \hat{x}_{43}^2 + 0.28 \hat{x}_{43}^3$ $\hat{x}_7 = 0.04 + 0.13 \hat{x}_{43} + 1.55 \hat{x}_{43}^2 - 0.67 \hat{x}_{43}^3$ $\hat{x}_{34} = 0.07 + 0.12 \hat{x}_{43} + 0.99 \hat{x}_{43}^2 - 0.61 \hat{x}_{43}^3$ $\hat{x}_{45} = 0.16 + 0.37 \hat{x}_{43} + 0.93 \hat{x}_{43}^2 - 0.37 \hat{x}_{43}^3$ $\hat{x}_{47} = 0.07 - 0.01 \hat{x}_{43} + 1.59 \hat{x}_{43}^2 - 0.97 \hat{x}_{43}^3$ $\hat{x}_{64} = 0.08 + 0.02 \hat{x}_{43} + 1.37 \hat{x}_{43}^2 - 0.70 \hat{x}_{43}^3$ $\hat{x}_{51} = 0.10 - 0.14 \hat{x}_{47} + 2.18 \hat{x}_{47}^2 - 1.10 \hat{x}_{47}^3$
Cluster3 : (\hat{x}_{67})	$\hat{x}_{50} = 0.16 - 0.25 \hat{x}_{67} + 2.18 \hat{x}_{67}^2 - 1.14 \hat{x}_{67}^3$
Cluster4 : (\hat{x}_{41})	$\hat{x}_{35} = 0.03 + 1.49 \hat{x}_{41} - 0.47 \hat{x}_{41}^2$ $\hat{x}_{36} = 0.09 + 0.94 \hat{x}_{41}$ $\hat{x}_{40} = 0.02 + 0.78 \hat{x}_{41} + 1.26 \hat{x}_{41}^2 - 1.03 \hat{x}_{41}^3$ $\hat{x}_{29} = 0.90 + 0.32 \hat{x}_{38} - 4.18 \hat{x}_{38}^2 + 3.03 \hat{x}_{38}^3$ $\hat{x}_{38} = 0.06 - 0.43 \hat{x}_{40} + 3.80 \hat{x}_{40}^2 - 2.40 \hat{x}_{40}^3$

Note that there is only one independent variable for each cluster. As some variables within a cluster are not highly correlated with the independent variable, they are expressed in relation to a dependent variable, forming a chain of relations ultimately linking all the dependent variables to the independent variable. The independent variables are marked for each cluster.

6.4.2 Gap Analysis Using KAIDO

Having found 10 fixed variables and 18 dependent variables with their simplistic polynomial relationships with four independent variables, we redefine the original problem to a reduced version using only the remaining 35 neutral and four independent variables. Thus, from 67 original variables the reduced problem has only 39 variables. There are 18 new equality constraints that relates 18 dependent variables with four independent variables. Figure 19a shows the reference points generated for the gaps using green circles. We use the focused R-NSGA-II to fill the gaps using the reference vectors generated at the center of the gaps, with $N = 100$, $T_{max} = 150$, $\epsilon = 0.005$. Figure 19b shows the solution found by R-NSGA-II. As observed in the figure, in several instances, we identify solutions within the gap regions that dominate the originally discovered PF, suggesting that the initial optimization suffered

