PALETTEVIZ: A METHOD FOR VISUALIZATION OF HIGH-DIMENSIONAL PARETO-OPTIMAL FRONT AND ITS APPLICATIONS TO MULTI-CRITERIA DECISION MAKING AND ANALYSIS

By

AKM Khaled Ahsan Talukder

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2022

ABSTRACT

PALETTEVIZ: A METHOD FOR VISUALIZATION OF HIGH-DIMENSIONAL PARETO-OPTIMAL FRONT AND ITS APPLICATIONS TO MULTI-CRITERIA DECISION MAKING AND ANALYSIS

By

AKM Khaled Ahsan Talukder

Visual representation of a many-objective Pareto-optimal front in four or more dimensional objective space requires a large number of data points. Moreover, choosing a single point from a large set even with certain preference information is problematic, as it imposes a large cognitive burden on the decision-makers. Therefore, many-objective optimization and decision-making practitioners have been interested in effective visualization methods to enable them to filter down a large set to a few critical points for further analysis. Most existing visualization methods are borrowed from other data analytics domains and they are too generic to be effective for many-criterion decision making. In this dissertation, we propose a visualization method, using star-coordinate and radial visualization plots, for effectively visualizing many-objective trade-off solutions. The proposed method respects some basic topological, geometric and functional decision-making properties of high-dimensional trade-off points mapped to a three-dimensional space. We call this method *Palette Visualization (PaletteViz)*. We demonstrate the use of PaletteViz on a number of large-dimensional multi-objective optimization test problems and three real-world multi-objective problems, where one of them has 10 objective and 16 constraint functions. We also show the uses of *NIMBUS* and *Pareto Race* concepts from canonical multi-criterion decision making and analysis literature and introduce them into PaletteViz to demonstrate the ease and advantage of the proposed method.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

CHAPTER 1

INTRODUCTION

With the current advancement in many/multi-objective evolutionary optimization algorithms for solving problems with four or more conflicting objective functions, an array of post-optimization or interactive-optimization issues now must be addressed on an immediate basis, especially if those algorithms are to be regularly used in practice. One of the most challenging issues is to visually comprehend the obtained non-dominated points in the objective space[1] produced by a many-objective optimization solver, so that a few critical points can be filtered out quickly by the decision-makers (DMs). If the optimization problem consists of two or three objectives, a two or three-dimensional scatter plot is the most intuitive way to illustrate and analyze the Pareto set. In such cases, DMs can easily locate and isolate critical points that correspond to most beneficial trade-offs among objectives (in terms of least gain to most sacrifice in comparison to the neighboring points). DMs can also easily visualize other preference criteria in lower dimensions, such as robustness or reliability of a point, or some other utility functions representing the practical usefulness of a solution. In this thesis we will try to address this challenging problem: we will try to devise a novel visualization framework that can provide many utilities that naturally come with a generic scatter-plot-like visualization method. We call our method *Palette Visualization*, or in short, *PaletteViz*.

1.1 Multi-objective Optimization and Pareto-optimal Front

As PaletteViz is designed to visualize high-dimensional Pareto-optimal fronts, we think it is necessary to formally define some of the key concepts of multi/many-objective[2] optimization problems (MOP).

[1] Here we loosely call them interchangeably, as 'Pareto set' or 'Pareto-optimal front' or simply 'Pareto-front'

[2] In the evolutionary multi-objective optimization (EMO) domain, problems with more than three objectives are generally referred to as "many-objective optimization problems", whereas up to three-objectives, they are known as "multi-objective optimization problem".

1.1.1 Multi-objective Optimization Problem (MOP)

A multi-objective optimization problem can be formulated as follows:

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})]^T \\ \text{subject to,} \quad & g_j(\mathbf{x}) \geq 0, \quad \forall j \in \{1, 2, \dots, k\} \\ \text{where,} \quad & \mathbf{x} \in \Omega \subseteq \mathbb{R}^n \text{ and, } \mathbf{f} \in \Lambda \subseteq \mathbb{R}^m \end{aligned} \tag{1.1}$$

Here f_i can be differentiable or non-differentiable functions. Ω is known as *design variable space*, Λ is known as *objective function space* and $\mathbf{f} : \Omega \to \Lambda$. $g(\cdot)$ are *constraint functions* and without loss of generality, feasibility is defined using $\geq$ relation. Ω is the feasible design variable space and Λ is the feasible objective space. A *Pareto-dominance* relation for this optimization problem can be defined as follows:

Definition 1.1. (Pareto-dominance) A solution $\mathbf{x} \in \Omega$ *Pareto-dominates* another point $\mathbf{y} \in \Omega$ if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})\ \forall i \in \{1, 2, \dots, m\}$ and $f_i(\mathbf{x}) < f_i(\mathbf{y})\ \exists\, i \in \{1, 2, \dots, m\}$. We denote this as relation as $f_i(\mathbf{x}) \prec f_i(\mathbf{y})$.

Now we can formally define *Pareto-optimal Front* as follows:

Definition 1.2. (Pareto-optimal Front) A solution $\mathbf{x}^* \in \mathcal{PS} \subset \Omega$ is Pareto-optimal if there is no other solution $\mathbf{x} \in \Omega$ such that $\forall i : f_i(\mathbf{x}) \prec f_i(\mathbf{x}^*)$. A Pareto-optimal front $\mathcal{PF}$ is the set of all points in Λ that are mapped from all such $\mathbf{x}^* \in \mathcal{PS}$, i.e. $\mathcal{PF} \subset \Lambda : \mathbf{f}(\mathbf{x}^*) \,|\, \forall \mathbf{x}^* \in \mathcal{PS}$. $\mathcal{PS}$ is called the Pareto-optimal set.

An illustration of objective space, variable space and mapping is presented in Figure 1.1 The mapping between variable and objective space is presented in a two-objective (f_1 and f_2) three-variable (x_1, x_2 and x_3) problem. The Pareto-front (red line in objective space) and Pareto-set (red in variable space) are optimal solutions in objective and variable space, respectively. In this thesis, our goal is to visualize $\mathcal{PF}$; furthermore, we call $\mathcal{PF}$ the *Pareto-optimal* "data point cloud" $\mathcal{C}$ in $\mathbb{R}^m$. PaletteViz creates visualizations of $\mathcal{C}$.

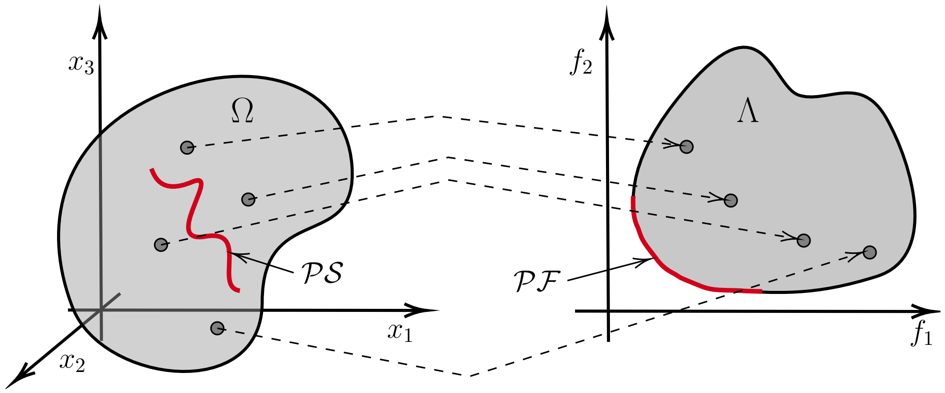


Figure 1.1: Mapping from design variable space Ω to multi-objective function space Λ, where $n = 3$ and $m = 2$.

1.2 Existing Examples

A good visualization technique should enable a graphical exploration of the Pareto set and should help the DM to obtain better insights into the problem [3]. Such visual comparisons must go not only beyond their dominance relationships to each other, but also include various practical *functional* properties of every data point, such as, a safe distance from the *constraint boundaries* to avoid the risk of being infeasible, a good trade-off benefit compared with their neighboring points, closeness to the *boundary* of the overall or an individual cluster of the high-dimensional Pareto set, robustness/reliability of a point due to uncertainties in the problem, etc. Moreover, the graphical representation of the Pareto set in a two- or at most three-dimensional space must be clear as well as relatively easy to be comprehended, interpreted, and manipulated by the DMs. Besides, the visualization mechanism should also preserve the structural properties of the original high-dimensional data-set as accurately as possible to the lower-dimensional visualization space. While DMs may be interested in such a visualization mainly to filter out a few selected critical points for further considerations, a visualization technique may also consider the DM's preferences directly, such as specification of a reference or an aspiration point [4, 5], relative importance of one objective over the others [6, 7], or other utility-based information [8].

While new procedures and advancements have still been being made in multi-criterion decision analysis (MCDA)[3] literature for various ways of analyzing large-dimensional and non-dominated data-sets since the early '70s, the evolutionary multi-objective optimization (EMO) community has been mainly concentrating on developing efficient algorithms for searching and finding multiple high-dimensional and non-dominated points. Besides the practical significance of developing efficient and convenient visualization techniques for large-dimensional data-sets, EMO researchers (and for this matter, all computational intelligence (CI) researchers are interested in multi-objective problem-solving tasks) wonder if their algorithmic acumen can be utilized somehow in coming up with effective visualization techniques. Our work makes an attempt in this direction and suggests a visualization technique which may perform a post-optimal computational operation by utilizing sensible geometric and decision-making properties (refer to Section 2.2) of a large, non-dominated data-set, before approaching the DMs. The hope is to reduce the large cardinality of points in the objective space by filtering through geometric, functional, and preferential decomposition approaches. This will then allow DMs to analyze fewer solutions using more DM-specific preference information in finally choosing a single preferred solution.

1.3 A Brief Taxonomy of Visual Analysis Methods for Multi-criteria Decision Making

There exist a number of different visualization techniques in the literature which are widely used in EMO and MCDM domains: parallel coordinate plot (PCP) [9], radial visualization or Radviz [10], multiple scatter plots [8], heatmap [11], self-organizing maps [12], and others. A number of recent studies [13, 14] have clearly demonstrated this aspect. While most of these methods allow high-dimensional data to be rendered in a two-dimensional mapped space, they are not able to convey many of the geometric and structural, functional, and decision-making properties (refer to Section 2.2). Due to the lack of such an MCDM-based visualization technique, existing methods are used, but it has been always

[3] Or, multi/many-criteria decision making (MCDA)

questionable whether those methods provide an effective and convenient way to choose a preferred solution.

In this thesis, we propose a novel visualization technique – a *palette* visualization (PaletteViz) – which allows a DM to look at a large number of non-dominated data points (in the objective space) in a functionally decomposed manner so that a few critical and preferential points can be isolated quickly for further analysis. The proposed visualization technique maps the data points onto a three-dimensional space that resembles a set of artist's palettes stacked on top of each other where the similar colors on the palettes resemble similar functional properties of the data points in their neighborhoods – hence the name. The PaletteViz technique is demonstrated on a number of different Pareto-optimal data points to show its usefulness. Wherever necessary, the plots are also compared with a few other existing visualization methods.

1.4 Summary of Contributions

Although we are going to discuss everything in detail in the later chapters of this thesis, we would like to summarize some key contributions of our work.

- The central contribution of this thesis is the introduction of computational topology concepts into the existing high-dimensional data visualization techniques – especially, in terms of its applications in multi-criterion decision making. To the best of our knowledge, the proposed approach is the first of such kind.

- Introduced a novel visualization framework called PaletteViz that can map the m-dimensional *shape* induced by $\mathcal{C}$ onto two- and three-dimensional space. As a result, the visualization method can work like a generic scatter plot or a bubble chart where a user can now *see* the data points in a high-dimensional ($m > 4$) space.

