Gap Finding and Validation in Evolutionary Multi- and Many-Objective Optimization

Pablo Valledor Pellicer, Miguel Iglesias Escudero,
Silvino Fernández Alzueta
Global R&D Asturias, ArcelorMittal
P.O. Box 90, Marques de Suances, Avilés, Spain
pablo.valledor-pellicer@arcelormittal.com,
{miguel.iglesias, silvino.fernandez}@arcelormittal.com

Kalyanmoy Deb
Michigan State University
428 S. Shaw Lane, 2120 EB
East Lansing, Michigan 48824,
USA
kdeb@egr.msu.edu

COIN Report Number 2020004

Abstract

Over 30 years, evolutionary multi- and many-objective optimization (EMO/EMaO) algorithms have been extensively applied to find well-distributed Pareto-optimal (PO) solutions in a single run. However, in real-world problems, the PO front may not always be a single continuous hyper-surface, rather several irregularities may exist involving disjointed surfaces, holes within the surface, or patches of mixed-dimensional surfaces. When a set of trade-off solutions are obtained by EMO/EMaO algorithms, there may exist less dense or no solutions (we refer as 'gaps') in certain parts of the front. This can happen for at least two reasons: (i) gaps naturally exist in the PO front, or (ii) no natural gaps exists, but the chosen EMO/EMaO algorithm is not able to find any solution in the apparent gaps. To make a confident judgement, we propose a three-step procedure here. First, we suggest a computational procedure to identify gaps, if any, in the EMO/EMaO-obtained PO front. Second, we propose a computational method to identify well-distributed gap-points in the gap regions. Third, we apply a focused EMO/EMaO algorithm to search for possible representative trade-off points in the gaps. We then propose two metrics to qualitatively establish whether a gap truly exists in the obtained dataset, and if yes, whether the gap naturally exists on the true Pareto-set. Procedures are supported by results on two to five-objective test problems and on a five-objective scheduling problem from a steel-making industry.

1 Introduction

Multi-objective optimization problems are increasingly being applied in several disciplines, including manufacturing industries. In particular, in recent years the interest has grown significantly in the steel industries [16, 23]. These problems involve several objective functions, incomparable, and non-uniformly scaled in their values, to be optimized simultaneously. A multi-objective problem can be formulated as follows:

$$\begin{array}{rl} \text{Minimize} & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_M(\mathbf{x}))\,, \\ \text{subject to} & \mathbf{x} \in \mathcal{S} \subset R^n, \end{array} \tag{1}$$

where $\mathbf{x}$ is an n-dimensional variable vector in the feasible design space $\mathcal{S}$, M is the number of objectives, and $f_i(\mathbf{x})$ is the i-th objective function. Problems having $M = 2$ or 3 are referred to as multi-objective optimization problems and those having $M > 3$ are called many-objective optimization problems. However, one common property of these problems is that they usually have more than one optimal solutions, which have a trade-off in their objective values and are called Pareto-optimal (PO) points.

There are two tasks in a multi- or many-objective optimization process: (i) find a well-converged and well-distributed set of Pareto-optimal solutions, and (ii) choose a single preferred solution from the set by performing a multi-criterion decision-making (MCDM) analysis. Several evolutionary multi- and many-objective optimization (EMO and EMaO) algorithms have been proposed since early nineties [6, 4]. However, not many computational algorithms have been proposed to choose a single preferred solution by the EC communities, whereas MCDM researchers have been active in devising different decision-making methods since early Seventies [17, 3]. However, finding a set of trade-off points first by an EMO or EMaO algorithm over the entire Pareto-optimal frontier allows decision-makers (DMs) to use various trade-off analysis procedures before choosing a single preferred solution.

This paper lies in the middle of the two above-mentioned tasks to ensure that the obtained set of optimized solutions covers the entire frontier, as this will have a great impact in the subsequent decision-making activities. Another reason for the importance of current study is that EMO or EMaO algorithms are stochastic and their final outcome may vary from one run to another, particularly (i) for solving a challenging problem involving nonlinear constraints, multi-modality, and isolation in the search space, and (ii) for optimizing with a low budget of solution evaluations. In many EMO/EMaO studies, an algorithm is run multiple times (30 or 50 times) from different random initial populations. For practical problem solving tasks, all obtained solutions are combined together, dominated solutions are eliminated, and resulting non-dominated solutions are reported.

Even with these careful and tedious efforts, the resulting non-dominated frontier may exhibit gaps or discontinuities. Such gaps may occur due to two reasons: (i) the true Pareto-optimal (PO) frontier contains natural gaps, hence no algorithm will be able to find any feasible non-dominated points in the gaps, or (ii) EMO/EMaO algorithms produce artificial gaps in the final set due to inefficiencies with the chosen algorithm. Before the DM proceeds to the next decision-making task, DM must ensure that it is the former reason and not the latter. This paper proposes a systematic approach for first identifying gaps, if any, in the obtained high-dimensional non-dominated front represented by a set of points, then locating a set of well-diversified gap-points in the gaps, and finally applying a focused EMO/EMaO approach in an attempt to fill the gaps. If there is no real gap in the Pareto-set, the third step will not find any point in the gaps. However, if the original EMO/EMaO algorithms could not discover any points in the gaps due to their inefficiencies, additional focused EMO/EMaO runs are expected to find points there. We present the computational steps involved in the entire process and demonstrate their use on a number of test problems and one steel industry problem.

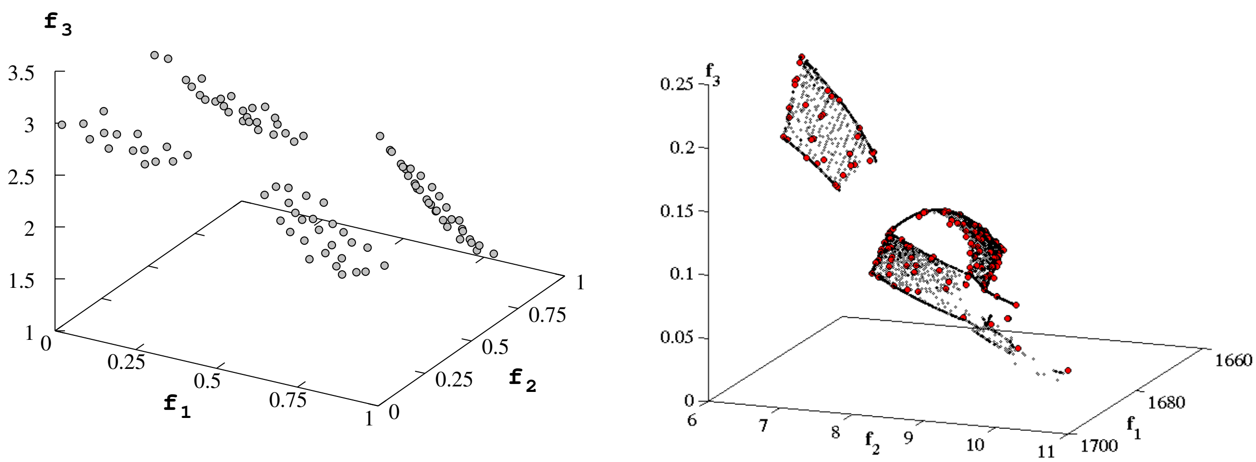
In the remainder of the paper, Section 2 defines the gap finding problem and discusses the limited past studies. Then, Section 3 presents the proposed three-step gap-investigation procedure. Section 4 presents the extensive results of the proposed method. Finally, conclusions are drawn in Section 5.

2 Gaps in Pareto-optimal Front and Past Studies

EMO and EMaO algorithms use specific *niche-preserving* operators to find and maintain a widely-distributed set of non-dominated points at the end. NSGA-II [7] emphasizes non-dominated points which have a large crowding distance – a measure providing relative distance of neighboring points from each other. This way, the extreme points and the isolated points within a non-dominated front gets a large selection advantage. SPEA2 [26] uses a clustering technique to identify and select well-sparsed points within a non-dominated front. PAES [13] uses an exclusive grid-based method which selects points that occupy least-crowded grids. MOEA/D [24], NSGA-III [8] and most recent EMaO algorithms provide equal importance to points close to a pre-defined array of well-sparsed reference directions in the objective space. Despite the use or non-use of any algorithmic parameter, most EMO and EMaO algorithms have demonstrated to find a well-distributed set of non-dominated points close to the true Pareto-optimal front.