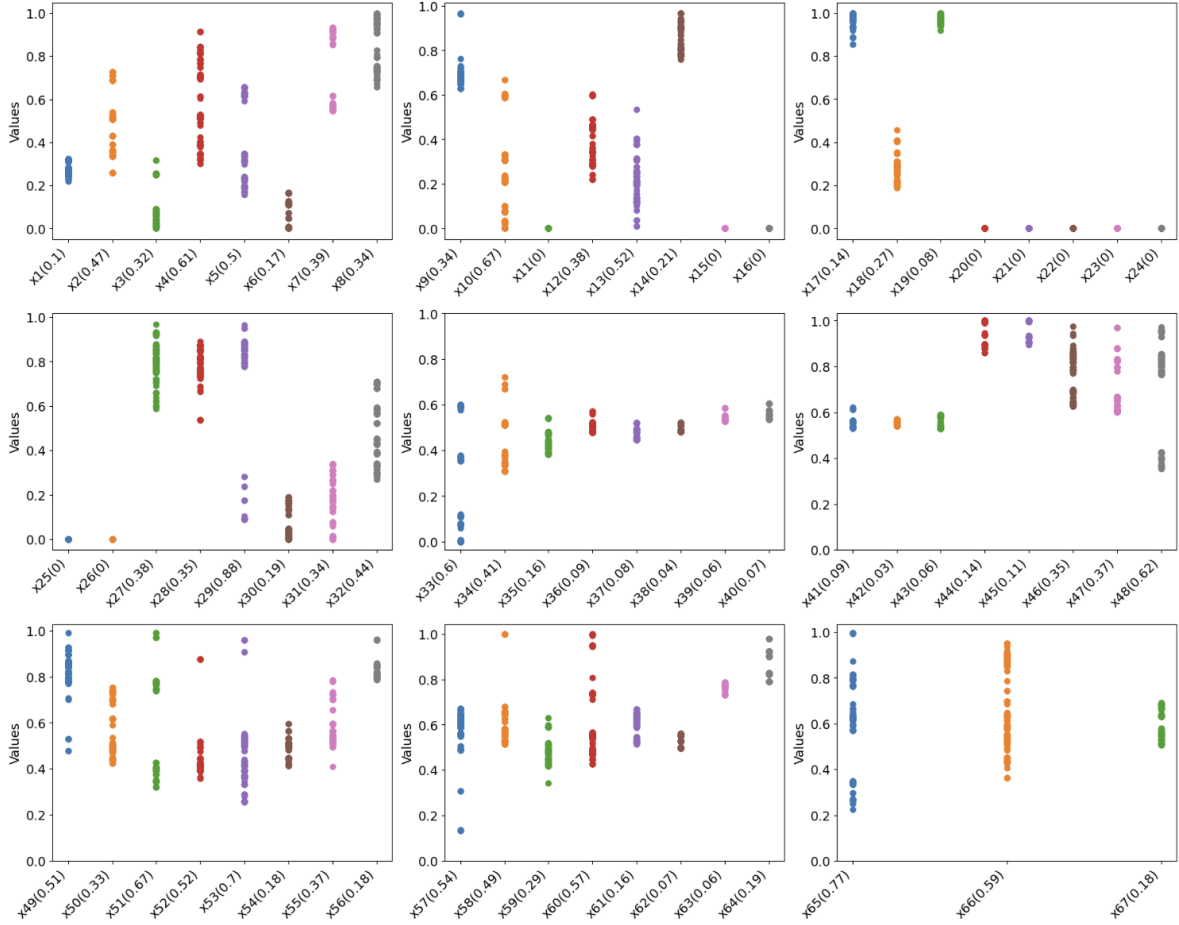


Figure 16: Variable spread in the PO set indicating different types of convergence levels for the 67-variable real-world engineering design problem.

from premature convergence. To address this, we reformulate the problem in its reduced form (using the variable relationships and having 39 variables) and perform a re-optimization using NSGA-II. The resulting PF, shown in Figure 20a, clearly dominates the previously obtained PF and exhibits a much smoother formation. Importantly, the gaps observed earlier are no longer present, confirming that they were artifacts of premature convergence of the original NSGA-II run. This outcome highlights the effectiveness of the KAIDO approach in detecting premature convergence manifested as artificial gaps. Another variable relationship study from the combined new and previous PFs can be performed and the process can be repeated until a proper convergence of the PF can be achieved. For brevity, we do not pursue this additional runs.