- Introduced a novel idea of constructing level sets based on *alpha*-hull. Demonstrated with a number of practical examples that the α-hull-based level-set can be a good way to visualize the shape of the high-dimensional space realized by $\mathcal{C}$.

- Introduced a set of new benchmark multi-objective optimization problems that can artificially create different challenging, sometimes topologically degenerate manifolds on the Pareto-optimal data point cloud $\mathcal{C}$. The new benchmark problems were used to assess the applicability and identifying the limitations of the proposed visualization method.

- Integrated the PaletteViz method into canonical interactive MDCM routines, namely, with NIMBUS [15] method and Pareto-race [16]. This implementation shows that PaletteViz can come as a great aid to decision makers (DMs), especially when geometric, functional and preferential information are not adequate (or missing) in the existing visualization method.

- Utilized PaletteViz to visualize $\mathbf{H}_{m-1}$ homology group boundary of an m-dimensional spherical Pareto-optimal front. This is a proof-of-concept demonstration to establish a bridge between MDCDM and topological data analysis (TDA) domain. Thus making PaletteViz a novel way to visualize level sets of a Pareto-optimal data point cloud $\mathcal{C}$.

- Developed a set of computational libraries to implement PaletteVizṪhe library is used to conduct different analyses on the resultant visualization framework. The library also implements the proposed new benchmark MOPs.

- Developed a functional web-application that showcases many examples of the visualization and analyses conducted in this thesis.

- Finally, introduced a new way to see the problem of data visualization, especially when the number of dimensions is not prohibitively huge and the user or DM needs to manipulate the high-dimensional data points in two-/three-dimensional space.

1.5 Organization of the Thesis

This section concludes the first chapter of the thesis. The rest of the dissertation is organized as follows – in Chapter 2, we discuss the motivation behind our development. We describe and analyze the existing high-dimensional Pareto-optimal front visualization methods and argue how they are not particularly suitable for MCDM use-cases. We also show how many of them fail to provide basic visual cues when it comes to faithful data visualization.

In the next chapter (i.e., Chapter 3), we delve into the theoretical backgrounds on the development of the PaletteViz method. We try to investigate what is the best way to map a high-dimensional space onto a two-/three-dimensional space while maintaining a set of key topological and decision making requirements. We also discuss some basic ideas on computational topology and show how everything falls together. We also discuss why an ideal visualization method should respect some human cognitive aspects and limitations.

Next, in Chapter 4, we introduce the PaletteViz method. Here we discuss in great detail how the proposed method is implemented and how it works. The algorithmic discussions are also provided in this chapter. We also discuss how the PaletteViz method is capable of reducing the limitations posed by the existing methods, along with other salient issues. After this, in Chapter 5, we apply PaletteViz on a set of newly proposed MOPs. We showcase different scenarios, especially in terms of presenting different MCDM analysis criteria. We also discuss how PaletteViz will behave in presence of quirks in topology, along with some degenerate cases. At the end of this chapter, we demonstrate how PaletteViz can be utilized in real-world engineering optimization problems, especially when there are 10 dimensions and 16 constraint functions. We also provide a short summary and a chart on comparative analysis with different existing visualization methods.

In Chapter 6, we demonstrate real-world MCDM applications of the PaletteViz method, namely with interactive MCDM routines like NIMBUS and Pareto-race. We show how PaletteViz can be of great help when it comes to meticulous visual analytics. Then we show

how PaletteViz can also be utilized in the computational topology domain, by demonstrating that PaletteViz can be used to visualize $\mathbf{H}_{m-1}$ homology group boundaries in m-dimensional space.

Next, in Chapter 7, we showcase a web-application that encapsulates a number of capabilities provided by the PaletteViz method. We discuss different functionalities and UI interaction. In Chapter 8, we conclude our thesis with a short review on what we have done so far and present a number of interesting directions (improvements, extensions) for future research.

CHAPTER 2

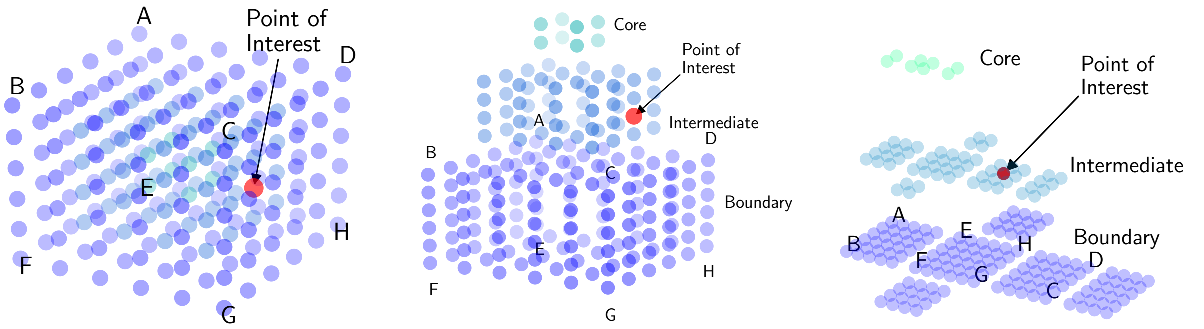
WHY ANOTHER VISUALIZATION METHOD?

Most existing visualization methods used in EMO studies today are borrowed from the general high-dimensional data visualization literature. From our previous experience with industry collaborations, we have observed that a decision making procedure requires a few unique functional considerations which the usual high-dimensional data visualization methods may not have considered to be important. Thus, it is time that Evolutionary Multi-objective Optimization (EMO) researchers develop new and more useful techniques for visualizing the trade-offs among data points in a more topologically consistent way, rather than simply following existing methods from the literature. Therefore, the next important question is what are these unique functionalities that a multi-objective data visualization technique must have, so that only a few meaningful and relevant Pareto-optimal points can be presented to DMs for a faster and worthy decision analysis.

2.1 Motivation: A Topological Perspective

Before going into the details, let us start our discussion by posing a simple scenario. Let us assume we have a set of data points in a three-dimensional space. They are arranged in a cube on a regularly spaced grid, which is illustrated in Figure 2.1a[1]. Also assume there is a data point in the set that is deemed as a "point of interest" and is marked in red. As the data points are laid on a regular grid, we can assume that the cube is composed of three smaller nested cubes, like that of a *Matroyeshka doll*. The point of interest is located in the intermediate cube and this is illustrated in Figure 2.1b. A user might be interested in analyzing this point, its location in the data set and how this point is related to its neighbors or which layer of cube it is on. If we could find a consistent way to open up or *flatten* each nested cube, we will be able to see an arrangement presented in Figure 2.1c. Here, the bottom

[1] All the plots in this dissertation are best visualized in color.



(a) Points in a nested cube. (b) Three interleaving cubes. (c) Each layer flattened.

Figure 2.1: (a) An example three-dimensional data set organized in three nested cubes on a regular grid. (b) The "point of interest" is located on the intermediate cube. (c) The red point can be examined more closely if we flatten each cube and put on top of each other. The plane with edge CD is a neighboring plane of the intermediate cube where the red point is located.

layer corresponds to all points on the boundary of the cube, the middle layer corresponds to all points on the intermediate cube and so on.

Now what if we ask our readers to describe (or visualize) this arrangement of points on a two-dimensional space? If readers are familiar with some standard ways to visualize high-dimensional data points, the first thing that might come to their mind might be a method called Parallel Coordinate Plot (PCP) [17]. In this approach, we arrange the coordinate axes as vertical lines and represent each point (i.e., vector normalized to $[0, 1]$) as a line connecting the corresponding values on those vertical axes. However, this representation does not say anything about the geometry or structure of the data point cloud, i.e., it is not possible to discern if the points are arranged in a cube. Also we can not navigate over the data point cloud as we could naturally be able to do on a scatter plot. Also it does not convey any information about the location of the red point in the original (i.e. three-dimensional) space. In order to highlight, we can paint the corresponding line in PCP with red color. However, still it does not convey any spatial information about this data point w.r.t. to the entire set. An example PCP visualization is presented in Figure 2.2a. However, if the geometry of a point cloud is known beforehand, a PCP can capture some aspect of the structural properties of the high-dimensional manifold [18]. The major limitation of PCP is that it

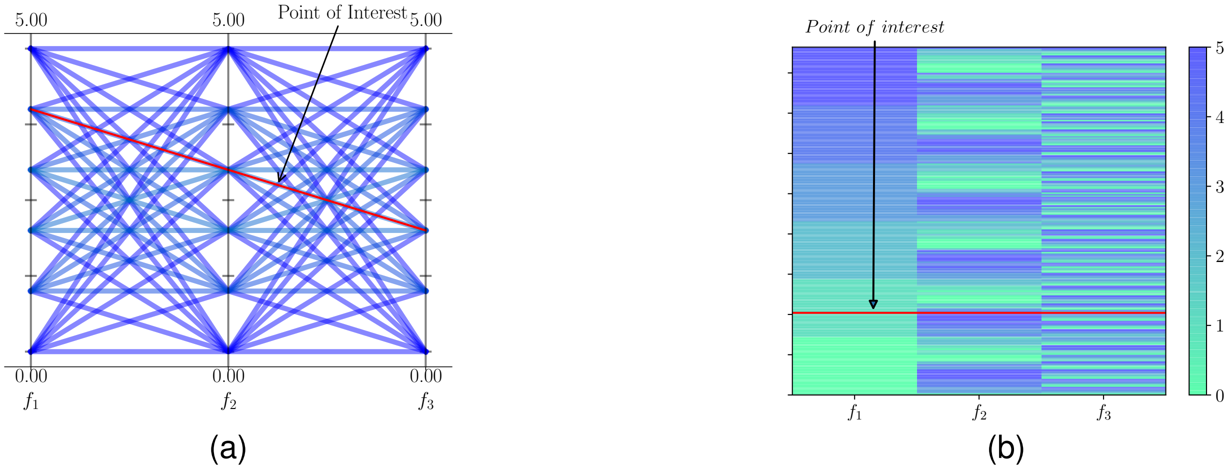


Figure 2.2: (a) PCP visualization of the cube in Figure 2.1a. (b) Heatmap visualization of the same cube.

generates different visualizations of the same data depending on the placement of its vertical axes. If the data points are in m-dimensional space, there will be $m!$ number of different PCP representations. Another approach would be to arrange the data points in a matrix and color each cell of the matrix according to their values (i.e. coordinate positions). This representation is known as Heat Map [19]. An example of a Heat Map visualization of the cube data set is presented in Figure 2.2b. Still, it is not possible to infer anything about the structure of the point cloud in its original dimension.

2.1.1 Radial Visualization (Radviz) Plot

A good way to capture the neighborhood relationships of the data points is to arrange them in a two-dimensional space in such a way that the relative distances among the data points are maintained. One such approach is Radial Visualization or Radviz [20]. In this method, the mapping procedure of points from m-dimensional space onto a two-dimensional plane is uniquely defined by first setting positions of m anchor points. The anchors are placed uniformly around a circle (but this is not necessary). An m-dimensional vector[2] $\mathbf{f} = (f_1, f_2, \ldots, f_m)$ is placed inside the circle at point $\mathbf{u} = (u_1, u_2)$ by assuming that each

[2] In this thesis, we denote each data point with $\mathbf{f}$ instead of a more standard notation $\mathbf{x}$. Since in our case, each data point $\mathbf{f} = (f_1, f_2, \ldots, f_m)$ is a vector of m multi-objective function values in $\mathbb{R}^m$.

