

This work is licensed under a [Creative Commons “Attribution-NonCommercial-ShareAlike 4.0 International”](https://creativecommons.org/licenses/by-nc-sa/4.0/) license.



# Filling Gaps on the Pareto Front in Multi- and Many-Objective Optimization

Richard Lussier

lussi036@morris.umn.edu

Division of Science and Mathematics

University of Minnesota, Morris

Morris, Minnesota, USA

## Abstract

Pareto fronts offer insight into the best found solutions of a given problem. Several algorithms have been developed to help maintain a well-distributed Pareto front and therefore offer a wide variety of solutions. However, in real-world problems, the Pareto front isn't necessarily a continuous surface and may contain holes and/or discontinuous lines. These irregular areas on the Pareto front are considered *gaps*. These gaps can either be *natural* or *artificial*. In their research, Pellicer, Escudero, Alzueta, and Deb suggest a three-step procedure to find, validate, and fill these gaps. First, they developed an algorithm to generate gap points. Second, they developed an equation that checks the validity of the gap points. Lastly, a focused algorithm looks for points around the validated gap points to determine if they are natural or artificial. This procedure was used on several test problems that range from two-to-five objectives, including a five-objective real-world problem in the steel industry.

**Keywords:** Gap Validation, Pareto front, NSGA, R-NSGA, Evolutionary Multi-Objective Optimization (EMO), Evolutionary Many-Objective Optimization (EMaO)

## 1 Introduction

There are problems that require you to optimize more than one objective. For example, while creating a school system, a decision needs to be made on how much will be spent on schools in order to provide a short transportation time for students. The two objectives in this problem are to minimize school construction cost and to minimize the average student transit time. On one hand, you could build only one school to limit costs but therefore cause students to have a higher average travel time. On the other hand, you could have a higher cost and build many schools so students have a low average travel time. Then, there are solutions in-between these two extremes. A visual of these points can be seen in figure 1(a). Algorithms are used to find solutions to these problems. While solutions are being generated, it's important to find as many points (potential solutions) as possible to have a wide variety of good solutions. The best found solutions are considered part of the *Pareto front*.

While attempting to find the best points, *gaps* can appear. A *gap* is considered as an area on the Pareto front where

the algorithm failed to find a point. In the school example, you can imagine a figure such as 1(b). In this figure, there is a visible *gap* in the Pareto front. These gaps need to be *validated* to ensure they're in an empty area on the Pareto front. Gaps are then classified as either *natural* or *artificial*. For the purposes of this paper, a gap is considered *natural* if a point is unable to be found inside of the gap after a secondary search by a more focused algorithm. If this secondary algorithm successfully found a point in the gap, we consider the gap *artificial*.

This paper focuses on finding gaps and filling them when possible to provide a wider variety of optimized points. In section 2, I give the background information and terminology needed to understand evolutionary computation. In section 3, I explain the four evolutionary algorithms that are used in Pellicer et al. In section 4, I elaborate on the method developed by Pellicer et al. In section 5, I summarize the results found by Pellicer et al. Finally, in section 6, I explain further possible improvements to this methodology and the real-world applications used by Ergezer [3]. [4]

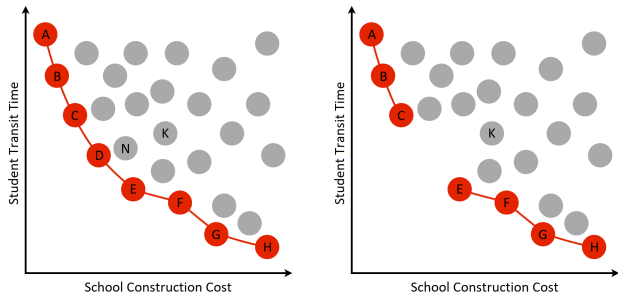
## 2 Background

### 2.1 Pareto Front

The Pareto front is the set of points that are non-dominated or incomparable to all other points on the graph [5]. A point  $p_1$  *dominates*  $p_2$  if all values in  $p_1 \geq p_2$ . Each value corresponds to an objective, meaning a two-objective problem would be a 2D point. Therefore, a *non-dominated* point  $p$  dominates all other points it can be compared to. Incomparable suggests that you are unable to compare some points to others and therefore you cannot determine which is better. An example of a Pareto front can be seen in figure 1(a). In this figure, the Pareto front is composed on points A-H. Additionally, points K and N are not in the Pareto front because points C-E optimized the objectives in a more efficient way. Points A-H are also considered *Pareto-optimal*.