6.4.3 Gap Analysis Using ML Approach

Following the same procedure as in the test problems, we generate pseudo-weight vectors from the obtained PF. We train the ML model using this training set. Next, we generate five intermediate points within each gap to predict the \mathbf{x} -vectors. The predicted solutions are shown in Figure 20b. As observed, most of the predicted solutions are infeasible, with the exception of the gap in the bottom-right corner, where all solutions are dominated. Importantly, no new non-dominated solutions emerge in any of the gaps, in contrast to the predictions obtained by the KAIDO approach. This outcome highlights a key limitation of the ML approach: while it can approximate solutions within gaps, it lacks the capacity to detect or correct the broader issue of premature convergence in the initial optimization run. Consequently, the ML approach would incorrectly conclude that most gaps arise from infeasibility, whereas the KAIDO framework reveals that they are in fact artifacts of premature convergence in the NSGA-II run.

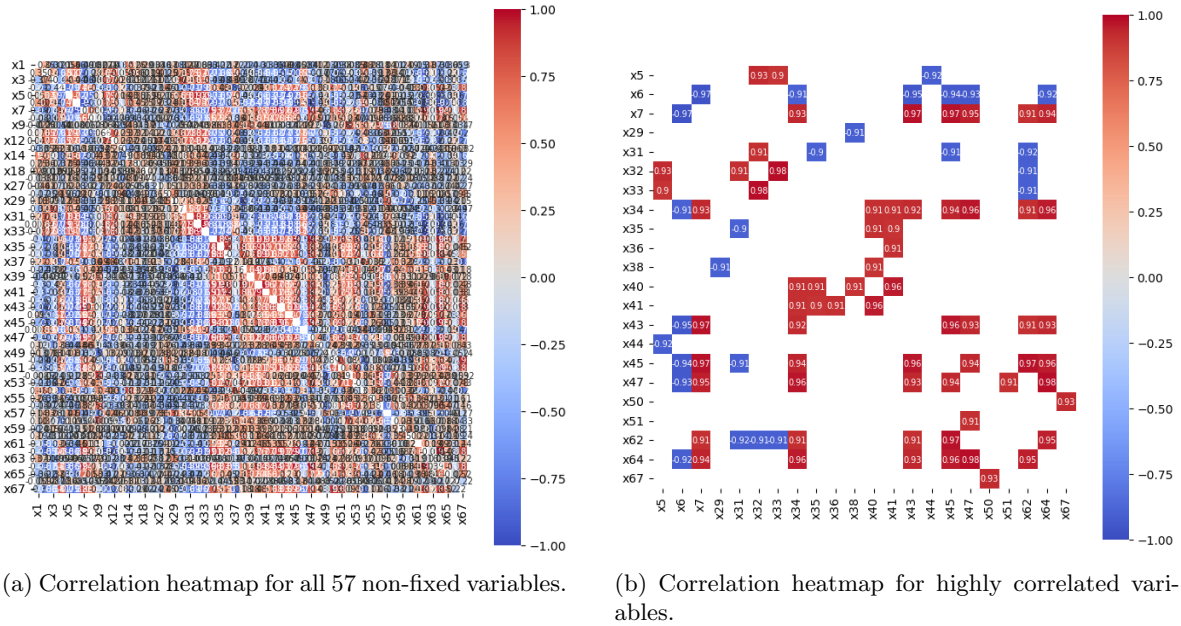


Figure 17: Correlation heatmaps for the variables from the PO set for the 67-variable real-world engineering design problem.