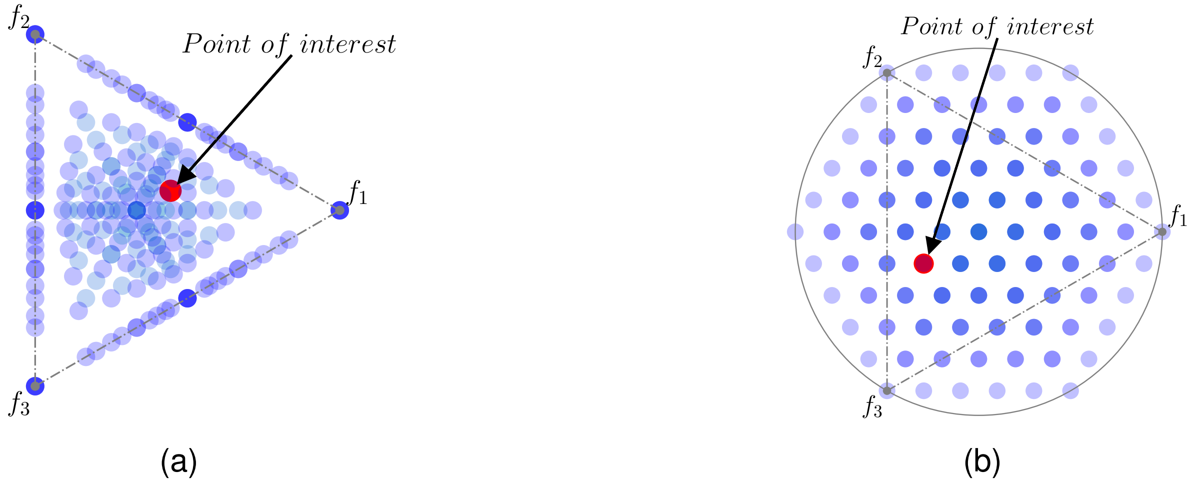


Figure 2.3: (a) Radviz visualization of the cube in Figure 2.1a. (b) Star-coordinate visualization of the same cube.

anchor, representing a data point f_i, connects point $\mathbf{u}$ with an imaginary spring having a stiffness f_i. The location of $\mathbf{u}$ is determined by the equilibrium of m spring force vectors. A little calculation will reveal the position of (u_1, u_2), as follows:

$$u_1 = \frac{\sum_{j=1}^{m} f_j \cos(\alpha_j)}{\sum_{j=1}^{m} f_j}, \quad u_2 = \frac{\sum_{j=1}^{m} f_j \sin(\alpha_j)}{\sum_{j=1}^{m} f_j}. \tag{2.1}$$

A resultant Radviz plot of the cube data set is presented in Figure 2.3a. We can see there are couple of problems with this visualization: i) the points are laid out as a triangle although the point cloud is structured in quadrilateral shape, ii) the mapping is not bijective, i.e. multiple points from m-dimensional space will be mapped onto the same point on the Radviz space. iii) If two points are within the same neighborhood radius in the high-dimensional space, then they will be in the same neighborhood in the Radviz space, however, the opposite is not true.

2.1.2 Star-coordinate Plot (SC-plot)

Another similar approach for mapping onto two-dimensional space is Star-coordinate Plot or SC-plot [21]. The idea of anchor replacement is similar to that of Radviz however the computation of $\mathbf{u}$ is done as follows:

$$u_1 = \sum_{j=1}^{m} f_j \cos(\alpha_j), \quad u_2 = \sum_{j=1}^{m} f_j \sin(\alpha_j). \tag{2.2}$$

The SC-plot mapping is basically done by arranging each coordinate axes in a radial fashion and computing the new point coordinates w.r.t those axes. A nice property of the SC-plot is that it can capture the shape of the original point cloud as one of its valid projections. An example SC-plot of the cube data set is presented in Figure 2.3b. The points are mapped as a hexagon and if we take a projection of all data points on a plane that is normal to the diagonal line AG of the cube ABCDEFGH in Figure 2.1a, the projection will be a hexagon. Also in terms of neighborhood relationship among the data points, SC-plot gives fewer non-bijective mappings of the data points than does Radviz. A detailed analysis of this aspect of Radviz and SC-plot can be found in [22].

If we apply a generic SC-plot in an MCDM scenario, the resulting plot doesn't conform with the interpretation of the mapping of the best (or the worst) values of the objective functions in a minimization problem. For example, the worst f_i value will be mapped onto the anchor point f_i. Therefore, we reverse the SC-formulation as follows:

$$u_1 = \sum_{j=1}^{m} (1 - f_j) \cos(\alpha_j), \quad u_2 = \sum_{j=1}^{m} (1 - f_j) \sin(\alpha_j). \tag{2.3}$$

Such that the worst f_i value is mapped on the extreme opposite of the anchor point f_i. Therefore, the data points closer to a certain anchor point f_j can be interpreted as having 'better' (i.e. lower) objective function values of f_j. This will make the reading of the generated PaletteViz plots easier.

2.1.3 Other Approaches

Yet another popular approach to high-dimensional data visualization is the so-called Scatter Plot Matrix. In this case we take two-dimensional scatter plots from each pair of axes and assemble them as a matrix. As a result, we end up with a collection of m^2 scatter

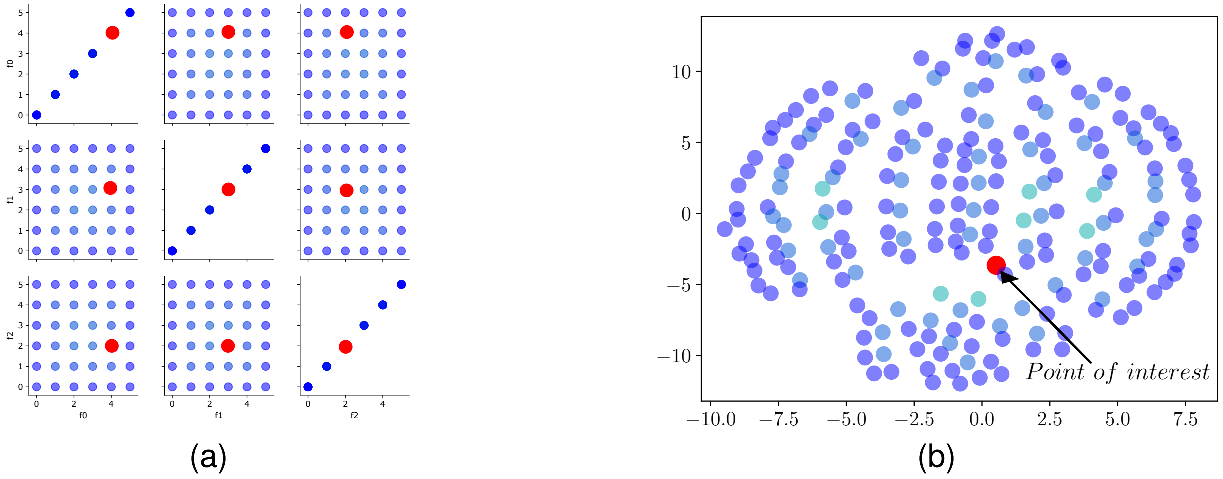


Figure 2.4: (a) Scatter plot matrix of the cube in Figure 2.1a. (b) t-SNE plot of the cube.

plots. Although this approach can capture the shape and structure of a data point cloud in a most correct manner, it is quite difficult to infer a clear mental image of the topology. Moreover, if the number of dimensions is increased, spatial navigation over the data points becomes problematic. An example of a scatter plot matrix of the cube data set is presented in Figure 2.4a.

There are another class of visualization methods that employs manifold learning and embedding of the latent variables in the data points and use that knowledge to map the points on lower dimensional space. For example, recently proposed t-Stochastic Neighborhood Embedding (t-SNE) [23] is one of such techniques. If we apply this method, the resultant visualization looks like the plot in Figure 2.4b. t-SNE can maintain neighborhood relationships in a local region, however it does not preserve the global structure of the data point cloud. In Figure 2.4b, the core points are distributed all over the plane. Also the relative position of the "point of interest" w.r.t to the entire data set does not conform to its position in the original high-dimensional space.

Over the years, various improvements and extensions of these methods have been proposed [24]. Unfortunately, none of them are directly useful in our case, particularly in the EMO and MCDM domain, since all of them miss some important criteria of visualization in terms of understanding the shape and structure of data points in high-dimensional space.

2.2 Visualization of a Pareto Set from a Decision Maker's Perspective

So far, the EMO literature has shown lukewarm interest in devising efficient methods for choosing a single preferred solution. This is probably due to the subjective, often non-analytic, considerations associated with the task. We describe a few such scenarios here:

- Points with large trade-off and *knee points*
- Boundary points
- Isolated points
- Active points lying on constraint boundaries or any other problem-specific desired points
- Robust and/or reliable points or any other problem-specific desired points

2.2.1 Points with Large Trade-off and Knee Points

Perhaps, the most desirable aspect of Pareto-optimal points is the points in the objective function space that offer the largest *trade-offs* with other points in the entire set [25, 1]. The trade-off of a point can be defined in many ways, but it involves the location of the point with respect to its neighbors in the objective space. A point having a large trade-off means the gain in moving to a neighbor is small compared to the loss in other objective values. Thus, there is not much motivation to make the move, as loss outweighs gain, and the point having a large trade-off value is most desired to the decision-makers. This is fairly reasonable in most decision-making tasks and it would be revealing to the DMs if such trade-off information were directly provided through a visualization technique. Towards this goal, we suggest the following. The trade-off of a Pareto point can be computed in two steps: (i) identify neighbors, and (ii) compute the trade-off. Since the nature of objectives and their relative values are well understood by DMs, a minimum threshold on the trade-off value can

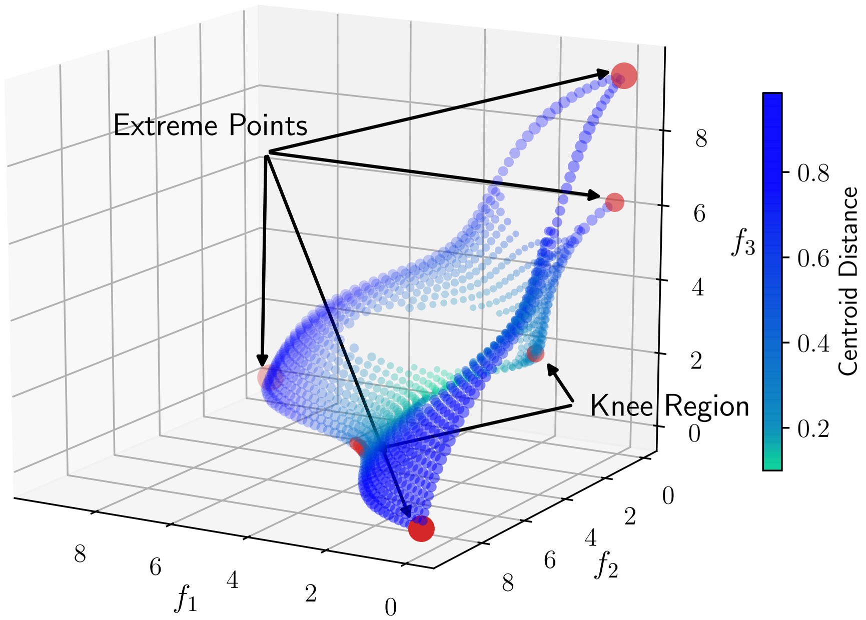


Figure 2.5: A three-dimensional Pareto-optimal front of DEB3D1K problem with a single knee [1] at its center. The knee points are presented with a red circle and the points with comparatively better trade-off are presented with circles of increasing radii. The color-coding represents the distance of each point from the centroid of the entire data-set. Here, light-green means the point is closer to the centroid and dark-blue color represents the points closer to the boundary. This data-set contains 1,005 points. This problem is defined in equation A.1 in appendix A.1.1.

be set by DMs and the Pareto-optimal points which exceed the threshold can be presented clearly in a visualization technique so that DMs can easily recognize these special data points from hundreds of other Pareto-optimal points for further consideration.