### 2.2 Evolutionary Computation

Evolutionary Computation (EC) is the process of solving a problem using guided search and simulated biological evolution. An *individual* is a single potential solution. In Pellicer et al[4], individuals are a vector of numbers. Each number



(a) An example Pareto front for the school example. (b) An example of a Pareto front with a gap.

**Figure 1.** Solutions for the school construction example. Points A-H are part of the Pareto front.

represents the value assigned to an objective through a series of trade-offs. These values are not combined together. Rather, they serve as a way to measure which points are non-dominated. A *population* is a group of these individuals. Each *generation* has its own population and generations progress until the algorithm has gone through as many generations as specified by the user. *Offspring* are created by applying crossover: combining properties of two individuals to each other to create a new individual, and mutation: changing part of a single individual with a new random value. [6]

### 2.3 Evolutionary Multi- and Many-Objective Optimization

Evolutionary multi-objective optimization (EMO) and many-objective optimization (EMaO) involve running an evolutionary algorithm on a set of objectives that you would like to optimize. Multi-objective means there are 2-3 objectives whereas many-objective is for more than three objectives. Pellicer et al [4] utilize two types of algorithms, an algorithm that performs a broad search in the problem space and a focused algorithm and focuses on a specified part of the problem space. These algorithms are outlined in section 3. An example of an EMO problem is the school example mentioned in section 1. In this example, you can imagine all of the points this algorithm generates to be plotted on a 2D plane where one axis is the cost spent on schools and the other axis is the travel time for students. The main idea is that these evolutionary algorithms will generate these points on the plane. Among these points you can find the Pareto front.

### 2.4 Gaps: Natural vs. Artificial

Gaps can be classified as one of two things: *natural* or *artificial*. Before a gap can be classified, it needs to be *validated*. A gap being *validated* means that it is a gap in the Pareto front and is deemed insufficiently searched. Once a gap is validated, it can be *natural* or *artificial*.

A gap is considered *natural* when the focused search algorithm was unable to fill the gap. Gaps that are classified as natural implies there are real discontinuities in the problem and there are spaces in the problem space where a solution is impossible. Additionally, when the focused search algorithm does fill the gap, the gap is considered *artificial* and the broad search algorithm simply didn't search that area of the Pareto front thoroughly enough. Classifying a gap point as artificial means that additional points were found on the Pareto front and therefore gives more Pareto-optimal solutions to choose from. Therefore, we prefer gaps to be *artificial* since it exposes us to more Pareto-optimal solutions.

## 3 Four EMO/EMaO Algorithms

Pellicer et al [4] utilize four different EMO/EMaO algorithms to accomplish different tasks. This section introduces the four EMO/EMaO algorithms used in their research: NSGA-II, NSGA-III, R-NSGA-II, and R-NSGA-III.

### 3.1 NSGA Variants

*Non-dominated sorting genetic algorithm* (NSGA) has two variants used in Pellicer et al [4]: NSGA-II and NSGA-III. NSGA-II is used for multi-objective optimization and NSGA-III is used for many-objective optimization. These algorithms provide a broad search for the problem. In other words, they search the entire problem space and return a set of solutions that approximate the Pareto front. NSGA algorithms provide well-distributed points. However, gaps can still remain, which need a different algorithm to focus on that specific area around the gap point.

### 3.2 R-NSGA Variants

*Reference point-based non-dominated sorting genetic algorithm* (R-NSGA) also has two variants that were used in Pellicer et al [4]: R-NSGA-II and R-NSGA-III. R-NSGA-II is used for multi-objective optimization and R-NSGA-III is used for many-objective optimization. Instead of searching the entire problem space, these algorithms apply a focused search. This means that they excel at finding points in a concentrated location. This is done by using *reference points*. *Reference points* are one or more points supplied by the user that act as a marker as to where the user wants to find additional points. An example of a reference point can be seen in figure 2. In this figure, R-NSGA prefers the grey circles over the black "x"s due to them being closer to the reference point. Pellicer et al [4] inserts these reference points in the vicinity of validated gap points in order to apply a focused search around them. Therefore, a new set of Pareto front points are returned. These new points will hopefully help fill the previously mentioned gaps.