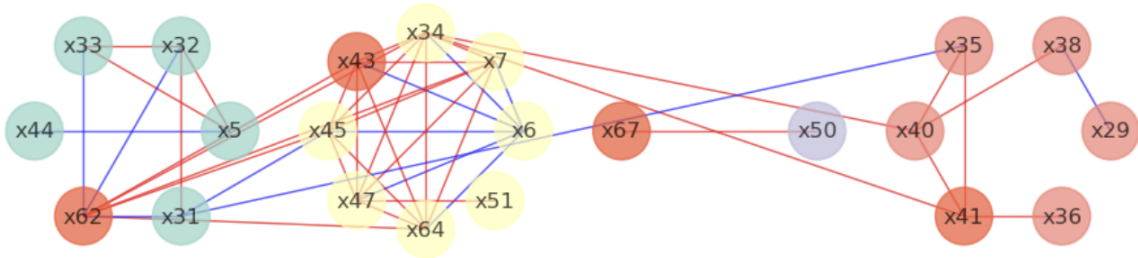


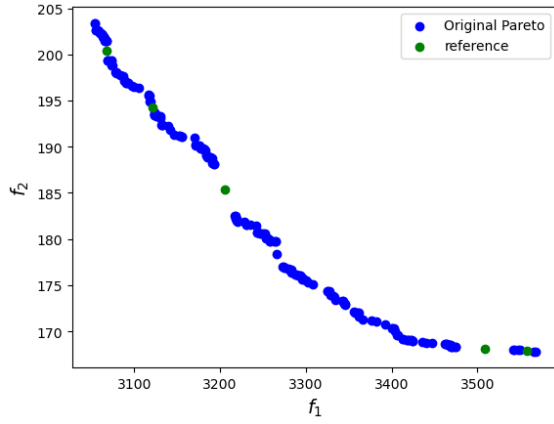
Figure 18: Obtained four clusters of correlated variables for the 67-variable real-world engineering design problem.

6.5 Comparison of Two Approaches

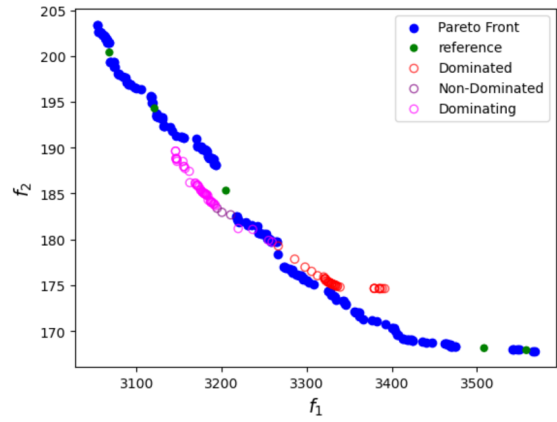
Both the KAIDO and ML-based approaches aim to learn the underlying structure of the PF in multi-objective optimization problems, but they differ significantly in their methodologies and the nature of the insights they provide. The KAIDO approach is more explicit, focusing on extracting meaningful and interpretable knowledge from the Pareto front. This knowledge can be reused across different design scenarios and provides broader insights into the problem domain. In contrast, the machine learning (ML) approach is more implicit, treating the learning task as a black-box function approximation. It is typically more resource-efficient in terms of gap filling, directly predicting solutions for unseen pseudo-weight vectors without additional optimization, but it does not provide any explicit and interpretable variable relationships at the end.

Another key distinction lies in solution precision and exploration. The ML approach tends to be more precise in the sense that it outputs a single predicted solution for each given pseudo-weight vector. However, the precision can be misleading in complex problems with hard constraints, as the observed gaps could be the result of premature convergence. The KAIDO approach, on the other hand, is more explorative – instead of providing a single solution, it identifies the best feasible solutions near a given reference point, often yielding a set of trade-off solutions that reveal more about the structure of the PF, especially in highly constrained problems.

This difference between the two approaches is negligible for simple, well-behaved problems (e.g., the

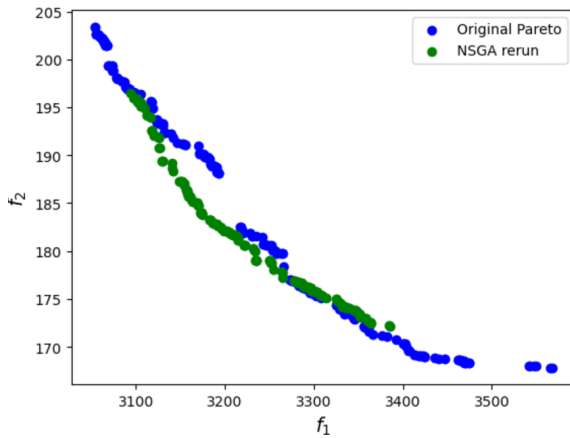


(a) Placement of a reference points for R-NSGA-II.