While the whole purpose of the niche-preserving operators is to distribute EMO/EMaO population members as uniformly as possible on the entire Pareto-optimal front, this may not be always possible due to a number of reasons. First, EMO/EMaO algorithms are stochastic in nature, meaning that two different runs even from the same randomly created initial population may not find exactly the same set of final points due to inherent probability-based evolutionary operators. Moreover, for solving challenging problems involving highly nonlinear and multiple constraints and multi-modalities in the objective functions, an EMO/EMaO procedure may not reach the global Pareto-optimal front and sometimes can get stuck to a sub-optimal front. In problems having biased difficulties [14], some part of the Pareto-optimal front may be relatively easy to converge to and some other parts may be difficult to converge to. In such problems, an EMO or EMaO algorithm may find only the easy part of the PO front, exhibiting a gap or lack of points on the difficult part of the PO front.

Practical problems from industries involve nonlinear, non-convex, and challenging objective and constraint functions. The resulting Pareto-optimal fronts are not expected to be smooth and continuous. Although some test problems contain such discontinuous PO fronts, as exhibited by the three-objective DTLZ7 problem in Figure 1a, the nature and extent of gaps and discontinuities can be severe in a practical problem, as demonstrated by a three-objective car crashworthiness problem in Figure 1b. The crashworthiness problem exhibits two distinct clusters of points with a gap in



(a) DTLZ7 with four sub-fronts [10].

(b) Crashworthiness front [8].

Figure 1: Difference between a test problem and a practical problem exhibiting gaps and discontinuities.

between and a few isolated points. Also, the intermediate cluster of points exhibit a large gap in the middle and low density of points at some parts of the sub-fronts. While these features may be natural to the crashworthiness problem, they are unknown to the DMs, who will have all the reasons to doubt and question the discontinuities and gaps for their real presence in a problem, before they can continue with their subsequent decision-making activities of choosing a single preferred solution.

2.1 Why Gap Finding is Important?

The shape of the PO front provides an important piece of information to the designers and DMs. When gaps or discontinuities exist in a PO front, they would be interested in knowing the reasons for such gaps. A gap in a PO front indicates that no optimal variable vector exist in the search space with such an objective combination. Inquisitive designers and DMs would like the reason for and specialities of such objective combinations. Is it a special constraint function or the dominant nature of objective combinations at the gaps which do not qualify them to be non-dominated and feasible with the rest of the PO solutions. Although the existence of gaps within a PO front is inquisitive on its own, the points lying on the boundary of a gap are also equally interesting. They denote objective combinations which are allowed but lie in the brink of being non-existent and Pareto-optimal. They must possess certain limiting and boundary characteristics from which a slight deviation may cause the resulting solution to be a non-PO solution. While these are important revelations associated with gaps which need further analysis, the existence of gaps in a PO front first must be established with confidence, before such analysis will make sense.

There is another reason for finding gaps in a PO front from an EMO/EMaO algorithmic point of view. A gap finding methodology can reveal if an EMO/EMaO algorithm is able to find the entire PO front properly. If an EMO/EMaO algorithm exhibits a gap in its obtained front, the gap finding method can verify if the gap really exists or it is a shortcoming of the algorithm in creating artificial gaps by mistake.

2.2 Past Studies

In order to not end up finding gaps in final non-dominated fronts, some previous studies have spent efforts in careful archiving strategies so that no part of the PO front is left out [20]. One of the main targets of these techniques is to avoid large or abrupt changes in the influential objective values, reaching a preferably uniformly distributed set of non-dominated solutions.

Reference point based EMO algorithms (R-NSGA-II [9] and R-NSGA-III [22]) were developed to focus on a part of the PO front, but in principle, can be used to find gap-points. Other works are based on the interaction with the decision-maker and the use of *Bliss* points [1]. A bliss point is defined as a desired target solution by the DM. Through a sampling process, points near the non-dominated solutions are found close to the bliss points in a bi-objective optimization problem.

However, to the best of our knowledge, previous studies did not specifically deal with identifying gaps in PO fronts for multi- or many-objective problems in a systematic computational manner. We address this important task and propose a systematic procedure in the following section.

3 Proposed Gap-Investigation Method

The proposed gap-investigation method is based on three steps:

1. **Search for PO front:** In this step, we obtain a well-distributed and well-converged set ($\mathbf{F}$) of near Pareto-optimal solutions. For this purpose, we use a classical EMO or EMaO algorithm depending on the number of objectives in the problem.

2. **Identification of gaps:** In this step, we propose a novel gap-finding algorithm to identify one or more gaps in the M-dimensional EMO/EMaO obtained front $\mathcal{F}$, represented by a set of trade-off points. The gaps are represented by a set of gap-points for further analysis.

3. **Gap validation:** In this step, we apply a reference point based EMO or EMaO to focus its search near the gaps in an attempt to find Pareto-optimal points near the gaps.

The first step is to find a set of near-PO points using an EMO or EMaO algorithm, as the case may be. In this study, for the first step, we use NSGA-II and NSGA-III as an EMO and EMaO algorithm, respectively, for solving multi-objective and many-objective optimization problems. Other existing EMO and EMaO algorithms can also be used instead. The second step is the main contribution of this paper, in which gaps on the Pareto-set are identified based on the obtained EMO/EMaO points and a desired set of representative gap-points are located. We provide a detailed description of the procedure in the next subsection. If a gap is identified, the third step uses an existing reference point based EMO/EMaO methods to concentrate on finding PO points close to the gap-points, so that the existence of a natural gap in the Pareto-set can be established with confidence. For the third step, we use R-NSGA-II [9] and R-NSGA-III [22] procedures.

3.1 Identification of Gaps

The gap-finding procedure is presented in Algorithm 1. The obtained non-dominated (ND) set

Algorithm 1: Automated gap-finding procedure.

Input: Non-dominated (ND) set and desired number of gap-points K_d
Result: Set of K_d gap-points detected
Initialize K_d gap-points to an empty set;
Calculate optimal number of clusters (n_{clust}) partitioning the input ND;
Run clustering technique on ND;
Define $k_{\text{neigh}} = \min(M, n_{\text{clust}}) - 1$;
while *k_d gap-points not found and $\binom{n_{\text{clust}}}{2}$ loops not checked,* **do**
  Calculate inter-cluster distances;
  Evaluate geometric mean distance on k_{neigh} clusters;
  Sort clusters in descendent order;
  Select cluster j with maximum geometric mean distance;
  Calculate a gap-point as the average of the medoids of the selected cluster j and its k_{neigh} closest clusters;
  if *gap-point is confirmed and gap-point is not already chosen,* **then**
    Add the gap-point to the resulting list of gaps;
    Create a new cluster for the gap-point generated;
  else
    Choose the next sorted-by-distance cluster as j;
  end
end

from an EMO/EMaO algorithm and the desired number of gap-points are sent as an input to this algorithm and the algorithm returns K_d number of gap-points as output. As a first step, we cluster the M-dimensional ND set by using an unsupervised clustering procedure. We partition the search space and get the most representative solutions, calculating the medoids, that are defined as solutions whose average dissimilarity to all the solutions in the cluster is minimal [19]. To calculate the optimal number of clusters in the M-objective search space, we implement the NbClust algorithm [15]. This technique provides an ensemble of 30 indices to determine the number of clusters by applying the majority rule, and proposes the best clustering scheme from different results obtained by varying all combinations of number of clusters, distance measures, and clustering methods, mixing information about intra-cluster compactness, and inter-cluster isolation, as well as other factors, such as geometric or statistical properties of the search space.

After identifying the best number of clusters in the supplied ND set, an unsupervised machine learning clustering method is performed based on PAM (Partitioning Around Medoids) [12]. A visualization technique called clusplot [18] is run to create a bivariate plot visualizing and validating the partition of the data. All observation are represented by points in the plot, using the principal component analysis (PCA) method to understand the behavior of the clustering technique. We show, together with the clusplot chart, the variability explained by the two main components of the PCA, calculated as the aggregation of the normalized standard deviations.