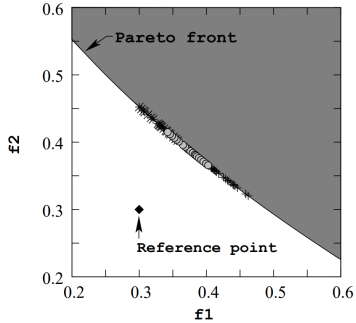


Figure 2. An example of how reference points are used [1].

## 4 Methodology

This section outlines the importance of the R-NSGA variants and how they are used to fill gaps in the Pareto front by Pellicer et al [4]. The procedure is broken down into three steps:

1. Generate a near Pareto-optimal solution using either NSGA-II or NSGA-III.
2. Use an algorithm to detect potential gaps inside of the Pareto front and validate them.
3. If gaps are validated in the previous step, run either R-NSGA-II or R-NSGA-III around the gap point(s) found in the previous step to search for additional points on the Pareto front. Reference points are selected as various points on/around the gap point(s).

### 4.1 Generating a Near Pareto-optimal Solution

In order to find gaps on the Pareto front, you need a Pareto front to search. This is done by generating near Pareto-optimal solutions using NSGA-II and NSGA-III on the test problems described in section 5.

### 4.2 Gap Finding Algorithm

Before we can apply the R-NSGA algorithms to gaps, we need a way to identify gaps. In other words, we need to know where the densest areas are on the Pareto front so we can check the empty areas. Algorithm 1 outlines the algorithm Pellicer et al [4] created to find potential gaps and validate them.

This algorithm takes a set of non-dominated points,  $ND$ , and an integer,  $g$ , representing the number of gap points you want to find. A *gap point* is considered a point where there are no NSGA Pareto front points close to it. Once the optimal number of clusters (groups) is found, we initialize  $k_{neigh} = \min(M, n_{clust}) - 1$ . Here,  $M$  is the number of objectives in the problem and  $n_{clust}$  is the optimal number of clusters found by the clustering technique. This is the number of gaps associated with each cluster [4]. For example, in figure 3(a), it was determined that there are five clusters ( $n_{clust} = 5$ ). However, there are only two objectives ( $M = 2$ ). Therefore,  $k_{neigh} = 1$ . In other words, each cluster is determined to have

one neighbor and therefore one potential gap with it. This is supported by figure 3(a) where there is visually one gap associated with each side-by-side pairing.

$\binom{n_{clust}}{2}$  is the total number of pairs between clusters. This notation is defined below:

$$\binom{n_{clust}}{2} = \frac{n_{clust}!}{2!(n_{clust} - 2)!}$$

For example, in figure 3(a), we found  $n_{clust} = 5$ . Therefore, we find that  $\binom{5}{2} = 10$ . This means that there are 10 total combinations of clusters and we want to check a maximum of 10 times.

After the clusters are organized in descending order, the most isolated cluster being first, a gap point needs to be generated based on the selected cluster  $j$  and its closest  $k_{neigh}$  neighbors. The gap point is made by averaging the *medoid* of cluster  $j$  and its  $k_{neigh}$  neighbors. A *medoid* of a cluster is defined as the point that keeps dissimilarities between all points in the cluster to a minimum.

### 4.3 Validating and Classifying Gaps

Once potential gap points have been found, they need to be validated and then be classified as *natural* or *artificial*. Recall that a gap being *natural* means that a point did not fill the gap. In contrast, an *artificial* gap is one that was filled by using an R-NSGA algorithm.

**4.3.1 Gap Validation.** After potential gap points have been found, there needs to be a way to validate if they are gaps. This is done with equation 1:

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

Equation 1 calculates  $d_{G-EMO}$ . The numerator averages the distance between each potential gap point in the set  $K_d(G_i, i = 1, \dots, K)$  and the Pareto front point  $(F_j, j = 1, \dots, N)$  closest to each potential gap point.  $K$  is the number of gap points inside of  $K_d$  and  $N$  is the number of points in the set of original Pareto front points,  $ND$ . The denominator calculates the average distance the Pareto front points are from each other. If  $d_{G-EMO} > 1$ , then the distance between the gap point(s) and the Pareto front points is greater than the distance the Pareto front points are from each other. Therefore, the gap is valid. An example of validated gaps can be seen in figure 3(b), where there are visibly three gap points. On the other hand, when  $d_{G-EMO} < 1$ , it means the gap point(s) are closer to the Pareto front points than the Pareto front points are from each other, meaning the gap isn't valid. An invalid gap shows that that location does not need to be searched further and is sufficiently filled. [4]