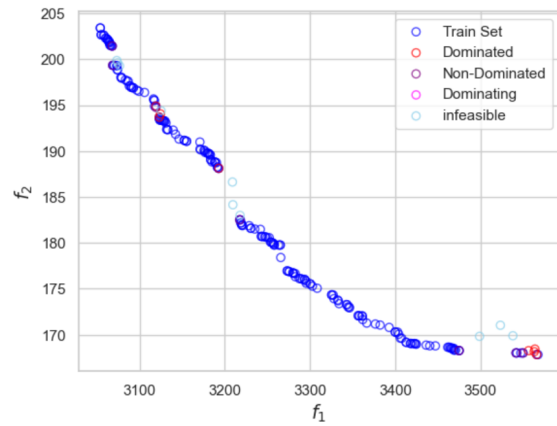


(b) Solutions obtained within the gap close to a reference point by R-NSGA-II.

Figure 19: Gap-filling study with KAIDO and R-NSGA-II on the 67-variable real-world engineering design problem.



(a) New PF using knowledge discovered with NSGA-II.



(b) Results from the ML approach.

Figure 20: Gap-identification with KAIDO and ML methods on the 67-variable real-world engineering design problem. The KAIDO approach is able to detect overall premature convergence and improve the PF, whereas the ML method concludes infeasibility as the reason.

two-bar truss), where both approaches yield similar results. However, as the complexity of the problem increases—particularly in constrained or real-world engineering problems, the differences become more pronounced. In these cases, the gaps in the PF are not merely due to infeasibility or domination but are largely artifacts of overall premature convergence of the optimization algorithm. The ML approach, while able to produce solutions exactly within the gap, cannot detect or correct this broader convergence issue. Its precision, which appears to be a strength on the surface, becomes a limitation: it can only show infeasible or dominated solutions in the gap but lacks the ability to improve the PF or reveal that the entire run has converged prematurely.

By contrast, KAIDO’s explorative and constraint-aware search, coupled with re-optimization (R-EMO), can identify and rectify premature convergence. Even when gaps appear infeasible or dominated, KAIDO can generate feasible alternatives near reference points and, more importantly, improve the PF itself. This capability was demonstrated in the real-world problems, where KAIDO successfully revealed that the true reason for the gaps was premature convergence of the NSGA-II run.

Due to the use of a focused optimization procedure (R-EMO) involved in the KAIDO approach, it can be computationally expensive compared to the ML approach. However, the training time required for the ML must be compared with the optimization time needed for the KAIDO approach.

Ultimately, the choice between these methods depends on the user’s objective. If the goal is to obtain a specific solution in a gap, the ML approach is more appropriate, as it provides precise predictions directly within gap regions. However, if the aim is to uncover the structural reasons for gaps – whether they arise from infeasibility, domination, or premature convergence – the KAIDO framework offers greater interpretability and robustness. By leveraging knowledge discovery of fixed variables and variable inter-dependencies, KAIDO not only identifies the causes of gaps but also reveals broader convergence issues, thereby enabling corrective re-optimization and improvement of the Pareto front.

Table 1: Comparison of ML and KAIDO approaches for identifying reasons for gaps in the Pareto front.

Gap Reason	ML Approach	KAIDO Approach
Premature Convergence	Incomplete: can only produce solutions in the gap. If premature convergence occurs around the gap, ML may show results, but it cannot detect or improve overall premature convergence.	Complete: detects premature convergence and, through reoptimization (R-EMO), can improve the PF and eliminate gaps.
Infeasibility	Direct: produces exact solutions in the gap, explicitly showing infeasibility.	Direct: Detectable but requires additional post-optimization runs to confirm infeasibility.
Domination	Direct: produces dominated solutions exactly in the gap, demonstrating domination explicitly.	Direct: Concluded indirectly due to absence of non-dominated solutions; requires post-optimization runs for confirmation.