Then, we generate the gap-points one by one until detecting the K_d specified gaps indicated as input of the algorithm. In order to perform this task, all inter-cluster distances are evaluated by the single linkage distance, defined as the closest distance between two samples belonging to two different clusters. The next step is to calculate for each cluster the geometric mean distance of its k_{neigh} closest clusters. k_{neigh} is defined as the minimum of the optimal number of clusters and the number of the objectives (M), minus one (to take care of effective number of gaps between clusters). The cluster having the maximum average geometric distance is then chosen as the farthest cluster j that will be impacted by the gap-point to be generated. Then, we get the medoids of cluster j and from k_{neigh} clusters close to j and calculate the average to define the final gap-point. The new gap-point is then added in the list of clusters and as a new Medoid. Finally, we loop back to the step of calculating the inter-cluster distances to generate the next desired gap-point.

An identified gap by the above procedure need not end up being a true gap due to peculiarity in the shape of the front which may fool our clustering procedure. For this purpose, we use a gap-confirmation procedure, which we describe in Section 3.2. The *while* loop is performed a maximum of $\binom{n_{\text{clust}}}{2}$ times, realizing that there can be one gap between every pair of clusters. Thus, although K_d gap-points were desired, the above procedure may end up finding K ($\leq K_d$) gap-points at the end of Algorithm 1.

3.2 Gap Validation Method

After finding the gaps and representative gap-points, we need to create some PO points near to them to validate if the gap exhibited is real or not. In order to do this gap validation step, we apply the reference-point-based EMO/EMaO methods. In particular, R-NSGA-II [9] or R-NSGA-III [22] methods are employed depending on the number of objectives and using the obtained gap-points as reference points to find R near-PO points. Both these methods allow multiple reference points to be used simultaneously, thereby finding representative PO points near all obtained gap-points.

With the purpose of validating the existence of real gaps and evaluate the performance of the original EMO/EMaO techniques used, we propose two new metrics to assess qualitatively the performance of the gap-finding method and the overall gap-investigation methodology. The metrics are given below:

1. **Gap-point to EMO/EMaO Points Distance:** For evaluating gap-finding algorithm's performance, we define a metric $d_{\text{G-EMO}}$ which is the average normalized distance of gap-points (G_i, $i = 1, \ldots, K$) from their closest EMO/EMaO points F_j, $j = 1, \ldots, N$) to the EMO/EMaO points:

$$d_{\text{G-EMO}} = \frac{\frac{1}{K}\sum_{i=1}^{K}\left(\min_{j=1}^{N}\|G_i - F_j\|_2\right)}{\frac{1}{N}\sum_{i=1}^{N}\left(\min_{j=1}^{N}\|F_i - F_j\|_2\right)}. \quad (2)$$

2. **Reference EMO/EMaO Points to Gap-point Distance:** For validating if there is really a gap, we define a metric $d_{\text{R-G}}$ which is the average distance of R-EMO/EMaO points (RF_i, $i = 1, \ldots, R$) to the gap-point compared to their distance from EMO/EMaO points:

$$d_{\text{R-G}} = \frac{1}{R}\sum_{i=1}^{R}\frac{\min_{j=1}^{K}\|RF_i - G_j\|_2}{\min_{j=1}^{N}\|RF_i - F_j\|_2}. \quad (3)$$

3.2.1 Is a Gap-point Confirmed?

To confirm whether an observed gap-point truly exist in the observed EMO/EMaO data-set, we use the first metric $d_{\text{G-EMO}}$. If this metric value is greater than one, it means that the obtained gap-point is more distance away from EMO/EMaO points than the EMO/EMaO points are from each other. In other words, the obtained gap-point can be confirmed to exist. If for a gap-point $d_{\text{G-EMO}} < 1$ is observed, the observed gap-point is too close to an EMO/EMaO point, thereby qualifying it to be an artificial gap and is more of a manifestation of our clustering algorithm.

3.2.2 Is there a Natural Gap on the PO Front?

If a gap is confirmed, the next important question is "Does the gap naturally exist on the PO front?" or "Is it a manifestation of the chosen EMO/EMaO algorithm for the problem?". We answer this question by observing the value of $d_{\mathrm{R-G}}$ and compare it with one. If $d_{\mathrm{R-G}} > 1$, this means that R-EMO/EMaO points are relatively at more distance away from the gap-point than they are from the EMO/EMaO points. Since the R-EMO/EMaO points are closer to the original EMO/EMaO points, there exists a natural gap in the Pareto-set and an additional application of R-EMO/EMaO could not find points in the gap region. On the other hand, if $d_{\mathrm{R-G}} < 1$, R-EMO/EMaO points are closer to the gap-point(s) than they are from the original EMO-EMaO points. Hence, the observed gap was artificial and the gap was the result of inefficiencies in the chosen EMO/EMaO algorithm.

4 Results

In this section, first we explain the experimental procedure followed here. Secondly, we present the parameters used for running different algorithms for the instances tested in our environment. Then we show the results of the gap-investigation procedure on each of the different problems, and finally we show the numerical results of the two performance metrics defined above.

We have chosen three different instances described in the literature under the continuous optimization category of problems, based on their different PO front shapes. In particular, we have selected ZDT3, a modified version of DTLZ2, and DTLZ7. ZDT3 [25] is a two-objective problem and has disjointed PO sets, thereby causing natural gaps in the overall PO front. DTLZ2 and DTLZ7 problems [10] are scalable to any number of objectives and have a spherical PO front shape and a constraint surface, respectively.

In addition, the gap-finding approach is tested on a real-world problem in the steel industry, described in Section 4.5. The instance is a combinatorial optimization scheduling problem with five conflicting objectives. Therefore in the experimental part we test both continuous and combinatorial problems, in multi- and many-objective environments.

In the case of ZDT3, we use NSGA-II as our first step of searching for ND solutions and R-NSGA-II as the gap-validation method, as both methods perform well for two objectives. For NSGA-II and R-NSGA-II, 92 population members are used. For R-NSGA-II, 36 reference points and an $\epsilon = 0.01$ are set.

For problems with three or more objectives, we use NSGA-III and R-NSGA-III for discovering ND solutions and for gap-validation method, respectively. Parameters are indicated in Table 1. For the Das-Dennis [5] points used in NSGA-III, $p = 12$ – the number of partitions per objective – is set to have 91 reference lines. For the real-world steel-making problem, we have used a reduced population size to complete the runs in a reasonable computational time. The parameter μ is the shrunk-factor of R-NSGA-III [22].

Table 1: Parameters used for NSGA-III and R-NSGA-III.

Problem	**Objs**	**PopSize**	**RefPoints**	**#Das-Dennis**	μ
DTLZ2	3	92	28	91	0.2
DTLZ7	3	92	36	91	0.2
	5	92	35	126	0.2
Steel-making	5	44	55	55	0.05

pymoo code [2] is used for NSGA-II [7], NSGA-III [8], R-NSGA-II [9] and R-NSGA-III [22] algorithms for solving all test problems. However, the real-world steelmaking problem is a challenging large-scale scheduling problem involving 4,000 slabs and we have used a customized NSGA-III and a customized R-NSGA-III procedures. A total of 600 generations are used for both algorithms and a total of 30 independent runs are performed for each instance to get statistically significant results. For the test problems we use the Simulated Binary Crossover (SBX) and the Polynomial Mutation operators. But as mentioned, for the steel-making problem, we have developed new customized crossover and mutation operators to deal with the specific constraints of the problem.

4.1 Two-objective ZDT3 Problem

Figure 2 shows results of ZDT3 problem with one and three desired gap-points in two independent simulations. Following our proposed methodology, the first step is to run NSGA-II algorithm to obtain a set of ND points, as shown in Figure 2c. It is clear from the figure that five disjointed sets are found by NSGA-II. Interestingly, Figure 2b shows that the maximum frequency obtained by the NbClust detection method occurs for five clusters. The respective unsupervised clustering of solutions are shown in Figure 2a. When we apply our gap-finding method for finding only one gap-point ($K_d = 1$), a gap-point (marked with a 'x') in between two disjointed sets is found, as shown in Figure 2d. When our gap-finding method is applied to three ($K_d = 3$) gap-points, three points in between consecutive ND sets are found, as shown in Figure 2e.