In this dissertation, we follow the definition of *knee points* (i.e. points with comparatively larger trade-off) discussed in [25]. Let us consider the simple Pareto set depicted in Figure 2.5, with three objectives which are to be minimized. The data-set has a clearly visible bulge in the middle. If we assume linear preference functions, and furthermore assume that each preference function is equally important, then the points at the knee (bulged) region are most likely to be a good choice to the DM. Note that in Figure 2.5, due to the concavity at the edges, similar reasoning holds for the three extreme points, which is why these extreme points [26] should also be considered as knee-like points.

2.2.2 Boundary Points

In most practical problems, the Pareto-optimal set is usually bounded in the objective space. In some occasions, there are disjoint Pareto-optimal sets (i.e., clusters of data points), which are also bounded.

Besides the nature of the points within the individual sets, DMs may be interested in knowing more about the boundary points of each set. This is because a boundary point [27] of a set (or manifold) means that there is no other Pareto-optimal point on one neighboring side, hence making it worth investigating further. Also, by definition, a boundary point is associated with an extreme value of at least one of the objectives in the point cluster. Thus, a DM would be interested in a further investigation of the respective design variable vector to understand the special properties of the point which prohibits any further increase or decrease of a set of specific objective functions.

2.2.3 Active Points

Boundary points may also lie on the intersection of several constraint surfaces in constrained many-objective optimization problems. Thus, in addition to boundary points, a DM may be interested in knowing the Pareto points that are *active* (lying on one or more constraint surfaces) and in contrast to points that are relatively far away from any constraint surface. A point lying on a constraint surface is usually less reliable if uncertainties in problem parameters and variables are expected. Thus, such active points must be considered from two conflicting purposes – being uniquely placed at the edge of the Pareto set and being unreliable due to uncertainties of the problem. In any case, such active points must be identified in a Pareto set, if they exist, and should be clearly represented through a visualization technique.

2.2.4 Isolated Points

In statistical data analysis, isolated or outlier points in a data-set are usually neglected and more focus is put on the remaining well-behaved data points. In many-objective Pareto set analysis, isolated points, in some sense, must be special and important to DMs, as they signify a unique and uncommon combination of objectives that place them in an atypical part of the objective space compared to the bulk of the other Pareto points. A DM should be interested in knowing these special outlier points, as they may portray a unique way of solving the problem, which may not have been known so far. Identifying such isolated points and showing them clearly through a visualization technique would be helpful to DMs.

There can be other practical considerations, such as locating robust and reliable points, points following certain trade-off paths, etc., which we do not consider here although we recognize that they can be included in our visualization framework in a future study.

2.2.5 Robust and Reliable Points

Many Pareto-optimal points may not be robust and reliable due to uncertainties in problem parameters and variables. If uncertainties are not considered during the optimization process, optimal solutions may not be robust or reliable against constraint violation. Thus, Pareto-optimal points which are robust or reliable can be clearly identified by suitable methods and represented clearly by the visualization technique to the DMs. To extend the concept of special points, any other problem-specific features which are difficult to quantify and consider during the optimization process can be used to identify desired points during the visualization process.

2.3 Chapter Summary

In this chapter, we have established the motivation behind the development and study conducted in the thesis. We have discussed how the existing visualization methods from the standard data analytics domain are not suitable for the Pareto-optimal data analysis and decision making. Even the existing visualization methods typically used in the MCDM domain do not convey enough insights into the Pareto-optimal front in a pragmatic manner, especially when the problem has many objectives. Furthermore, some important functional properties of the obtained Pareto-optimal solutions are either not considered or overlooked in the existing visualization methods. As a result, it is now apparent that an alternative visualization technique needs to be developed. In the next chapter we are going discuss a set of fundamental concepts in topological data analysis that will be very useful in the later developments of this thesis.

CHAPTER 3

UNDERSTANDING SHAPE IN HIGH DIMENSION

Data visualization for many-objective decision-making demands certain properties of the visualization techniques that are different from those used for other high-dimensional visualization tasks, including data analytics, multivariate statistics, and machine learning. Before going into the details, let us define the concept of data visualization for our purpose:

Definition 3.1. (Visualization) Given an m-dimensional data point cloud (or data-set, or point-set) $\mathcal{C} \subset \mathbb{R}^m$, where $|\mathcal{C}| = n$, a visualization is a transformation function $\mathcal{D} \leftarrow V(\mathcal{C})$ that maps $\mathcal{C}$ onto $\mathcal{D}$, where $\mathcal{D} \subset \mathbb{R}^2$.

The reason for a low-dimensional mapping of a higher-dimensional data point cloud is obvious: humans are not capable of discerning more than three dimensions. However, a desirable mapping procedure should make a visualization technique to accurately portray the high-dimensional nature of the data points. Therefore, we are interested in a practical and easy-to-understand visualization procedure for which the transformation function maintains certain key properties of the high-dimensional data-set in the lower-dimensional (i.e. two) mapped space.

The idea of *visualization* itself comes from the fact that we fundamentally want a *vis-a-vis* representation of the underlying space where the data points are situated. This calls for the need to understand the shape realized by all the data points in question – a reader might ask if there is any way to *see* a sphere (or a spherical shape) in a set of 5-dimensional data points when they are placed on the surface of a 5-dimensional hypersphere. Such question can be answered analytically through so called *Topological Data Analysis (TDA)* techniques. The proposed visualization method, i.e. PaletteViz, depends on some key concepts in TDA. In this chapter, we will conduct a short review of the formal background on TDA.

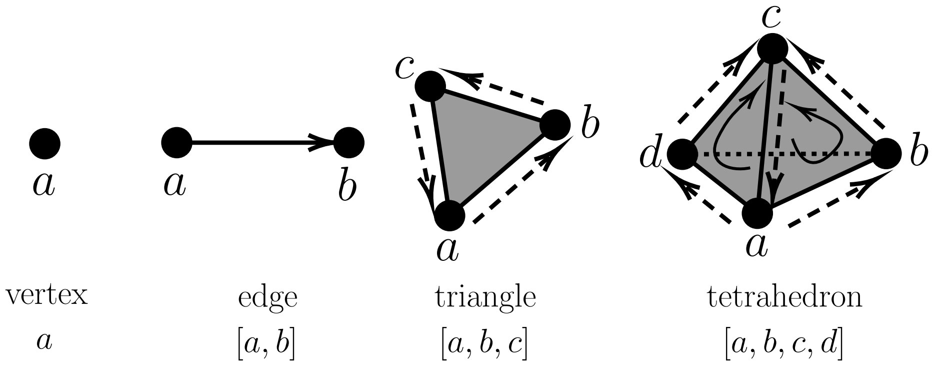


Figure 3.1: Oriented k-simplexes in $\mathbb{R}^3$, $0 \leq k \leq 3$. The orientations are shown along the edges. In the case of 3-simplexes, the orientations are understood as a chain of directions over each sub-simplex, e.g., in the case of tetrahedron, the orientation of the 2-simplex $\{a, b, c\}$ is counter-clock-wise.

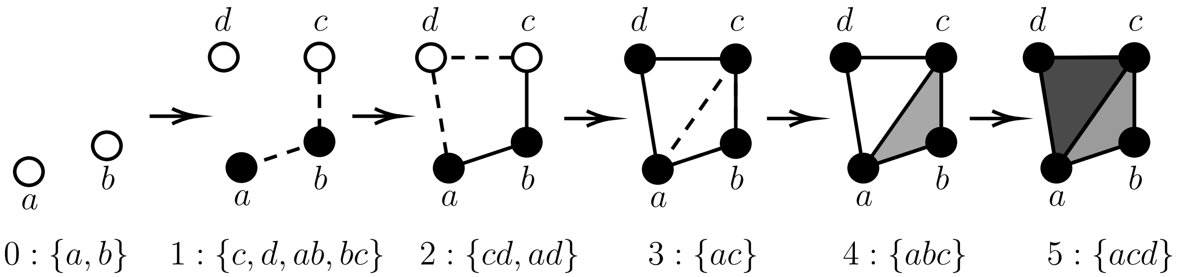
3.1 A Formal Background on Topological Data Analysis

In this section we describe some basic definitions that will be necessary to understand the algorithms and procedures discussed in this thesis. Due to the space constraints, we will exclude the relevant algebra which can be found in more detail in [28].

3.1.1 Simplicial Complexes

A *simplicial complex* is a set K, where $\Sigma_{\mathcal{C}}$ is a collection of subsets of K called *simplexes* (simplex, if singular) such that $\forall v \in K, \{v\} \in \Sigma_{\mathcal{C}}$, and if $\tau \subseteq \sigma \in \Sigma_{\mathcal{C}}$, then $\tau \in \Sigma_{\mathcal{C}}$. The sets $\{v\}$ are the vertices of K. When it is apparent from the context what $\Sigma_{\mathcal{C}}$ means, we call set K simply as a *complex*. We say $\sigma \in \Sigma_{\mathcal{C}}$ is a k-simplex of dimension $0 \leq k \leq m$ if $|\sigma| = k + 1$. If $\tau \subseteq \sigma$, τ is a *face* of σ, and σ is a *coface* of τ.

An *orientation* of a k-simplex σ, $\sigma = \{v_0, \ldots, v_k\}$, is an equivalence class of orderings of the vertices of σ, where $(v_0, \ldots, v_k) \sim (v_{\tau(0)}, \ldots, v_{\tau(k)})$ are equivalent if the sign of τ is 1. We denote an *oriented simplex* by $[\sigma]$. A simplex may be realized geometrically as the convex hull of $k + 1$ affinely independent points in $\mathbb{R}^m, m \geq k$. This realization gives us the familiar low-dimensional k-simplexes: vertices, edges, triangles, and tetrahedra, for $0 \leq k \leq 3$, shown in Figure 3.1. In a realized complex, the simplexes must meet at common



$0 : \{a, b\}$ $1 : \{c, d, ab, bc\}$ $2 : \{cd, ad\}$ $3 : \{ac\}$ $4 : \{abc\}$ $5 : \{acd\}$

Figure 3.2: A filtered complex with newly added simplexes highlighted.

faces. A *subcomplex* of K is a subset $L \subseteq K$ that is also a simplicial complex. A *filtration* of a complex K is a nested subsequence of complexes $\emptyset = K^0 \subseteq K^1 \subseteq \ldots \subseteq K^m = K$. For generality, we set $K^i = K^m$, $\forall i \geq m$. We call K a *filtered complex.* An example filtered complex is illustrated in Figure 3.2.

3.2 Vietoris-Rips Complex and α Complex

The geometric realization of simplicial complexes can be defined based on how the simplexes are included in K – which results in different notions of complexes. In this thesis we only focus on *Vietoris-Rips Complex* and α *Complex.* The former is needed to compute the boundary points of $\mathbf{H}_{k-1}$ homology groups (discussed in Chapter 3) and the latter is the basis for the PaletteViz visualization method (discussed in Chapter 4).

3.2.1 Vietoris-Rips Complex

Given a set $\mathcal{C}$ of points in Euclidean space $\mathbb{E}^k$, where $0 \leq k \leq m$, the *Vietoris-Rips* complex $\mathcal{R}_\epsilon(\mathcal{C})$ is the abstract simplicial complex whose k-simplexes are determined by subsets of $k + 1$ points in $\mathcal{C}$ with diameter at most ϵ.