In summary, both ML and KAIDO approaches are capable of **filling** gaps in the **discovered** Pareto front; however, as highlighted in Table 1, their mechanisms and implications differ. The ML approach, with its precise predictions, can directly demonstrate infeasibility and domination by producing solutions exactly within gap regions, but it remains incomplete in diagnosing and correcting overall premature convergence. KAIDO, by contrast, requires additional post-optimization runs to confirm infeasibility and domination, yet its explorative and constraint-aware search uniquely enables the detection and rectification of premature convergence, thereby improving the Pareto front itself. This comparative analysis underscores that while ML offers efficiency and precision, KAIDO provides deeper interpretability and robustness, particularly in complex or constrained problems where premature convergence is the dominant factor.

7 Conclusions

We have introduced a Knowledge-Augmented Innovative Design Optimization (KAIDO) approach aimed at identifying the structural reasons for gaps in the Pareto front (PF), specifically distinguishing between infeasibility, domination, and premature convergence. The method has been validated on four distinct problems and bench-marked against a baseline machine learning (ML)-based gap-filling approach. While both approaches can identify gap regions, their mechanisms and implications differ. The ML approach offers computational efficiency and precise predictions for given weight vectors, but its implicit nature limits it to local gap identification. It does not extract variable-level knowledge hidden in the PO solutions and therefore cannot provide any new feasible Pareto-optimal (PO) solutions or improve the PF itself. In contrast, KAIDO not only identifies the reasons for gaps but also produces knowledge in terms of variable relationships and structural characteristics of the PF. This knowledge can be used for broader purposes beyond just the gap identification, for example, as demonstrated, to improve the PF by correcting premature convergence and generating more diverse feasible solutions. Although KAIDO requires additional solution evaluations due to its re-optimization procedure, this added computational effort is offset by the interpretability and actionable insights it provides.

The findings demonstrate that neither approach is universally superior to each other; rather, their utility depends on the specific goals of the decision-maker. For applications where computational speed and deterministic predictions are prioritized, ML methods may be preferable. Conversely, for scenarios requiring deeper understanding of trade-offs and feasible design space exploration, the proposed KAIDO approach has shown promise. Further ideas to relax satisfaction of obtained variable relationships and provide more emphasis on constraint satisfaction or other local requirements to create more diverse non-dominated and feasible solutions at the gaps would be an interesting future study. A future study may also explore a hybrid approach and make a computational complexity comparison of both approaches. Definitely, applying both approaches on more test and real-world problems will provide further insights into them. Nevertheless, this study has provided a systematic Pareto data analysis approach as a post-optimality study for deciphering useful variable relationships in terms of knowledge for a better understanding of the problem and also for further improvements of the PF or other tasks, before designers choose a single preferred solution for implementation.

8 Data Availability

The authors confirm that the data supporting the findings of the test problems in this study are available within the article. As for the Real-world problem, data cannot be made available due to commercial restrictions.

References

- [1] BEJARANO, L. A., ESPITIA, H. E., AND MONTENEGRO, C. E. Clustering analysis for the pareto optimal front in multi-objective optimization. *Computation* 10, 3 (2022), 37.
- [2] BLONDEL, V. D., GUILLAUME, J.-L., LAMBIOTTE, R., AND LEFEBVRE, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008, 10 (2008), P10008.
- [3] BURKOTOVÁ, J., AGHAEI POUR, P., KRÁTKÝ, T., AND MIETTINEN, K. Interactive multiobjective optimization of an extremely computationally expensive pump design problem. *Engineering Optimization* 56, 8 (2024), 1318–1333.
- [4] CERDA-FLORES, S. C., ROJAS-PUNZO, A. A., AND NÁPOLES-RIVERA, F. Applications of multi-objective optimization to industrial processes: a literature review. *Processes* 10, 1 (2022), 133.
- [5] CHANKONG, V., AND HAIMES, Y. Y. *Multiobjective Decision Making Theory and Methodology*. New York: North-Holland, 1983.