At this stage, let us compute our first metric to establish if there actually exist a gap in the NSGA-II set. Table 2 shows the metric value. For $K_d = 1$, we find $d_{\rm G-EMO} = 2.876$, which is larger than one. This means that NSGA-II points are further away than they are from themselves, indicating that there is at least one gap in the set. For $K_d = 3$, $d_{\rm G-EMO} = 5.316$, which is also greater than one. Now that the existence of a gap is established, let us validate if the gap is natural or artificial to the NSGA-II set.

For this purpose, in the third step, R-NSGA-II is applied from the gap-point(s). We find a few solutions (shown by blue circles) for $K = 1$ case (Figure 2d). Interestingly, no ND point is found in the gap and close to the gap-point. Instead, R-NSGA-II finds a few points on the two neighboring ND fronts and one point which is somewhat away from the gap-point. This means that the gap identified by our gap-finding algorithm (between fourth and fifth ND sets from left) on the original NSGA-II points is genuine and natural. To establish this fact, we compute the second proposed metric $d_{\rm R-G}$ and it is found to be 1969.522 and 35.897, respectively, for $K_d = 1$ and 3. Both these numbers are greater than one, meaning that R-NSGA-II points are far away from gap-point(s) than they are from NSGA-II points. This ensures that the gaps discovered by our gap-identification method in NSGA-II set in solving ZDT3 problem are genuine, as confirmed by our gap-validation method.

Based on both visual and numerical validations, we conclude that ZDT3 really possesses gaps and that no non-dominated points exists in the gaps detected by our overall procedure.

4.2 Three-objective DTLZ2 Problems: DTLZ2$^{\rm orig}$ and DTLZ2$^{\rm void}$

Now, we consider a three-objective problem (DTLZ2$^{\rm orig}$), which has a smooth PO front (positive octant of a unit sphere) without any gap in its entire domain. If our gap-investigation procedure works well, it should not be able to find any gap in the ND front. Our three-step procedure discovers that there are three artificial clusters, but it finds that no real gap in the Pareto set exists after three ($\binom{3}{2}$) failed gap-point identification tasks, all resulting in d_{G-EMO} values less than one. Since there is no gap-point found, meaning that there is no gap in the Pareto-set, R-NSGA-III is not applied.

Next, we simulate a situation where an EMO is artificially considered to have not discovered a part of the original PO front, making a void in the Pareto set. Figure 3b shows such a scenario (DTLZ2$^{\rm void}$) in which the middle part of the true PO front is undiscovered by an EMO algorithm. When this

Table 2: Results after 30 independent runs of experiments. K_d represents number of desired gap-points.

Problem	M	K_d	$d_{\mathrm{G-EMO}}$	$d_{\mathrm{R-G}}$	Gap?
ZDT3	2	1	2.876	1969.522	Yes
	2	3	5.316	35.897	Yes
DTLZ2$^{\mathrm{orig}}$	3	1	0.0006, 0.604, 0.540	–	No
DTLZ2$^{\mathrm{void}}$	3	1	1.330	0.680	No
	3	5	1.124	0.572	No
DTLZ7	3	1	3.706	9.511	Yes
	3	3	3.322	5.324	Yes
	3	5	3.510	4.750	Yes
	5	1	1.044	1.145	Yes
Steel-making	5	1	2.302	1.442	Yes
	5	3	2.430	1.688	Yes

dataset is sent to our gap-finding algorithm (Step 2 of the overall gap-investigation procedure), three clusters are found by our unsupervised approach, as shown by Figure 3a. The resulting medoids are marked in Figure 3b. With $K = 1$, the resulting gap-point is found at the center of the three clusters with $d_{\mathrm{G-EMO}} = 1.330$ (Table 2). Since $d_{\mathrm{G-EMO}} > 1$, we conclude that gap exists in the Pareto set. Next, we call R-NSGA-III in Step 3 with 28 reference points. Resulting 28 R-NSGA-III ND points are shown in Figure 3c and cause $d_{\mathrm{R-G}} = 0.680$, meaning that R-NSGA-III points are closer to the gap-point than they are to the NSGA-III points. This means that R-NSGA-III points exists in the gap, establishing that the gap identified before from the Pareto set is not natural. It is interesting to see from Figure 3c, supporting our conclusion. Our three-step gap-investigation and validation procedure are able to identify the missing ND points in the gaps.

When $K = 5$ gap-points are desired, our Step 2 find five gap-points in the gap between the three clusters. R-NSGA-III is run with 28 points for each of the five gap-points and the resulting ND points obtained in Step 3 of our overall procedure are shown in Figure 3d. This figure and $d_{\mathrm{G-EMO}} > 1$ and $d_{\mathrm{R-G}} < 1$ (in Table 2) establish that no gap exists in the originally found EMO points. This example makes it clear that our proposed gap-investigation procedure can not only validate if there is really a gap in the obtained EMO dataset, it can also help discover missing ND points in the artificial gaps, which may have not been found by the original EMO runs.

4.3 Three-objective DTLZ7 Problem

We now consider another three-objective problem – DTLZ7 – with a well-established fact of having gaps in its PO front. The existence of four patches of ND sets are shown in Figure 4.

When this dataset is sent to our gap-finding algorithm in Step 2, four clusters are detected with the medoids shown in the figure. Also, Figure 4b shows that four clusters produce optimal clustering by the NbClust method. The clustering result is shown in Figure 4a, explaining 97.56% of the point variability according to the Principal Component Analysis (PCA). Based on four ($n_{\mathrm{clust}} = 4$) and $M = 3$, we obtain $k_{\mathrm{neigh}} = 2$. Using these parameters, our gap-finding algorithm finds relevant gap-points for three different cases: $K = 1$, $K = 3$ and $K = 5$, in Figures 4d, 4e and 4f, respectively. The gap-points are marked with a 'x' in each figure.

Next, we discuss the numerical values of the first performance metric proposed in this paper. The metric values presented in Table 2 show that there are actual gaps in DTLZ7 problem with three dimensions, as the distance of $d_{\mathrm{G-EMO}}$ is larger than one (3.706 for $K_d = 1$, 3.322 for $K_d = 3$ and 3.510 for $K_d = 5$), meaning that gap-points are away from NSGA-III points. They clearly indicate that there are gaps in the NSGA-III ND front.

To establish if these gaps are genuine, in Step 3, we apply R-NSGA-III for each of the three cases. Obtained points are shown in blue circles in Figures 4d and 4e, and 4f, respectively. In all cases, it is visually clear that no ND point is found in the gaps marked by the gap-points. In fact, DTLZ7 problem has natural gaps in between the four clusters of ND points and our gap-investigation methodology with three steps is able to verify this aspect visually through all sub-figures in Figure 4. In addition, no closest solution (solutions marked in red color in Figures 4d, 4e and 4f) is found by R-NSGA-III.

Our second proposed metric (d_{R-G}) is now computed for all three cases. Since $d_{\mathrm{R-G}}$ values are greater than one (9.511, 5.324, and 4.750 for $K = 1$, 3, and 5, respectively), they indicate that R-NSGA-III points are closer to NSGA-III points than they are from the gap-point(s). This means that the R-NSGA-III points are not in the gaps, but in the midst of the NSGA-III points. This establishes that three-objective DTLZ7 problem may have natural gaps in its Pareto-optimal front.

4.4 Five-objective DTLZ7 Problem

Next, to investigate the scalability of our proposed gap-investigation procedure, we apply it to five-objective DTLZ7 problem. The problem is redefined so that there exists only one gap (instead of five gaps as in the three-objective case shown earlier) in the entire PO set. For the five-objective DTLZ7 problem, eight clusters are detected as described by NbClust method in a parallel coordinate plot (PCP) in Figure 5a. On two principal components, the clusters are marked in Figure 5a. Although shown in two principal directions, there is no big gap visually observed. We specify to find $K = 1$ gap-point and Figure 5b shows the resulting gap-point in the PCP. It is interesting to note that there are not many similarly patterned NSGA-III lines as the gap-point in the PCP, meaning that may exist a gap in the obtained NSGA-III dataset. Figure 5c also shows the gap-point with a 'x' in a space with the first three objectives. In Step 3, 35 R-NSGA-III points (in blue circles) are discovered, as shown in the figure. To make these R-NSGA-III points clear, we show them on a separate PCP in Figure 5d. We compute $d_{\mathrm{G-EMO}}$ to establish if there is a gap in the Pareto-set. Since $d_{\mathrm{G-EMO}}$ is 1.044 (> 1), there exists a gap.