3.2.2 α Complex and α Shape

The notion of α *complex* comes from the so-called *union of balls* in a *Delaunay triangulation* [29].

Definition 3.2. (Union of Balls) Let $\mathcal{C}$ be a finite set of points in $\mathbb{R}^k$, where $0 \leq k \leq m$ and ρ a non-negative real number. For each $p \in \mathcal{C}$, we let $\mathcal{B}_p(\rho) = p + \rho\mathbb{B}^k$ be the closed ball with center p and radius ρ. The union of these balls is the set of points at distance at most ρ from at least one of the points in $\mathcal{C}$,

$$U(r) = \|x - p\| \leq r, \text{ s.t. } x \in \mathbb{R}^k \mid \exists p \in \mathcal{C} \tag{3.1}$$

Definition 3.3. (Delaunay Triangulation) Given $\mathcal{C} \subset \mathbb{R}^k$ in *general position* [28], the Delaunay triangulation of $\mathcal{C}$ is the simplicial complex $\Delta(\mathcal{C}) \subseteq \Sigma_{\mathcal{C}}$ consisting only of

1. all d-simplexes $\sigma_d \in \Delta(\mathcal{C})$ such that the circumsphere of σ_d does not contain any other points of $\mathcal{C}$, and

2. all k-simplexes which are faces of other simplexes in $\Delta(\mathcal{C})$.

To decompose the union $U(r)$, we intersect each ball with the corresponding *Voronoi cell* [29] V_p, i.e. $\mathcal{R}_p(r) = \mathcal{B}_p(r) \cap V_p$. Since balls and Voronoi cells are convex, the $\mathcal{R}_p(r)$ are also convex. Any two of them are disjoint (i.e., overlap along a common piece of their boundaries), and together the $\mathcal{R}_p(r)$ cover the entire union.

Definition 3.4. (α Complex and α Shape) The α complex $\mathcal{K}_\alpha(r)$ is isomorphic to *the nerve* [28] of the attained cover at a certain r,

$$\mathcal{K}_\alpha(r) = \left\{ \sigma \subseteq \mathcal{C} \mid \bigcap_{p \in \sigma} \mathcal{R}_p(r) \neq \emptyset \right\} \tag{3.2}$$

Since $\mathcal{R}_p(r) \subseteq V_p$, the α complex is a subcomplex of the *Delaunay complex* (i.e. simplicial complexes arose from the triangulation). All the boundary points (i.e., $\delta_1\mathcal{K}_\alpha(r)$) of $\mathcal{K}_\alpha(r)$ gives the α-shape of $\mathcal{C}$.

It follows from the fact that for a set $\mathcal{C}$ in general position [28], the geometric realization is defined by its triangulation. In order to count the homology, we need to understand how each k-dimensional simplicial complex is formed by a set of lower-dimensional simplicial complexes. In the next section, we formalize this structure.

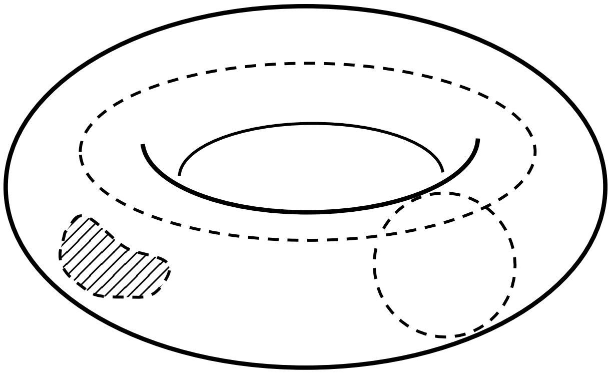


Figure 3.3: The shaded 1-boundary rests on the surface of a torus. The two solid 1-cycles (broken lines) form a basis for the first homology class of the torus. Cycles do not induce boundary, i.e. neither a piece of surface.

3.2.3 Chain Complex

The k^{th} *chain group* $\mathbf{C}_k$ of K is the free Abelian group on its set of oriented k-simplexes, where $[\sigma] = -[\tau]$ if $\sigma = \tau$ and σ and τ are oriented opposite. An element $c \in \mathbf{C}_k$ is a k-chain, $c = \sum_i n_i\,[\sigma_i]$, $\sigma_i \in K$ with coefficients $n_i \in \mathbb{Z}$. The *boundary operator* ∂_k : $\mathbf{C}_k \to \mathbf{C}_{k-1}$ is a homomorphism defined linearly on a chain c by its action on any simplex $\sigma = [v_0, v_1, \ldots, v_k] \in c$,

$$\partial_k \sigma = \sum_i (-1)^i\,[v_0, v_1, \ldots, \hat{v}_i, \ldots, v_k] \tag{3.3}$$

where $\hat{v}_i$ indicates that v_i is excluded from the sequence. The boundary operator connects the chain groups into a chain complex $\mathbf{C}_*$:

$$\cdots \to \mathbf{C}_{k+1} \xrightarrow{\partial_{k+1}} \mathbf{C}_k \xrightarrow{\partial_k} \mathbf{C}_{k-1} \to \cdots$$

We can also define subgroups of $\mathbf{C}_k$ using the boundary operator, which is defined with the cycle group $\mathbf{Z}_k = \ker \partial_k$ and the boundary group $\mathbf{B}_k = \operatorname{im} \partial_{k+1}$. We show examples of cycles in 3.3. An important property of the boundary operators is that **the boundary of a boundary is always empty**, $\partial_k \partial_{k+1} = \emptyset$. This fact, along with the definitions, implies that the defined subgroups are nested, $\mathbf{B}_k \subseteq \mathbf{Z}_k \subseteq \mathbf{C}_k$, as in Figure 3.4.

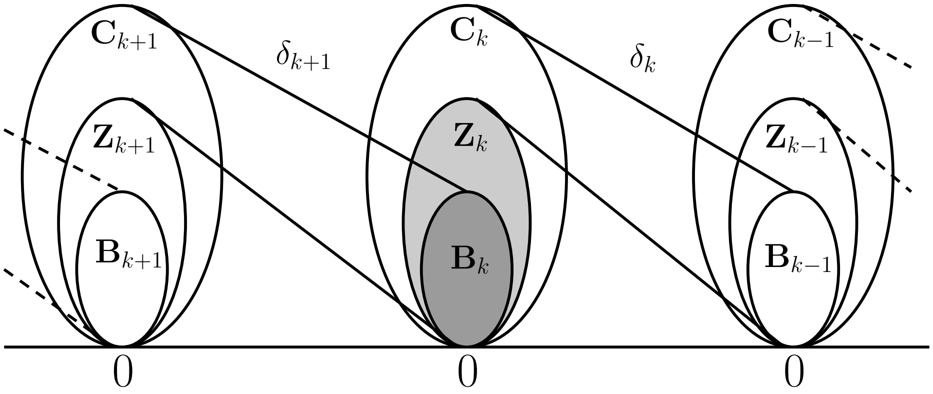


Figure 3.4: A chain complex with its components: chain, cycle, and boundary groups, and their images under the boundary operators. The null-space of M_k corresponds to $\mathbf{Z}_k$ and its range-space to $\mathbf{B}_{k-1}$.

3.3 Homology

The k^{th} *homology group* is defined as $\mathbf{H}_k = \mathbf{Z}_k/\mathbf{B}_k$. Its elements are classes of homologous cycles. To describe its structure, the Abelian groups are viewed as modules over the integers. This view allows alternate ground rings of coefficients (including fields). If the ring is a Principal Ideal Domain (PID) D, $\mathbf{H}_k$ is a D-module, then the rank of the module $\beta \in \mathbb{Z}$ is the Betti number of the module, and $\mathrm{d}_i \in D$ are its torsion coefficients. When the ground ring is $\mathbb{Z}$, the homology group describes the structure of finitely generated Abelian groups. The torsion sub-module disappears over a field, such as $\mathbb{R}$, $\mathbb{Q}$, or $\mathbb{Z}p$ for p a prime. The module is a vector space that is fully described by a single integer, its rank β, which depends on the chosen field. In the parlance of Pareto-optimal front, the homology corresponds to the void (or holes) appear if the objective function values $\mathbf{f}_i \in \mathbb{R}^k$ construct K.

3.3.1 Reduction

The standard method for computing homology is the reduction algorithm. We describe this method for integer coefficients as it is the more familiar ring. The method extends to modules over arbitrary PIDs.

As $\mathbf{C}_k$ is free, the oriented k-simplexes form the generic basis. We define the boundary operator $\partial_k : \mathbf{C}_k \to \mathbf{C}_{k-1}$ relative to the generic bases of the chain groups as an integer matrix M_k with entries in $\{-1, 0, 1\}$. The matrix M_k is called the standard matrix representation of ∂_k and we denote i^{th} row and j^{th} column with $M_k[i][j]$. It has n_k columns and n_{k-1} rows (the number of k- and $(k-1)$-simplexes, respectively). The null-space of M_k corresponds to $\mathbf{Z}_k$ and its range-space to $\mathbf{B}_{k-1}$, as explained in Figure 3.4. The reduction algorithm generates alternate bases for the chain groups, relative to which the matrix for ∂_k is diagonal. The algorithm utilizes the following elementary row operations on M_k:

1. exchange row i and row j.

2. multiply row i by -1.

3. replace row i by $\forall k : M_k[i][k] + qM_k[j][k]$, where $q \in \mathbb{Z}$ and $j \neq i$.

The algorithm also uses elementary column operations that are identical. Each column (row) operation corresponds to a change in the basis for $\mathbf{C}_k(\mathbf{C}_{k-1})$. For example, if e_i and e_j are the i^{th} and j^{th} basis elements for $\mathbf{C}_k$, respectively, a column operation of type (3) replaces e_i with $e_i + qe_j$. A similar row operation on basis elements $\hat{e}_i$ and $\hat{e}_j$ for $\mathbf{C}_{k-1}$, however, replaces $\hat{e}_j$ by $\hat{e}_j - q\hat{e}_i$. We shall make use of this fact in Section 3.3.2. The algorithm systematically modifies the bases of $\mathbf{C}_k$ and $\mathbf{C}_{k-1}$ using elementary operations to reduce M_k to its *(Smith) normal form*:

$$\tilde{M}_k = \left[\begin{array}{ccc|c} b_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & b_{l_k} & \\ \hline & 0 & & 0 \end{array}\right], \tag{3.4}$$

where $l_k = \operatorname{rank} M_k = \operatorname{rank} \tilde{M}_k$, $b^i \geq 1$, and $b_i | b_{i+1}$ for all $1 \leq i < l_k$. The algorithm can also compute corresponding bases $\{e_j\}$ and $\{\hat{e}_i\}$ for $\mathbf{C}_k$ and $\mathbf{C}_{k-1}$, respectively, although this is unnecessary if a decomposition is all that is needed. Computing the normal form in all dimensions, we get a full characterization of $\mathbf{H}_k$:

1. the torsion coefficients of $\mathbf{H}_{k-1}$ ($\mathrm{d}_i \in D$) are precisely the diagonal entries b_i greater than one.

2. $\{e_i \,|\, l_k + 1 \leq i \leq n_k\}$ is a basis for $\mathbf{Z}_k$. Therefore, $\operatorname{rank} \mathbf{Z}_k = n_k - l_k$.

3. $\{b_i \hat{e}_i \,|\, 1 \leq i \leq l_k\}$ is a basis for $\mathbf{B}_{k-1}$. Equivalently, $\operatorname{rank} \mathbf{B}_k = \operatorname{rank} M_{k+1} = l_{k+1}$.