- [6] COELLO COELLO, C. A., AND REYES SIERRA, M. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *MICAI 2004: Advances in Artificial Intelligence: Third Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, April 26-30, 2004. Proceedings 3* (2004), Springer, pp. 688–697.
- [7] CUIRIEL-OLIVARES, G., ESCOBAR, G., JOHNSON, S., AND SCHACHT-RODRÍGUEZ, R. Mpc-based ems for a series hybrid electric tractor. *IEEE Access* 12 (2024), 135999–136010.
- [8] DEB, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK, 2001.
- [9] DEB, K., BANDARU, S., GREINER, D., GASPAR-CUNHA, A., AND TUTUM, C. C. An integrated approach to automated innovation for discovering useful design principles: Case studies from engineering. *Applied Soft Computing* 15 (2014), 42 – 56.
- [10] DEB, K., SINDHYA, K., AND HAKANEN, J. Multi-objective optimization. In *Decision sciences*. CRC Press, 2016, pp. 161–200.
- [11] DEB, K., AND SRINIVASAN, A. Innovization: Innovating design principles through optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2006)* (New York: ACM, 2006), pp. 1629–1636.
- [12] DEB, K., AND SRINIVASAN, A. Innovization: Discovery of innovative design principles through multiobjective evolutionary optimization. In *Multiobjective Problem Solving from Nature: From Concepts to Applications*, J. Knowles, D. Corne, and K. Deb, Eds. Berlin: Springer, 2008, pp. 243–262.
- [13] DEB, K., AND SRINIVASAN, A. Innovization: Discovery of innovative design principles through multiobjective evolutionary optimization. *Multiobjective Problem Solving from Nature: From Concepts to Applications* (2008), 243–262.
- [14] DEB, K., SUNDAR, J., UDAY, N., AND CHAUDHURI, S. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research (IJ CIR)* 2, 6 (2006), 273–286.
- [15] ERFANI, T., AND UTYUZHNIKOV, S. V. Directed search domain: a method for even generation of the pareto frontier in multiobjective optimization. *Engineering Optimization* 43, 5 (2011), 467–484.
- [16] HAN, L., XU, A., LIU, Z., AND ZHENG, Q. A knowledge-based optimization design method for supertall buildings with a strong outer frame structural system. *Journal of Building Engineering* (2025), 112803.
- [17] HER, Q. L., AND WONG, J. Significant correlation versus strength of correlation. *American Journal of Health-System Pharmacy* 77, 2 (2020), 73–75.
- [18] KATZENBACH, A., BERGHOLZ, W., AND ROLINGER, A. Knowledge-based design—an integrated approach. In *The Future of Product Development: Proceedings of the 17th CIRP Design Conference* (2007), Springer, pp. 13–22.
- [19] KAYA, C. Y., AND MAURER, H. Optimization over the pareto front of nonconvex multi-objective optimal control problems. *Computational Optimization and Applications* 86, 3 (2023), 1247–1274.
- [20] LUSSIER, R. Filling gaps on the pareto front in multi-and many-objective optimization. *Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal* 9, 2 (2022), 5.
- [21] MIETTINEN, K. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.
- [22] NEWMAN, M. E. J. *Networks: An Introduction*. Oxford University Press, 2010.
- [23] PELLICER, P. V., ESCUDERO, M. I., ALZUETA, S. F., AND DEB, K. Gap finding and validation in evolutionary multi-and many-objective optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (2020), pp. 578–586.