**4.3.2 Is the Gap Natural?** Now that the algorithm has validated gaps, R-NSGA will be run and gaps can be classified as natural or artificial. This is determined by equation 2:

---

**Algorithm 1:** Gap point identification (based on Pellicer et al [4]).

---

**Input** :Set of non-dominated points  $ND$ , number of desired gap points  $g$ .

**Output**:Set of gap points up to size  $g$ .

- 1  $K_d$  is initialized to an empty set of gap points;
- 2 Calculate the optimal number of clusters  $n_{clust}$  for  $ND$  and run a clustering algorithm on  $ND$ .
- 3 Define  $k_{neigh} = \min(M, n_{clust}) - 1$ , which determines how many gaps between clusters there are. Recall that  $M$  is the number of objectives the problem has;
- 4 **while**  $g$  gap points have not been found and  $\binom{n_{clust}}{2}$  loops haven't been checked, **do**
- 5 For each pair of clusters, calculate the distance between the two closest points of the pair;
- 6 Evaluate the average distance that each cluster is from the  $k_{neigh}$  closest clusters to it;
- 7 Sort clusters in descending order so the most isolated cluster is first;
- 8 Select cluster  $j$  as the most isolated cluster;
- 9 **do**
- 10 Calculate a gap point as the average of the medoids for  $j$  and its  $k_{neigh}$  closest clusters;
- 11 **if** gap point is validated (see section 4.3.1) and doesn't already exist in  $K_d$  **then**
- 12 Add cluster  $j$  into  $K_d$  and create a new cluster for  $j$ ;
- 13 Go to line 4;
- 14 **else**
- 15 Choose the next sorted cluster as  $j$ ;
- 16 Go to line 10;
- 17 **end**
- 18 **while** Gap point  $j$  has not been validated;
- 19 **end**

---

$$d_{R-G} = \frac{1}{R} \sum_{i=1}^R \frac{\min_{j=1}^K \|RF_i - G_j\|_2}{\min_{j=1}^N \|RF_i - F_j\|_2} \quad (2)$$

$d_{R-G}$  is the average of a ratio. The numerator is the distance that the closest R-NSGA point ( $RF_i, i = 1, \dots, R$ ) is from each gap point. The denominator is the shortest distance each R-NSGA point is from an original NSGA Pareto front point. Therefore, if  $d_{R-G} < 1$ , that means that the R-NSGA points are (on average) closer to the gap points than they are to the NSGA points. This classifies the gap as artificial and is considered filled. Vice versa, when  $d_{R-G} > 1$ , it means the R-NSGA points are closer to the Pareto front points (on average) than they are to the gap points. This classifies the gap as natural. A natural gap means that no point can exist in that location of the problem space. [4]

## 5 Results

This section explains the test problems used by Pellicer et al [4] and the results they found. In all of the figures provided, such as figure 3(b), the grey dots (F-NSGA) correspond to the original Pareto front of the given problem. This is generated using an NSGA algorithm. Black "x"s can correspond to two things: a medoid or a gap point. In images containing just the Pareto front such as figure 3(a), the black "x" corresponds to the medoids. In images that contain R-NSGA points (figure 3(b)), the black "x" corresponds to gap points. Additionally, blue bots correspond to the Pareto front points found by R-NSGA. Red dots signify the closest R-NSGA point to the gap point(s) (black "x").

### 5.1 ZDT3 Results

ZDT3 is a two-objective test problem that was created by Zitzler et al [7]. ZDT3 has a discontinuous Pareto front as seen in figure 3(a). Pellicer et al [4] attempts to fill these discontinuities by using R-NSGA-II to fill validated gaps.