Therefore, applying (2) and (3):

$$\beta_k = \operatorname{rank} \mathbf{Z}_k - \operatorname{rank} \mathbf{B}_k = n_k - l_k - l_{k+1} \tag{3.5}$$

3.3.2 Example

For the last complex in Figure 3.2, let's fix an orientation according to the alphabetical order on the vertices. Then the standard matrix representation of ∂_1 is

$$M_1 = \left[\begin{array}{c|ccccc} & ab & bc & cd & ad & ac \\ \hline a & -1 & 0 & 0 & -1 & -1 \\ b & 1 & -1 & 0 & 0 & 0 \\ c & 0 & 1 & -1 & 0 & 1 \\ d & 0 & 0 & 1 & 1 & 0 \end{array}\right], \tag{3.6}$$

where we show the bases within the matrix. Reducing the matrix, we get the normal form

$$\tilde{M}_1 = \left[\begin{array}{c|ccccc} & ab & bc & cd & z_1 & z_3 \\ \hline b-a & 1 & 0 & 0 & 0 & 0 \\ c-b & 0 & 1 & 0 & 0 & 0 \\ d-c & 0 & 0 & 1 & 0 & 0 \\ a & 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{3.7}$$

where $z_1 = ab + bc + cd - da$ and $z_2 = ab + bc - ca$ form a basis for $\mathbf{Z}_1$ and $\{b - a, c - b, d - c\}$ is a basis for $\mathbf{B}_0$.

We can use a similar procedure to compute homology over PIDs. A homogeneous basis is a basis of homogeneous elements. We begin by representing ∂_k relative to the standard basis of $\mathbf{C}_k$ and a basis for $\mathbf{Z}_{k-1}$. Reducing to normal form, we read off the description provided by direct sum using the new basis $\{\hat{e}_j\}$ for $\mathbf{Z}_{k-1}$:

1. zero row i contributes a free term with shift $\alpha_i = \deg \hat{e}_i$,
2. a row with diagonal term b_i contributes a torsional term with homogeneous $\mathrm{d}_j = b_j$ and shift $\gamma_j = \deg \hat{e}_j$.

The reduction algorithm requires $O(m^3)$ elementary operations, where m is the number of simplexes in K. The operations must be performed in exact integer arithmetic.

3.3.3 Persistence

The algorithms to find the boundary points of a $\mathbf{H}_{k-1}$ homology group are based on the computation of *persistence.* Given a filtered complex, the i^{th} complex K^i has associated boundary operators ∂_k^i, matrices M_k^i, and groups $\mathbf{C}_k^i$, $\mathbf{Z}_k^i$, $\mathbf{B}_k^i$ and $\mathbf{H}_k^i$ for all $i, k \geq 0$[1]. The *p-persistent k^{th} homology group of K^i* is:

$$\mathbf{H}_k^{i,p} = \mathbf{Z}_k^i / \left(\mathbf{B}_k^{i+p} \cap \mathbf{Z}_k^i \right) \tag{3.8}$$

Here both groups in the denominator are subgroups of $\mathbf{C}_k^{l+p}$, so their intersection is also a group, a subgroup of the numerator. The p-persistent k^{th}*Betti number* of K^i is $\beta_k^{i,p}$ the rank of the free subgroup of $\mathbf{H}_k^{i,p}$. We can also define *persistent homology groups* using the injection $\eta_k^{i,p} : \mathbf{H}_k^i \to \mathbf{H}_k^{i+p}$, that maps a homology class into the one that contains it, i.e. $\operatorname{im} \eta_k^{i,p} \simeq \mathbf{H}_k^{i,p}$ [30]. This definition can be extended over arbitrary PIDs as well.

Intuitively, the computation of persistence requires compatible bases for $\mathbf{H}_k^i$ and $\mathbf{H}_k^{i+p}$. It is not clear when a precise description is available for the compatible bases. In this section, we show how to combine the homology of all the complexes in the filtration into a single

[1] Note that superscripts indicate the filtration index and are not related to cohomology.

$$\begin{tikzcd}
{} \arrow[d, "\partial_3"] & {} \arrow[d, "\partial_3"] & {} \arrow[d, "\partial_3"] & \\
\mathbf{C}_2^0 \arrow[r, "f^0"] \arrow[d, "\partial_2"] & \mathbf{C}_2^1 \arrow[r, "f^1"] \arrow[d, "\partial_2"] & \mathbf{C}_2^2 \arrow[r, "f^2"] \arrow[d, "\partial_2"] & \ldots \\
\mathbf{C}_1^0 \arrow[r, "f^0"] \arrow[d, "\partial_1"] & \mathbf{C}_1^1 \arrow[r, "f^1"] \arrow[d, "\partial_1"] & \mathbf{C}_1^2 \arrow[r, "f^2"] \arrow[d, "\partial_1"] & \ldots \\
\mathbf{C}_0^0 \arrow[r, "f^0"] & \mathbf{C}_0^1 \arrow[r, "f^1"] & \mathbf{C}_0^2 \arrow[r, "f^2"] & \ldots
\end{tikzcd}$$

Figure 3.5: A portion of a persistence complex, with the chain complexes expanded.

algebraic structure. We then construct a correspondence that reveals a simple description over fields. Most importantly, we show that the persistent homology of a filtered complex is the standard homology of a particular graded module over a polynomial ring. In order to explain this, we need couple of more definitions.

Definition 3.5. (Persistence Complex) A *persistence complex* is a family of chain complexes $\{\mathbf{C}_*^i\}_{i \geq 0}$ over a commutative ring R with unity, equipped with chain maps $f^i : \mathbf{C}_*^i \to \mathbf{C}_*^{i+1}$, so that we have the following diagram:

$$\begin{tikzcd}
\mathbf{C}_*^0 \arrow[r, "f^0"] & \mathbf{C}_*^1 \arrow[r, "f^1"] & \mathbf{C}_*^2 \arrow[r, "f^2"] & \cdots
\end{tikzcd}$$

The *filtered complex* K with inclusion maps for the simplexes becomes a persistence complex. We show an example of a persistence complex, with the chain complexes expanded in Figure 3.5. The filtration index increases horizontally to the right under the chain maps f_i, and the dimension decreases vertically to the bottom under the boundary operators ∂_k.

3.4 Bound on k in $\mathbf{H}_k$ of a Pareto-optimal Front

In a Pareto-optimal front, not all homology groups exist. The highest dimensional homology groups of a Pareto-optimal front in m-dimensional objective space is $\mathbf{H}_{m-1}$.

Lemma 3.1. (Bound on k in $\mathbf{H}_k$ of a Pareto-optimal Front) Let $\mathcal{PF}$ denote the set of Pareto optimal values, and let $\mathcal{F}$ denote the set of achievable objective values. First we show that $\mathcal{PF} \subseteq \mathcal{F} \cap \mathrm{bd}(\mathcal{F})$, i.e., every Pareto optimal value is an achievable objective value

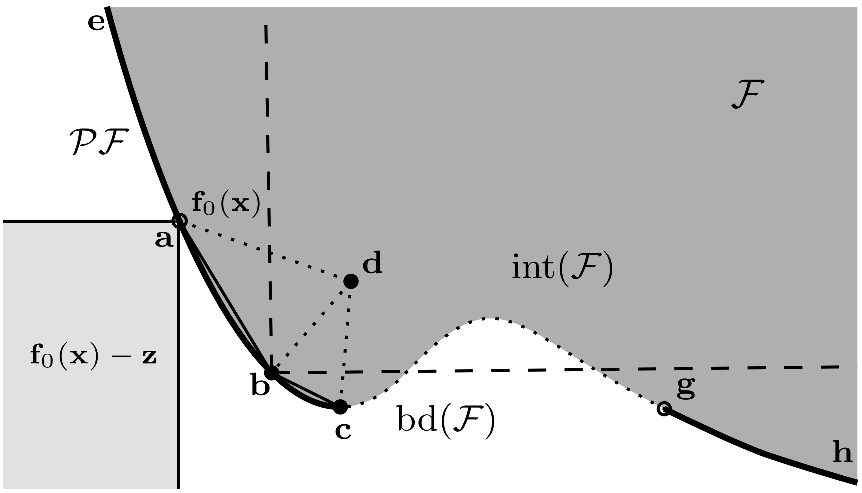

e
$\mathcal{F}$
$\mathcal{PF}$
$\mathbf{f}_0(\mathbf{x})$
a
d
$\mathrm{int}(\mathcal{F})$
$\mathbf{f}_0(\mathbf{x}) - \mathbf{z}$
b
g
c
$\mathrm{bd}(\mathcal{F})$
h

Figure 3.6: $\mathcal{F}$ is a set of achievable objective values in two dimensional space. The dark shaded region of $\mathcal{F}$ is the *interior set* $\mathrm{int}(\mathcal{F})$. $\mathrm{bd}(\mathcal{F})$ is the *boundary set* of $\mathcal{F}$. The bold curves ($\mathbf{e} - \mathbf{c}$ and $\mathbf{g} - \mathbf{h}$) on $\mathrm{bd}(\mathcal{F})$ are in $\mathcal{PF}$. The curve $\mathbf{c} - \mathbf{g}$ is in $\mathrm{bd}(\mathcal{F})$, but not in $\mathcal{PF}$. All the points in region contained in a dashed rectangle at $\mathbf{b}$ is dominated by it. Two adjacent 2-simplexes $\{\mathbf{a}, \mathbf{b}, \mathbf{d}\}$ and $\{\mathbf{b}, \mathbf{d}, \mathbf{c}\}$ are in $\mathrm{int}(\mathcal{F})$, where $\mathbf{b} \prec \mathbf{d}$.

that lies in the boundary of the set of achievable objective values. $\mathcal{PF} \subset \mathcal{F}$, because that is part of the definition of Pareto optimal points. Suppose $\mathbf{f}_0(\mathbf{x}) \in \mathcal{PF}$ and $\mathbf{f}_0(\mathbf{x}) \in \mathrm{int}(\mathcal{F})$. Then $\mathbf{f}_0(\mathbf{x}) - \mathbf{z} \in \mathcal{F}$ for all sufficiently small $\mathbf{z}$. In such case, $\mathbf{f}_0(\mathbf{x})$ is not a Pareto-optimal point. Therefore, $\mathcal{PF} \subseteq \mathcal{F} \cap \mathrm{bd}(\mathcal{F})$ (See Figure 3.6).