It is interesting to observe how the 35 obtained R-NSGA-III points obtained using the gap-point are closer to the NSGA-III lines in the figure. Although the gap-point line (in black) is different from the NSGA-III line patterns, blue lines (R-NSGA-III points) take similar patterns as that of NSGA-III lines (shaded lines). To estbalish if this is a natural gap using our proposed metric, we compute d_{R-G}, which is 1.145 (> 1). This means that R-NSGA-III points are closer to NSGA-III points than they are from the gap-point, establishing that there is a natural gap in the Pareto-set of the five-objective DTLZ7 problem.

It is important to note that with increasing dimensions, more points are needed to sample the five-dimensional space. The distance metrics described here also depends on the chosen shrink factor μ of R-NSGA-III algorithm, dictating the spread of R-NSGA-III solutions. Better metrics must be designed to capture the gap validation event for higher dimensional problems.

4.5 Five-objective Slab Scheduling Problem from the Steel Industry

The above test problems were defined on the continuous space and extracted from the existing EMO literature. Now, we demonstrate the working of our proposed gap-investigation procedure on a real-world combinatorial scheduling problem involving a large number (4,000) of slabs. The problem statement is similar to the galvanising line scheduling problem, described in [11].

The specific case to solve is a slab scheduling optimization problem, where n slabs (pieces of steel with rectangular cross section) are to be selected from the slab yard of the plant, and then sequenced in the hot strip mill to produce the coils (a finished steel product coiled after the rolling process) demanded by the customers or the next process stages downstream in a manufacturing plant (like for pickling or galvanizing lines). The problem can be understood analogous to a permutation flowshop scheduling problem with specific constraints defined by the schedulers of the mill and five conflicting objectives [21]. We minimize the makespan, total weighted tardiness, production cost, maintenance operations, and overall scheduling constraint violations. Due to the large-scale nature of the problem and a quick-time solution requirement, we have devised a customized NSGA-III algorithm to solve this problem taking into account the particularities of the plant and the business knowledge from the scheduling department.

Figure 6 shows the results of the slab scheduling problem. NbClust detects five optimal clusters (Figure 6b) that can be clearly seen through the clusplot chart 6a. Step 2 of our overall procedure is applied two times ($K = 1$ gap-points, shown in Figure 6d and $K = 3$ gap-points, shown in Figure 6f). From graphical results one can be enticed to infer that there are gaps in the obtained customized NSGA-III solutions, mainly caused by differences in objectives f_4 and f_5 in the PCP. Resulting customized R-NSGA-III solutions are marked in blue lines for both $K = 1$ and $K = 3$ cases.

Our performance metric $d_{\text{G}-\text{EMO}}$ demonstrates that there is a gap in the Pareto-set, by having a value greater than one. Moreover, $d_{R-G} > 1$ indicates that the gap is natural. Similar observations ($d_{\text{G}-\text{EMO}} = 2.430$ and $d_{R-G} = 1.688$ for $K = 3$ desired gap-points) indicate that there is a natural gap in the obtained non-dominated set of points.

5 Conclusions and Future Work

In this paper, we have proposed a three-step methodology to find and validate gaps in a PO front for two to five-objective problems. A new technique, based on unsupervised machine learning has been introduced to find a set of user-defined gap-points. Focused evolutionary multi- and many-objective optimization algorithms have been used to detect if any non-dominated points exist in the gaps identified by our procedure. A new metric has been proposed to confirm if the predicted gap truly exist in the non-dominated set and a second metric is proposed to validate the existence of the gap in the Pareto-set. On a few two to five-objective test problems, correct outcome about the presence or absence of gaps in their respective PO fronts has been established by our proposed procedures. On a real-world slab scheduling problem involving 4,000 variables, our procedure has also predicted the presence of gaps in the obtained five-dimensional non-dominated front.

The future work will be focused on extending our approach to (i) find multiple gap-points simultaneously using more sophisticated algorithms, (ii) enable capabilities to dynamically adapt and find the requisite number of gap-points, and (iii) apply the approach to more complex and larger-dimensional problems to fully evaluate its merit.

Acknowledgement: The work was supported by ArcelorMittal Global R&D. Authors acknowledge discussions with Julian Blank.

References

[1] Hussein Abbass. An inexpensive cognitive approach for bi-objective optimization using bliss points and interaction. volume 3242, pages 712–721, 09 2004.

[2] J. Blank and K. Deb. pymoo - Multi-objective Optimization in Python. https://pymoo.org, 2019.

[3] V. Chankong and Y. Y. Haimes. *Multiobjective Decision Making Theory and Methodology.* New York: North-Holland, 1983.

[4] C. A. C. Coello, D. A. VanVeldhuizen, and G. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Boston, MA: Kluwer, 2002.

[5] I. Das and J.E. Dennis. An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection. In *Proceedings of the Third World Congress on Structutal and Multidisciplinary Optimization*, 1999.

[6] K. Deb. *Multi-objective optimization using evolutionary algorithms*. Wiley, Chichester, UK, 2001.

[7] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[8] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, 2014.

[9] K. Deb, J. Sundar, N. Uday, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research (IJCIR)*, 2(6):273–286, 2006.

[10] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002)*, pages 825–830, 2002.

[11] S. Fernandez, S. Alvarez, D. Díaz, M. Iglesias, and B. Ena. Scheduling a galvanizing line by ant colony optimization. In *Proceedings of the Swarm Intelligence: 9th International Conference, ANTS 2014, Brussels, Belgium*, pages 146–157, 2014.

[12] Leonard Kaufman and Peter J. Rousseeuw. Clustering by means of medoids, 1987.

[13] J. D. Knowles and D. W. Corne. Approximating the non-dominated front using the Pareto archived evolution strategy. *Evolutionary Computation Journal*, 8(2):149–172, 2000.

[14] H. Li, Q. Zhang, and J. Deng. Biased multiobjective optimization and decomposition algorithm. *IEEE transactions on cybernetics*, 47(1):52–66, 2017.

[15] V. Boiteau M. Charrad, N. Ghazzali and A. Niknafs. Nbclust: An r package for determining the relevant number of clusters in a data set. *Journal of Statistical Software*, 61, 2014.

[16] Alessandro Maddaloni, Giacomo Filippo Porzio, Gianluca Nastasi, Valentina Colla, and Teresa Branca. Multi-objective optimization applied to retrofit analysis: A case study for the iron and steel industry. *Applied Thermal Engineering*, 91:638–646, 12 2015.

[17] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer, Boston, 1999.

[18] G. Pison, A. Struyf, and P.J. Rousseeuw. Displaying a clustering with clusplot. *Computational Statistics and Data Analysis*, 30:381–392, 1999.

[19] Peter Rousseeuw, Mia Hubert, and Anja Struyf. Clustering in an object-oriented environment. *Journal of Statistical Software*, 01, 02 1997.

[20] O. Schutze, M. Laumanns, E. Tantar, C. A. C. Coello, and E.-G. Talbi. Computing gap-free Pareto front approximations with stochastic search algorithms. *Evolutionary Computation*, 18(1):65–96, 2010.

[21] Pablo Valledor, Alberto Gomez, Paolo Priore, and Javier Puente. Solving multi-objective rescheduling problems in dynamic permutation flow shop environments with disruptions. *International Journal of Production Research*, 56(19):6363–6377, 2018.