- [24] RASMUSSEN, C. E., AND WILLIAMS, C. K. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.
- [25] SONI, J., AND BHATTACHARJEE, K. Integrating renewable energy sources and electric vehicles in dynamic economic emission dispatch: an oppositional-based equilibrium optimizer approach. *Engineering Optimization* 56, 11 (2024), 1845–1879.
- [26] SURESH, A., AND DEB, K. Machine learning based prediction of new pareto-optimal solutions from pseudo-weights. *IEEE Transactions on Evolutionary Computation* (2023).
- [27] TRAPPLER, V., HELBERT, C., AND RICHE, R. L. Multiobjective optimization under uncertainties using conditional pareto fronts. *arXiv preprint arXiv:2504.04944* (2025).
- [28] VESIKAR, Y., DEB, K., AND BLANK, J. Reference point based NSGA-III for preferred solutions. In *IEEE Symposium Series on Computational Intelligence (SSCI-2018)* (2018).
- [29] WANG, W., WANG, X., AND DONG, Z. Neural architecture search for microscopic image segmentation using a constrained multi-objective evolutionary algorithm. *Engineering Optimization* (2025), 1–20.
- [30] ZITZLER, E., DEB, K., AND THIELE, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal* 8, 2 (2000), 125–148.

A Simplex-Space Gap Detection and Region Formation Algorithm

Algorithm 1 converts the Pareto front into its pseudo-weight representation, constructs the smallest Das–Dennis reference vector set whose size exceeds that of the front, and assigns each pseudo-weight to its most similar reference vector. Reference vectors that receive no assignment form the set of unassociated directions.

Algorithm 1: PF–RV Association Pipeline

Input: Pareto front $F = \{f_i\}_{i=1}^N$, number of objectives M .
Output: Pseudo-weights W^{PF} , reference vectors W^{RV} , unassociated RV set U , lattice resolution H .

```

 $z \leftarrow \min(F)$ ; // ideal point
for  $i \leftarrow 1$  to  $N$  do
   $d_i \leftarrow f_i - z$ ;
   $w_i^{PF} \leftarrow d_i / \left(\sum_{j=1}^M d_{ij}\right)$ ;
Find smallest  $H$  such that  $\binom{H+M-1}{M-1} \geq N$ ;
Construct all integer tuples  $h$  with  $\sum h_i = H$  and set  $W^{RV} \leftarrow \{h/H\}$ ;
Normalize all  $w_i^{PF}$  and  $w_k^{RV}$ ;
Compute similarity  $S_{ik} = (w_i^{PF})^\top w_k^{RV}$ ;
for  $i \leftarrow 1$  to  $N$  do
   $A(i) \leftarrow \arg \max_k S_{ik}$ ;
 $U \leftarrow$  RVs not appearing in  $A$ ;
return  $(W^{PF}, W^{RV}, U, H)$ ;

```

Algorithm 2 examines each unassociated reference vector and determines whether it lies inside the region spanned by the observed Pareto front or outside it. This classification is performed by projecting the unassociated vector onto the ray from the corresponding PF extreme toward the appropriate simplex vertex.

Algorithm 2: Classify Unassociated Reference Vectors

Input: Unassociated RVs U , pseudo-weights W^{PF} , reference vectors W^{RV} .
Output: Inside set I , outside set O .

```

 $I \leftarrow \emptyset, O \leftarrow \emptyset$ ;
for each  $u \in U$  do
   $w \leftarrow W^{RV}[u]$ ;
   $k \leftarrow \arg \max_m w^m$ ; // closest simplex vertex
   $e_k \leftarrow$   $k$ -th simplex vertex;
   $p_k \leftarrow \arg \max_i w_{i,k}^{PF}$ ; // PF extreme in direction  $k$ 
  Compute projection:  $t = \frac{(w^* - p_k)^\top (e_k - p_k)}{\|e_k - p_k\|_2}$ 
  if  $t \leq 0$  then
     $I \leftarrow I \cup \{u\}$ ;
  else
     $O \leftarrow O \cup \{u\}$ ;
return  $(I, O)$ ;

```

Algorithm 3 groups all inside reference vectors into contiguous gap regions. The integer lattice structure induced by the Das–Dennis construction provides a natural adjacency relation, and connected components of this adjacency graph yield the final set of gap regions. The overall flow of the procedure is illustrated in Figure 21.