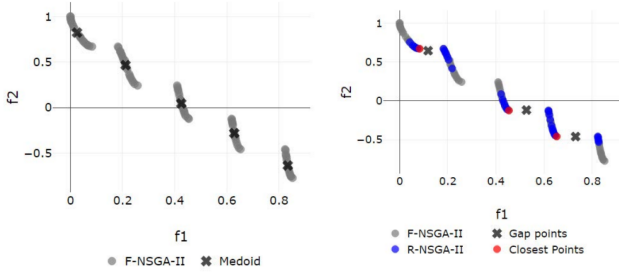
There are distinct breaks in the graph as seen in figure 3(a). In this figure,  $f_1$  and  $f_2$  represent the two objectives that ZDT3 is trying to minimize. Next, they utilized algorithm 1 to find one gap point and three gap points.

The results of equation 1 ( $d_{G-EMO}$ ) and equation 2 ( $d_{R-G}$ ) can be seen in table 1. When one gap point was being detected,  $d_{G-EMO} = 2.876$ . Since  $d_{G-EMO} > 1$ , this shows that the found gap point is farther away from its average NSGA-II point than the average distance NSGA-II points are from each other. Since the gap is now validated, we can run R-NSGA-II around the gap point to try to fill it. Once R-NSGA-II was run,  $d_{R-G} = 1929.522$ . This number is greater than 1, which signifies the R-NSGA-II points are closer to the original NSGA-II points than the closest R-NSGA-II point is to the gap point. Moreover, this means that the gap is considered to be natural. Recall that a natural gap point implies that no point can exist in that portion of the problem space. A graph for these results can be found in Pellicer et al [4]. In other words, R-NSGA-II found points already found by NSGA-II and could not find additional points in the gap provided.

Additionally, ZDT3 was tested on three gap points. These three points are visualized on figure 3(b). The  $d_{G-EMO}$  value of these three gap points was 5.316, which is over 1 and therefore validates the gap points. Once the R-NSGA-II points are generated,  $d_{R-G} = 35.897$ . Since this value is greater than 1, the R-NSGA-II points are closer to NSGA-II points than they are from the gap points. Therefore, the gaps are considered natural. As seen in figure 3(b), all three gap points are visibly not filled with R-NSGA-II "rediscovering" preexisting NSGA-II points.

### 5.2 DTLZ2 Results

DTLZ problems were created by Deb et al [2]. DTLZ2 is a three-objective test problem that has a quarter spherical



(a) Pareto front generated by the (b) Three gap points and the R-NSGA-II points generated.

**Figure 3.** Results for ZDT3 by Pellicer et al [4].

Problem	$M$	$K_d$	$d_{G-EMO}$	$d_{R-G}$
ZDT3	2	1	2.876	1969.522
	2	3	5.316	35.897
DTLZ2 <sup>orig</sup>	3	1	0.0006, 0.604, 0.540	—
DTLZ2 <sup>void</sup>	3	1	1.330	0.680
	3	5	1.124	0.572
DTLZ7	3	1	3.706	9.511
	3	3	3.322	5.324
	3	5	3.510	4.750
	5	1	1.044	1.145
Steel-Industry	5	1	2.302	1.442
	5	3	2.430	1.688

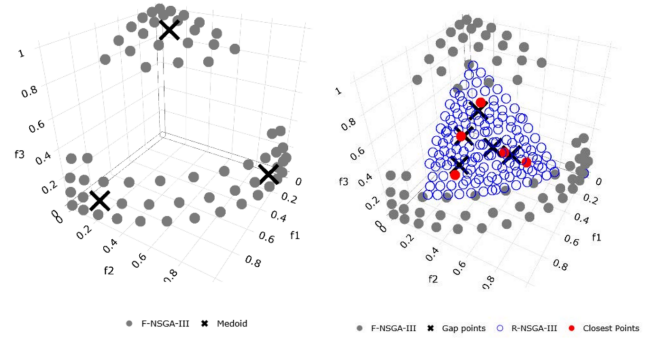
**Table 1.** Results from Pellicer et al [4] for all test problems.

Pareto front. DTLZ2 is used in two different forms in this paper: DTLZ2<sup>orig</sup> and DTLZ2<sup>void</sup> [4]. DTLZ2<sup>orig</sup> is the original Pareto front with no gaps. This acts as a sanity check for algorithm 1. DTLZ2<sup>void</sup> is the Pareto front with points intentionally removed from the center as seen in figure 4(a). Therefore, Pellicer et al [4] expected to find artificial gaps.

DTLZ7 is a five-objective test problem that consists of four patches in its Pareto front. Both DTLZ2 and DTLZ7 use NSGA-III for their Pareto fronts and R-NSGA-III for filling validated gaps.