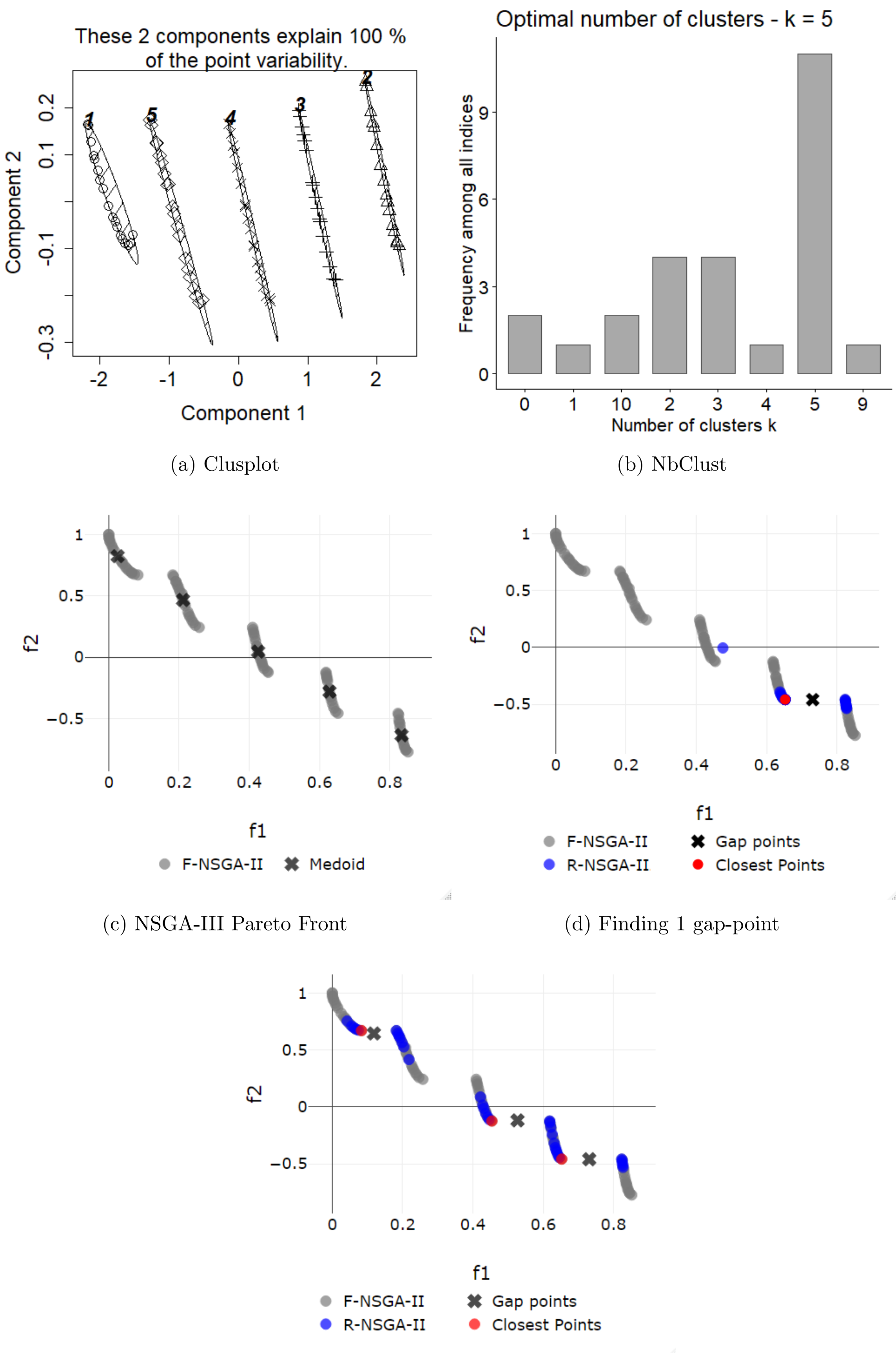
[22] Y. Vesikar, K. Deb, and J. Blank. Reference point based NSGA-III for preferred solutions. In *IEEE Symposium Series on Computational Intelligence (SSCI-2018)*, 2018.

[23] Jianping Yang, Bailin Wang, Caoyun Zou, Xiang Li, Tieke Li, and Qing Liu. Optimal charge planning model of steelmaking based on multi-objective evolutionary algorithm. *Metals*, 8:483, 06 2018.

[24] Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.

[25] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation Journal*, 8(2):125–148, 2000.

[26] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In K. C. Giannakoglou et al., editor, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2001.



(a) Clusplot

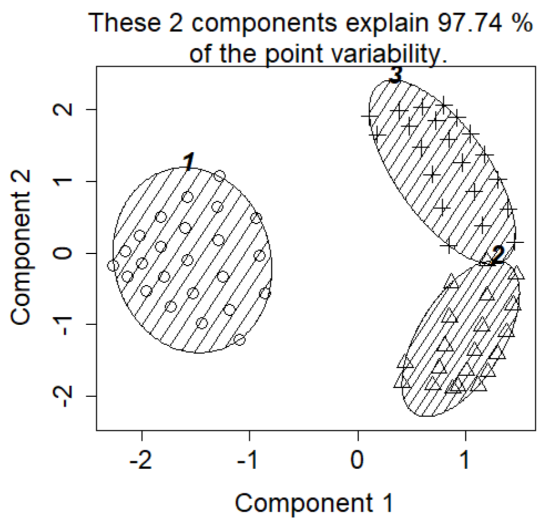
(b) NbClust

(c) NSGA-III Pareto Front

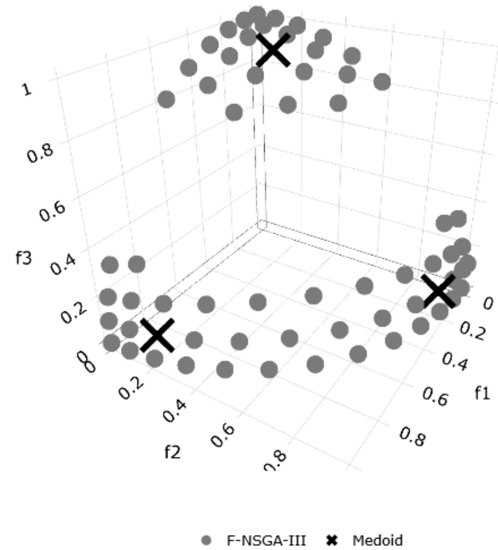
(d) Finding 1 gap-point

(e) Finding 3 gap-points

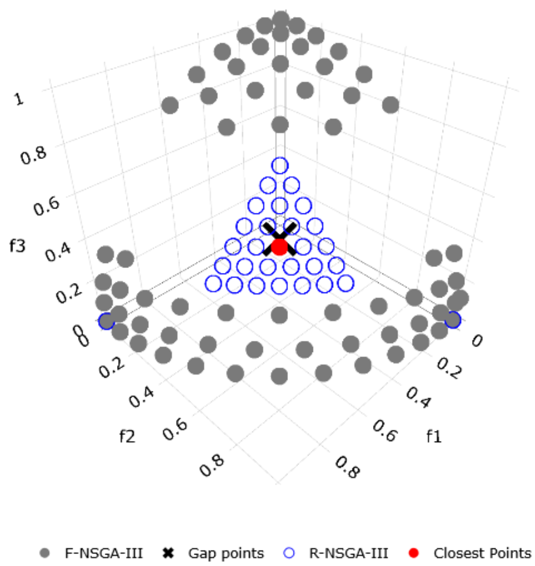
Figure 2: Results on two-objective ZDT3 problem.