Now assume sufficiently enough points $\mathcal{C}$ are sampled in $\mathcal{F}$, i.e. $\mathcal{C} \subseteq \mathcal{F}$. Then geometric simplicial complexes K are formed over $\mathcal{C}$. If we apply boundary operator ∂_k on the k^{th} chain group $\mathbf{C}_k$ of K, we arrive at subgroup $\mathbf{C}_{k-1}$ of K on its set of oriented $(k-1)$-simplexes on the point set in $\mathrm{bd}(\mathcal{F})$. Consequently, both the *boundary group* $\mathbf{B}_{k-1} = \mathrm{im}\, \partial_k$ and the *cycle group* $\mathbf{Z}_{k-1} = \ker\, \partial_{k-1}$ excludes the subspace spanned by all the k-simplexes in $\mathrm{int}(\mathcal{F})$. Therefore, $k - 1^{\mathrm{th}}$ homology group $\mathbf{H}_{k-1} = \mathbf{Z}_{k-1}/\mathbf{B}_{k-1}$ is defined only by $\mathrm{bd}(\mathcal{F})$. As we have already proved $\mathcal{PF} \subseteq \mathcal{F} \cap \mathrm{bd}(\mathcal{F})$, the Pareto dominance relation excludes all the points in $\mathrm{int}(\mathcal{F})$. Therefore, The highest dimensional homology groups of a Pareto-optimal front in m-dimensional objective space is $\mathbf{H}_{m-1}$. As a result, if the $\mathcal{PF}$ is disconnected, the boundary of the discontinuity has homology group of $\mathbf{H}_{m-2}$. $\square$

This fact is illustrated in Figure 3.6 in a two-dimensional objective space. Two adjacent 2-simplexes $\{\mathbf{a}, \mathbf{b}, \mathbf{d}\}$ and $\{\mathbf{b}, \mathbf{d}, \mathbf{c}\}$ are in $\mathcal{F}$. The Pareto-optimal front excludes $\mathbf{d}$ since $\mathbf{b} \prec \mathbf{d}$. Therefore, two 1-simplicial chains $\mathbf{a} - \mathbf{b}$ and $\mathbf{b} - \mathbf{c}$ are in $\mathrm{bd}(\mathcal{F})$, but $\mathbf{d}$ is not. Therefore, a two-dimensional $\mathcal{PF}$ is bounded by a set of at most 1-simplexes (but not by a 2-simplex). Consequently, the end-points of the disconnected $\mathcal{PF}$ is bounded by a point (i.e. 0-simplex), e.g. the chain group $\mathbf{e} - \mathbf{c}$ is bounded by points $\mathbf{e}$ and $\mathbf{c}$.

3.5 Level Sets and Persistence Diagrams

So far we have developed a fairly comprehensive idea on the computation of simplicial homology of a point cloud $\mathcal{C} \subset \mathbb{R}^k$. However, what PaletteViz basically does is that it finds the *alpha* shape's boundary points of $\mathcal{C}$ and consider each set of boundary points as so called *level set* with respect to the centroid of $\mathcal{C}$. In order to find those points, we need to understand how the simplicial homology is represented in terms of *level sets* and *persistence diagrams.*

Definition 3.6. (Level Sets) Given a piecewise-linear function $f : |K| \to \mathbb{R}$ and a value $\lambda \in \mathbb{R}$, we define the upper[2] λ-level set of f as

$$L_\lambda = \{x : f(x) \geq \lambda\} = f^{-1}([\lambda, \infty)) \tag{3.9}$$

For any two level thresholds $\lambda_1 > \lambda_2$, the corresponding level sets satisfy $L_{\lambda_1} \subset L_{\lambda_2}$. Thus, the collection of level sets $\mathcal{L} = \bigcup_\lambda \{L_\lambda\}$ forms a filtration with the level as the index set.

3.5.1 Persistence Diagrams

For each level set L_λ, the corresponding topological features are captured by the generators of its homology groups. In a simpler term, the 0^{th} order homology groups (0^{th} order topological feature) capture the connected components (i.e. chain), the 1^{st} order homology groups

[2] Some literatures consider the lower level set as $f^{-1}((-\infty, \lambda))$.

capture regions forming a cycle, and 2^{nd} order homology groups capture regions forming a void structure (i.e. boundary of a 2-simplex), etc.

If we decrease the level λ, new generators for the homology groups may be created (e.g., the formation of new components), existing generators may merge together (e.g., two connected components joining together), and existing generators may be eliminated (e.g., a loop getting filled in). The level at which a generator is created is called its *birth time* and the level at which a generator is destroyed (or merges with another generator that has an earlier birth time) is called its *death time*. Therefore, every generator in $\mathcal{L}$ encodes three features – homology order, birth time, and death time. The persistence diagram is a collection of all these tuples found from of filtration formed by a given f. Thus, if a function's filtration has $|T|$ generators, the corresponding persistence diagram is

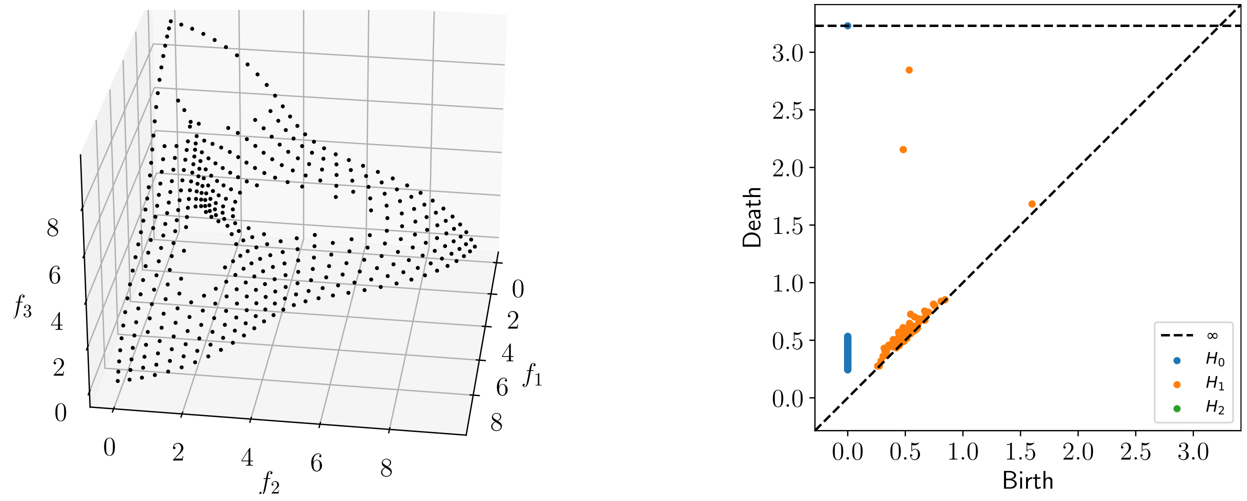
$$\mathsf{Dgm} = \left\{ \left(r_j, t_{b_j}, t_{d_j} \right) : j = 1, \cdots, |T| \right\} \tag{3.10}$$

where r_j, t_{b_j} and t_{d_j} are the homology order, birth time, and death time of the j^{th} generator, respectively. The norm $|T|$ specifies the number of off-diagonal elements in the persistence diagram Dgm. The norm of $|T|$ will be used to find the boundary points in a $\mathbf{H}_{m-1}$ homology groups in Section 6.5. When we compute Dgm of a data point cloud $\mathcal{C}$, we denote the diagram as $\mathsf{Dgm}(\mathcal{C})$.

An example $\mathsf{Dgm}(DEB3D1K)$ is presented in figure 3.7b. Here we can see $\mathsf{Dgm}(DEB3D1K)$ does not have any $\mathbf{H}_2$ homology group. The two off-diagonal values of $\mathbf{H}_1$ indicates that the Pareto-optimal front has two larger bounding chains in the interior – which effectively shows that there are two large *voids* in the Pareto-optimal front. Different $\mathbf{H}_1$ homology groups of DEB3D1K Pareto-optimal front at different scaling (death time) is presented in figure 3.8.

3.6 Level Sets Based on Depth: Point Depth Ranking

The notion of *data depth* or *point depth* is frequently encountered in non-parametric multivariate data analysis [31]. This concept provides *center-outward orderings* of points in the Euclidean space of any dimension and leads to a novel non-parametric multivariate



(a)

(b)

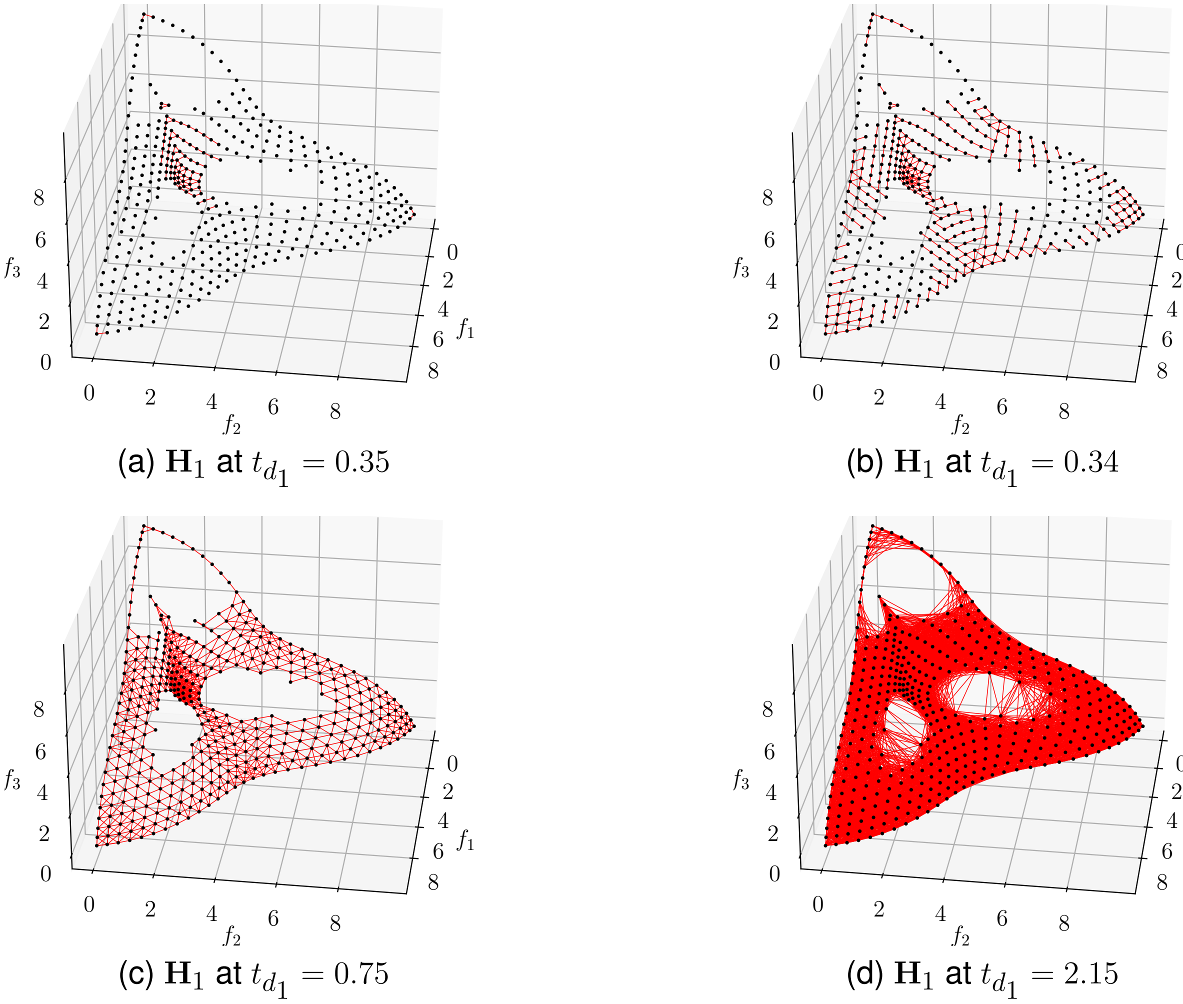
Figure 3.7: (a) A Pareto-optimal front found by solving DEB3D1K problem. This data-set consists 100 data points. (b) Persistence diagram of (a). Here we can see there is no homology group $\mathbf{H}_3$ due to the lemma 4.1.

statistical analysis method where no assumption regarding the point cloud distribution is needed. A point depth measures how deep (or central) a given point[3] $\mathbf{f} \in \mathcal{C}$ is relative to the entire data-set $\mathcal{C}$. In our proposed visualization method, we use the concept of *non-convex hull peeling depth* [32] of a point $\mathbf{f}$ with respect to the entire data-set $\mathcal{C}$, is the level of the non-convex layer that $\mathbf{f}$ belongs to. A non-convex layer is defined as follows:

Construct the smallest non-convex hull which encloses all points in $\mathcal{C}$. The points on the perimeter are designated as the first non-convex layer and are removed. The non-convex hull of the remaining points is constructed and the points on the next perimeter are marked as the second non-convex layer. The process is repeated, and a sequence of nested non-convex layers is formed. The higher the layer a point belongs to, the deeper the point is within the data-set. The non-convex layer of a data-set can be obtained through the construction of α-shapes [33]. Conceptually, α-shapes are a generalization of the convex hull of a point set.