For DTLZ2<sup>orig</sup>, algorithm 1 was run three times in order to ensure the result was correct.  $d_{G-EMO}$  was calculated to be 0.0006, 0.604, and 0.540 for each of the three runs. Since this value is less than 1, the gaps were too close to the generated NSGA-III points to be considered gaps. Therefore, the gaps aren't valid and R-NSGA-II was never run.

For DTLZ2<sup>void</sup>, Pellicer et al [4] chose to run their algorithm twice: once to locate one gap point and the second time to locate five gap points. For the one gap point, which can be seen in Pellicer et al [4],  $d_{G-EMO} = 1.330$ , meaning that the gap point was validated. Once R-NSGA-III points were generated,  $d_{R-G}$  was found to be 0.680. Since  $d_{R-G} < 1$ , the R-NSGA-III points were closer to the gap points than



(a) The Pareto front generated by (b) Five gap points and the R-NSGA-III points generated.

**Figure 4.** Results for DTLZ2<sup>void</sup> found by Pellicer et al [4].

they were to the NSGA-III points. In this case, the gap is artificial and was able to be filled. This is important because it shows that the original Pareto front missed Pareto-optimal solutions.

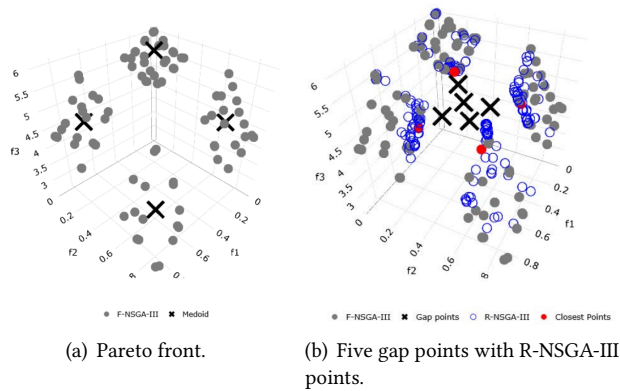
In the case of the five gap points, which are visualized in figure 4(b), Pellicer et al [4] found  $d_{G-EMO} = 1.124$ , which means the gap points have been validated and R-NSGA-III can be used. Then, using equation 2,  $d_{R-G}$  was found to be 0.572, meaning the R-NSGA-III points are closer to the gap points than they are to the NSGA points. Therefore, the gaps are artificial and were filled. Artificial gaps signify that we found previously undiscovered points.

### 5.3 DTLZ7 Results

Pellicer et al [4] ran DTLZ7 as both a three-objective and a five-objective problem. The three-objective Pareto front can be seen in figure 5(a). The five-objective Pareto front and the three-objective one and three gap point figures can be seen in Pellicer et al [4]. As shown in table 1, DTLZ7 is observed with one, three, and five gap points when it has three objectives and one gap point when there are five objectives.

For one gap point, Pellicer et al [4] received a  $d_{G-EMO} = 3.706$ , validating the gap. The obtained  $d_{R-G}$  was 9.511. Since  $d_{R-G} > 1$ , the gap is considered to be natural. Similarly, when searching for three gaps,  $d_{G-EMO} = 3.322$  and  $d_{R-G} = 5.324$ . Therefore, all three gaps are considered to be natural. Finally, with five gaps (visualized in figure 5(b)),  $d_{G-EMO}$  was found to be 3.510 and  $d_{R-G}$  was found to be 4.750. Therefore, the five gaps are also considered to be natural.

Pellicer et al [4] also tested five objectives to observe the scalability of algorithm 1. This test was conducted with one gap point, which resulted in  $d_{G-EMO} = 1.044$ . Since this is greater than 1, the gap is validated. The graphs can be found in Pellicer et al [4]. Then, Pellicer et al [4] utilize equation 2 to measure whether the R-NSGA-III points filled the gap. With a  $d_{R-G}$  value of 1.145, the gap is considered natural.



**Figure 5.** Results for DTLZ7 found by Pellicer et al [4].

## 5.4 Steel Industry Results

This is a real-world problem in the steel industry. This problem involves cutting slabs of steel, transforming them into coils, and distributing the coils to the customer. The problem contains five objectives to optimize, which are [4]:

1. Minimize the time it takes to complete the order.
2. Prioritize more important projects and ones with a closer deadline than others.
3. The production cost of the project.
4. The cost for maintaining the equipment.
5. Minimize inconvenient schedules such as being understaffed and working on holidays.