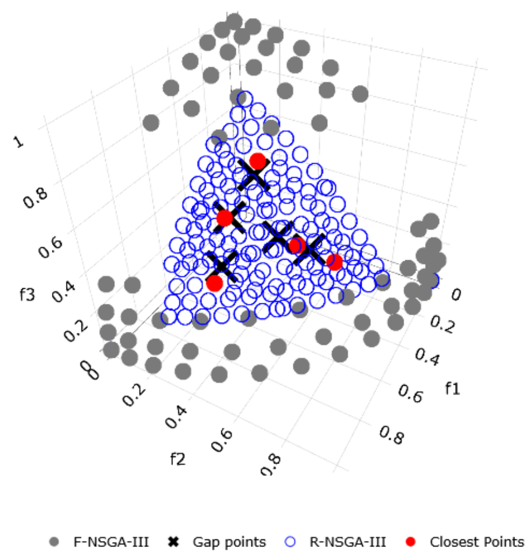
(a) Clusplot



(b) NSGA-III Pareto Front

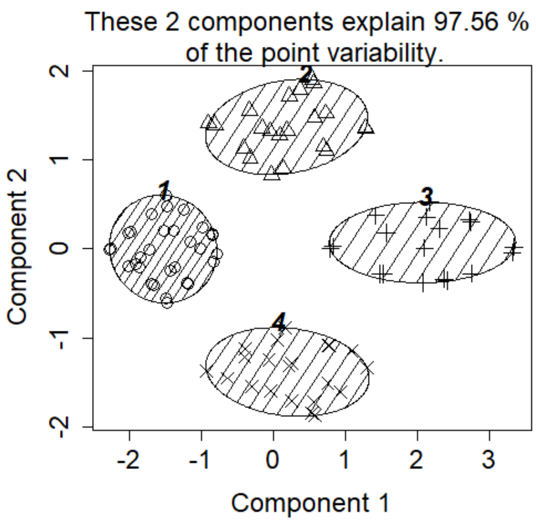


(c) Finding 1 gap-point

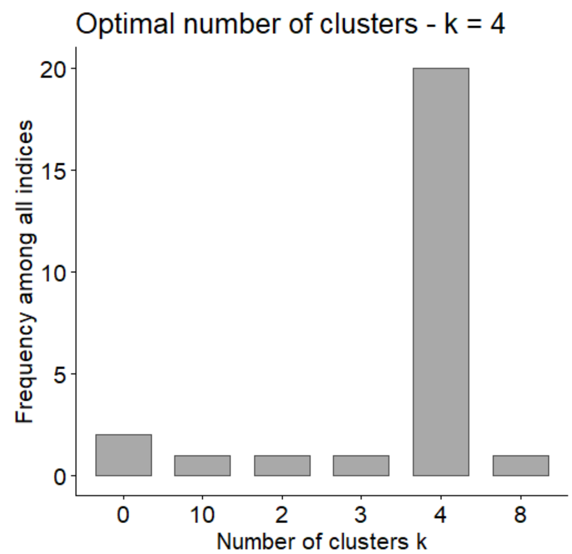


(d) Finding 5 gap-points

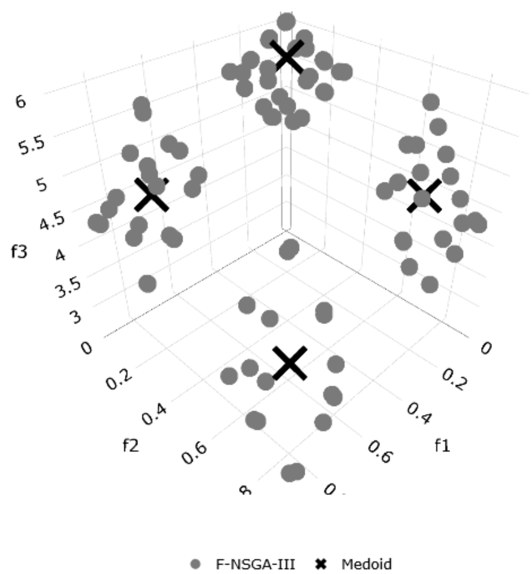
Figure 3: Results on modified three-obj. DTLZ2 problem.