Pellicer et al [4] searched for one gap point and three gap points. For the singular gap point, Pellicer et al [4] found  $d_{G-EMO}$  to be 2.302. Once R-NSGA-III was ran,  $d_{R-G} = 1.442$ . this means the gap is natural and was not filled. When three gap points were located, Pellicer et al [4] found a  $d_{G-EMO}$  value of 2.430. Once R-NSGA-III points were generated,  $d_{R-G} = 1.688$ . Since  $d_{R-G} > 1$ , the gap wasn't filled and is considered to be natural. These results suggest that there are real discontinuities in this problem and there are solutions to this problem that simply aren't feasible or possible. All figures for this problem can be found in Pellicer et al [4].

## 6 Conclusion

In this paper, I outlined the research of Pellicer et al [4] and their three-step methodology for detecting, validating, and filling gaps on the Pareto front. Pellicer et al [4] tested this methodology on four test problems and found interesting results. Gaps were validated and were found to be both natural and artificial.

One area where Pellicer et al's [4] methodology is being applied is the development of drones carrying a slung-load. These drones have the challenge of avoiding obstacles along their path. A slung-load is a piece of cargo that is hanging down from the drone by some form of rope. With this

new methodology, Ergezer was able to identify gaps and conclude that the gaps were natural and could not be filled. This allowed Ergezer to ensure he had the complete Pareto front. [3]

Based on their current success, Pellicer et al hope to improve on their current work in three aspects [4]:

1. Find multiple gap points simultaneously using a more sophisticated algorithm.
2. Develop a way for algorithm 1 to dynamically adapt and find the optimal number of clusters.
3. Apply this methodology to higher dimension problems to further test scalability.

Note that in each of the test problems, all of the validated gaps were found to be natural except for  $DTLZ2^{void}$ , where points were intentionally removed from its Pareto front. This highlights a big question. Are gaps in most cases indeed natural and NSGA is a good algorithm for finding all of the best possible points? Or, are there problems where NSGA produces artificial gaps and R-NSGA is able to fill them?

## Acknowledgments

I would like to thank my advisor Nicholas McPhee for his advice and suggestions. I would also like to thank the course instructors, Elena Machkasova and K.K. Lamberty, for their guidance throughout the course. Lastly, I would like to thank Leonid Scott for his detailed feedback.

## References

- [1] Kalyanmoy Deb and J. Sundar. 2006. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (Seattle, Washington, USA) (GECCO '06)*. Association for Computing Machinery, New York, NY, USA, 635–642. <https://doi.org/10.1145/1143997.1144112>
- [2] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. 2002. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, Vol. 1. 825–830 vol.1. <https://doi.org/10.1109/CEC.2002.1007032>
- [3] Halit Ergezer. 2021. Multi-Objective Trajectory Planning for Slung-Load Quadrotor System. *IEEE Access* 9 (2021), 155003–155017. <https://doi.org/10.1109/ACCESS.2021.3129265>
- [4] Pablo Valledor Pellicer, Miguel Iglesias Escudero, Silvino Fernández Alzueta, and Kalyanmoy Deb. 2020. Gap Finding and Validation in Evolutionary Multi- and Many-Objective Optimization. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference (Cancún, Mexico) (GECCO '20)*. Association for Computing Machinery, New York, NY, USA, 578–586. <https://doi.org/10.1145/3377930.3389835>
- [5] Wikipedia contributors. 2021. Pareto front — Wikipedia, The Free Encyclopedia. [https://en.wikipedia.org/w/index.php?title=Pareto\\_front&oldid=1057242200](https://en.wikipedia.org/w/index.php?title=Pareto_front&oldid=1057242200) [Online; accessed 12-February-2022].
- [6] Wikipedia contributors. 2022. Evolutionary computation — Wikipedia, The Free Encyclopedia. [https://en.wikipedia.org/w/index.php?title=Evolutionary\\_computation&oldid=1068699276](https://en.wikipedia.org/w/index.php?title=Evolutionary_computation&oldid=1068699276) [Online; accessed 2-March-2022].
- [7] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. 2000. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* 8, 2 (June 2000), 173–195. <https://doi.org/10.1162/106365600568202>