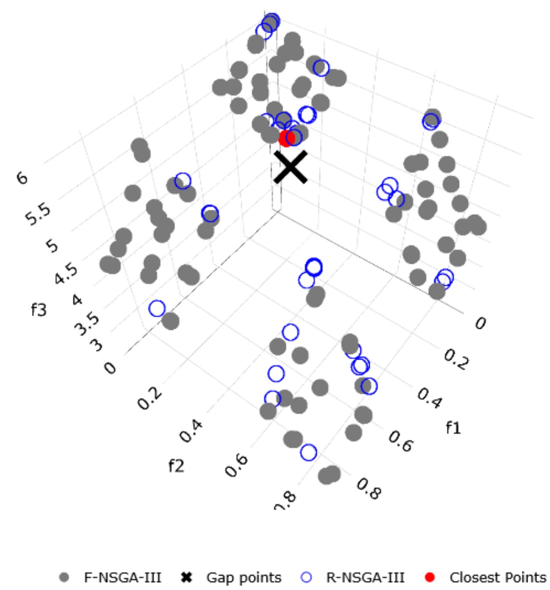
(a) Clusplot



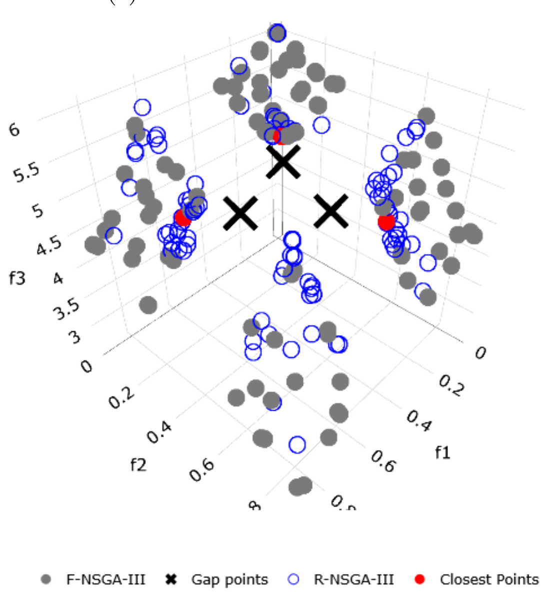
(b) NbClust



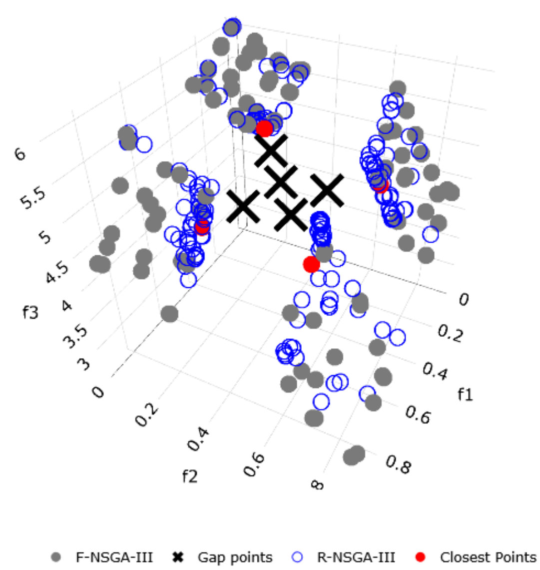
(c) NSGA-III Pareto Front



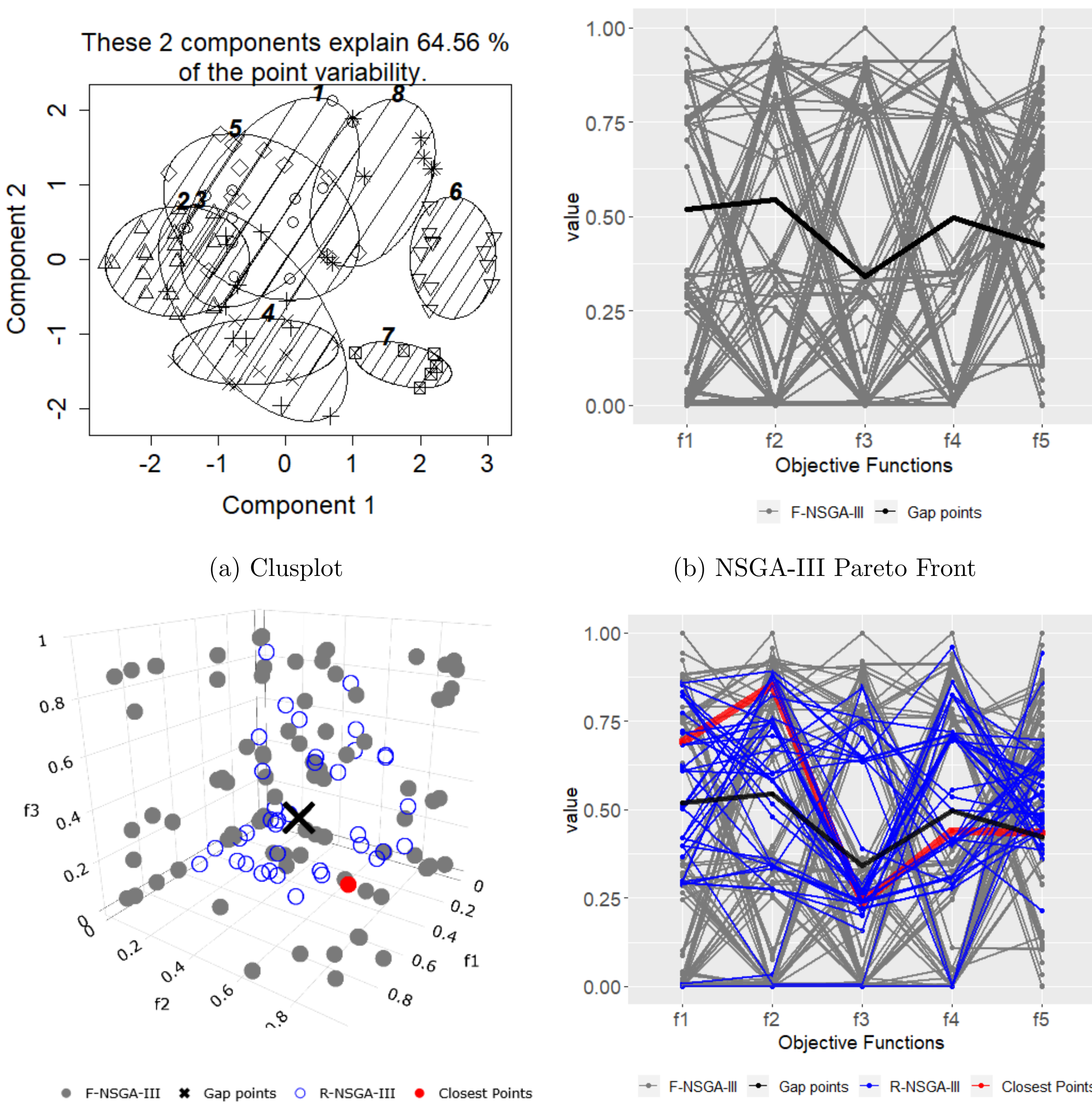
(d) Finding 1 gap-point



(e) Finding 3 gap-points



(f) Finding 5 gap-points



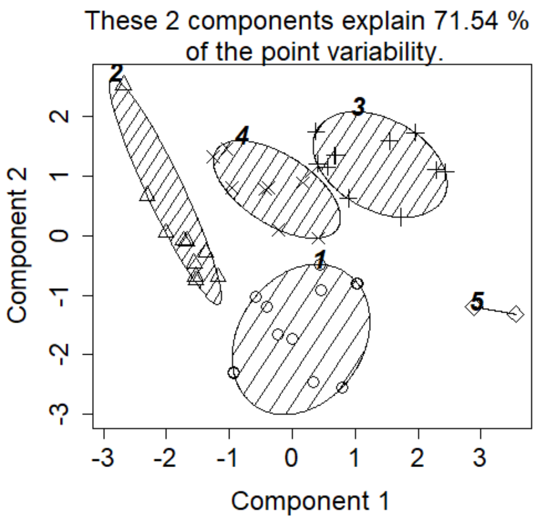
(a) Clusplot

(b) NSGA-III Pareto Front

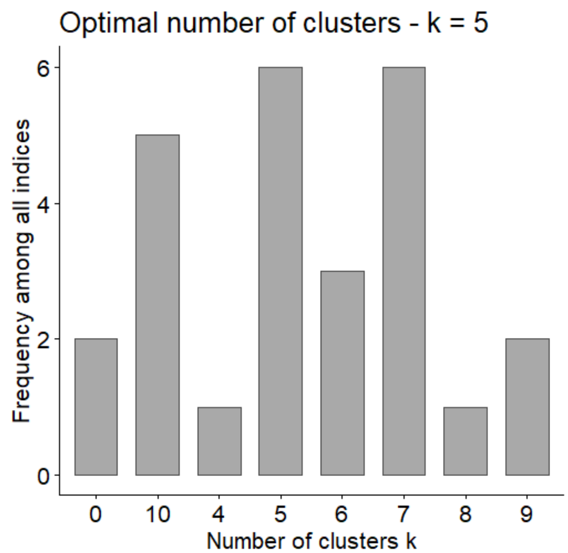
(c) 1 gap-point in 3 dimensions

(d) R-NSGA-III points

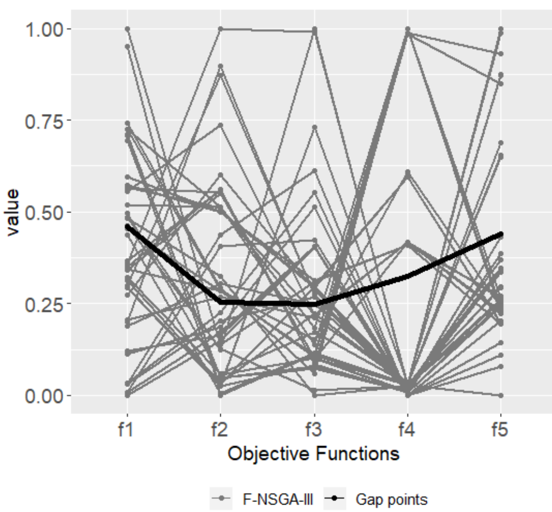
Figure 5: Results on five-objective DTLZ7 problem.



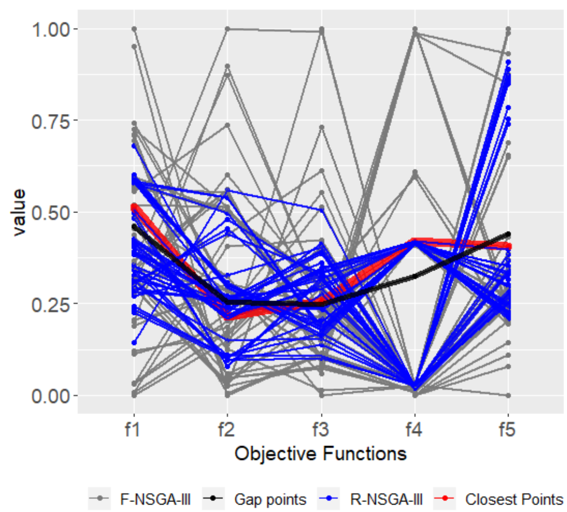
(a) Clusplot



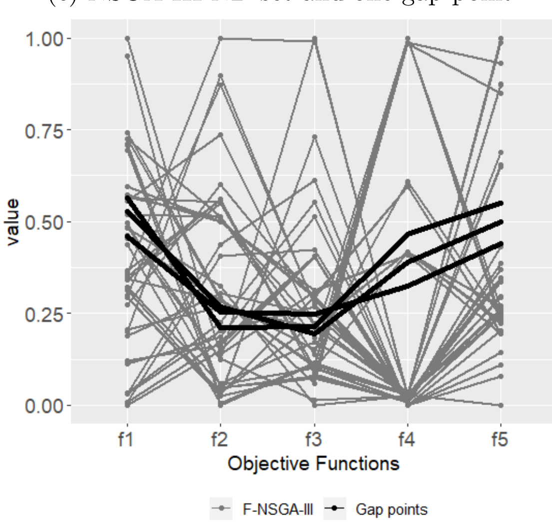
(b) NbClust



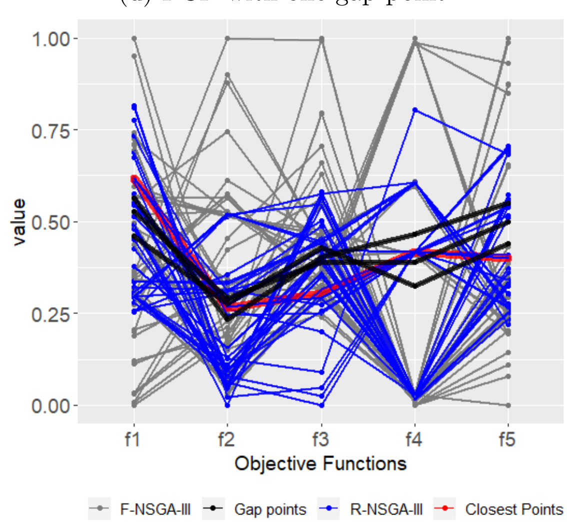
(c) NSGA-III ND set and one gap-point



(d) PCP with one gap-point



(e) NSGA-III ND set and three gap-points



(f) PCP with three gap-points

Figure 6: Results on the five-objective real-world steelmaking problem.