Given $\mathcal{C} \subset \mathbb{R}^m$ and α, a real number with $0 \leq \alpha \leq \infty$, the α-shape of $\mathcal{C}$ is a *polytope* that is neither necessarily convex nor necessarily connected. For $\alpha = \infty$, the α-shape is identical

[3] In this thesis, we denote each data point with $\mathbf{f}$ instead of a more standard notation $\mathbf{x}$, since in our case, each data point $\mathbf{f} = (f_1, f_2, \ldots, f_m)$ is a vector of m multi-objective function values in $\mathbb{R}^m$.



(a) $\mathbf{H}_1$ at $t_{d_1} = 0.35$

(b) $\mathbf{H}_1$ at $t_{d_1} = 0.34$

(c) $\mathbf{H}_1$ at $t_{d_1} = 0.75$

(d) $\mathbf{H}_1$ at $t_{d_1} = 2.15$

Figure 3.8: Different $\mathbf{H}_1$ homology groups of DEBM3D1K induced at different t_{d_1} values.

to the convex hull of $\mathcal{C}$. It is a polytope in a fairly general sense: it can be concave and even disconnected; it can contain any simplexes from $(m-1)$-simplex to 1-simplex; and its components can be as small as 0-simplex (single isolated points). The parameter α controls the maximum *curvature* of any cavity of the polytope. The palette visualization basically finds the boundary of an α-shape of $\mathcal{C}$ and separates the boundaries layer by layer.

Henceforth, $D_{\mathcal{C}}(\mathbf{f})$ is used to indicate a non-convex layer depth of point $\mathbf{f}$ in $\mathcal{C}$, unless otherwise specified. A larger $D_{\mathcal{C}}(\cdot)$ always implies a deeper (or more central) $\mathbf{f}$ in $\mathcal{C}$. When no possibility of confusion or we are only interested in one point cloud $\mathcal{C}$, we may omit $\mathcal{C}$ and use $D(\cdot)$ to denote $D_{\mathcal{C}}(\cdot)$. All the points that belong to the same depth form a *depth-contour* and belong to the same *depth-equivalence class*. Such equivalence classes are the basic cue to understand the shape of a high-dimensional data-set [31] [34]. Our proposed visualization technique is based on the depth-contour measure of a data-set. For example, in Figure 2.1b, all the points on the cube on the outer layer have depth contours of 0, the points on the second layer have depth contours of 1, the points in the inner-most layer have depth contours of 2 and so on.

In other way, the point depth ranking can also be defined as the enumeration of level-sets constructed with respect to the distances from the centroid (or center of mass) of the data point cloud $\mathcal{C}$:

$$\mu = \frac{1}{n}\left[\sum_{i=1}^{n} f_1(\mathbf{x}_i), \sum_{i=1}^{n} f_2(\mathbf{x}_i), \ldots, \sum_{i=1}^{n} f_m(\mathbf{x}_i)\right]^T, \quad \forall \mathbf{x}_i \in \mathcal{C} \tag{3.11}$$

Alternatively, any other centrality measure can also be used, e.g. Chebyshev center $\hat{\mathbf{x}}$ for a given radius r:

$$\min_{\hat{\mathbf{x}}, r} \left\{ r : \|\mathbf{x}_i - \hat{\mathbf{x}}\|_2 < r, \, \forall \mathbf{x}_i \in \mathcal{C} \right\} \tag{3.12}$$

All the points that belong to the same level set (constructed from the centroid of $\mathcal{C}$) have the same depth ranks. Depth ranks are the key component in understanding the shape realized by Pareto-optimal data through PaletteViz. We discuss the α-shape construction process in the next chapter in more detail.

3.7 Neighborhood Relationship

Given the notion of depth and a way to enumerate the depth-contours of all the points in the high-dimensional data-set $\mathcal{C} \subset \mathbb{R}^m$, we need to find a way to represent them in a two-dimensional space. However, a bijective mapping is not possible universally. In order to map points onto two dimensions, we need to discard (or collapse) $m - 2$ dimensions in such a way that the values from the remaining column vectors do not map multiple data points from $\mathbb{R}^m$ onto $\mathbb{R}^2$ at the same coordinate position (to avoid the surjective mapping).

The only way to circumvent this situation is to slightly change the coordinate positions in $\mathbb{R}^2$ so that the points within the same vicinity in $\mathbb{R}^m$ are not moved far away from each other in $\mathbb{R}^2$. This condition basically tells us that the points in higher dimension should maintain the same neighborhood relationship when they are mapped onto the two-dimensional space. More formally, we can define this condition with the notion of a *hyper-sphere neighborhood.* Assuming a function $\mathcal{D} \subset \mathbb{R}^2 \leftarrow V(\mathcal{C} \subset \mathbb{R}^m)$, if two points $\{\mathbf{p}, \mathbf{q}\} \in \mathcal{C}$ are within a hyper-sphere of radius $r_{\mathcal{C}}$, then $\{\mathbf{r}, \mathbf{s}\} \in \mathcal{D}$ should be within a circle of radius $r_{\mathcal{C}}$, where $\{\mathbf{r}, \mathbf{s}\} \leftarrow V(\{\mathbf{p}, \mathbf{q}\})$, assuming all points in both $\mathcal{C}$ and $\mathcal{D}$ are normalized within $[0, 1]$. This condition of the neighborhood is another basic requirement for an *ideal visualization* technique. The idea is to find a way to achieve a mapping that ensures neighborhood relationships as well as the correct arrangements of depth contours.

There are some techniques known as *neighborhood embedding methods* [23] that try to ensure the neighborhood condition during a visualization. However, such methods require a stochastic optimization algorithm, but they are not capable of maintaining the point depth relations simultaneously. If we apply such methods, the boundary points may be mapped onto the lower-dimensional space at the cost of minimizing the error in the neighborhood inclusion of data points [35] in the higher-dimension.

For example, if we use t-SNE (t-Stochastic Neighborhood Embedding) [23] to visualize a three-dimensional Pareto set for a vehicle crash-worthiness problem [2], we observe that two different runs generate two different visualizations and the *center-outward* relationship among the points is not maintained. As a result, if we navigate along a set of points in the mapped two-dimensional space, it is not possible to tell if we are traversing the same region in the higher-dimensional space. This situation is presented in Figure 3.9. There are two separate clusters in the data-set. The light green points are the Pareto-optimal points that are close to the cluster centroids and dark blue points are close to the cluster boundaries. Figure 3.9b: t-SNE mapping of the same data-set. Here, we see that the neighborhood relationships are completely distorted in cluster B. The t-SNE mapping divides cluster B into two different clusters, resulting in a total of three clusters in the two-dimensional mapped space. Points that are close to the cluster B centroid are separated by some boundary points.

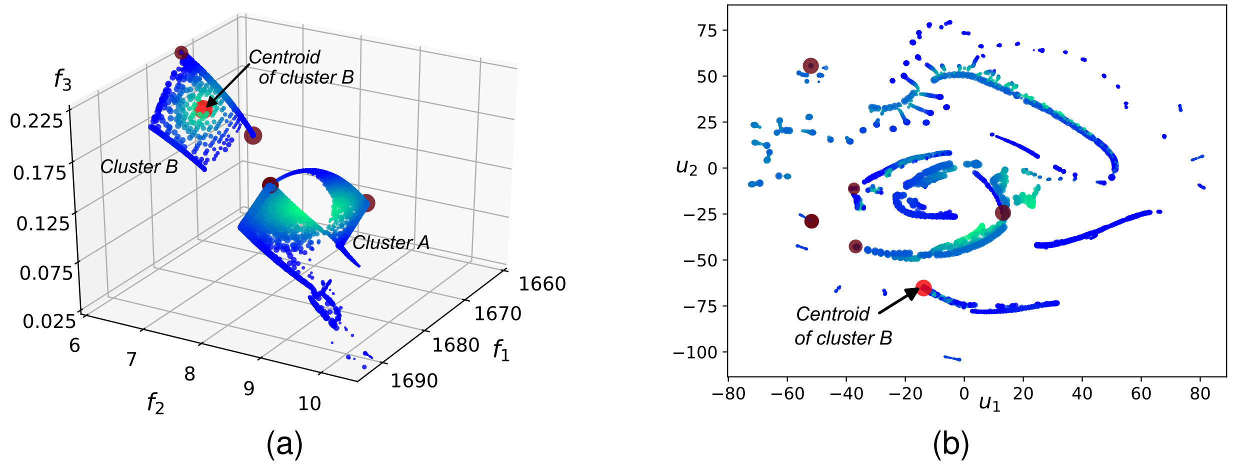


Figure 3.9: (a) Three-dimensional Pareto-optimal front for the vehicle crash-worthiness problem [2]. (b) Same data points rendered using t-SNE.

Also, one boundary point of cluster B with high trade-off (red color) is placed close to the farthest corner of cluster A. There are 4,450 data points in this data-set.

However, there are other deterministic methods that are able to map high-dimensional data onto a two-dimensional space while maintaining neighborhood relationships with minimal surjective mapping. The Radial visualization (Radviz) technique [20] is one such method, which we have already described in the previous chapter.

3.8 Spatial Navigability

The above properties of the Pareto set, if they exist in a problem either individually or together, must be clearly identifiable by the DMs through an efficient visualization technique. This calls for a reduced dimension for a visualization, as the human mind is trained to navigate in a spatial dimension involving two (or at most, three) dimensions. Thus, an ideal visualization technique should be able to provide the capability of *spatial navigability* after the transformation $\mathcal{D} \subset \mathbb{R}^2 \leftarrow V(\mathcal{C} \subset \mathbb{R}^M)$ is completed. By being *spatially navigable*, we mean an arrangement of points in the lower-dimensional space is done in such a way that if a DM navigates through a specific path in $\mathcal{C}$, the path should closely imitate a path in $\mathcal{D}$. More formally,

Definition 3.7. (Spatial Navigability) Let us assume two points $\{\mathbf{p}, \mathbf{q}\} \in \mathcal{C}$ which are connected by a path $\mathfrak{p}$ and the set of points that constitute $\mathfrak{p}$ are $\mathcal{P} \subset \mathcal{C}$. If the mapping function $V(\cdot)$ maintains spatial navigability, then there is a path $\mathfrak{q}$ with a set of points $\mathcal{Q} \subset \mathcal{D}$ that connects two points $\{\mathbf{r}, \mathbf{s}\} \in \mathcal{D}$, where $\{\mathbf{r}, \mathbf{s}\} \leftarrow V(\{\mathbf{p}, \mathbf{q}\})$ and $\mathcal{Q} \leftarrow V(\mathcal{P})$.

The above definition is basically a corollary to the neighborhood relationship criterion that has been discussed in the previous section. Henceforth, if a visualization method can faithfully capture the shape and structure of the point cloud in a high-dimension while mapping it over the lower-dimensional space, we can say that such visualization method is able to provide the capability of *spatial navigability.* In our proposed method, we seek to provide this property.

Therefore, both the concept of point depth and neighborhood relationships need to be maintained so that a DM can traverse through the low-dimensional data points without confusing the relative neighborhood relationships of the original (high-dimensional) data points. In this way, a DM can also understand which points lie on the boundary of the original high-dimensional space or have any special properties, and navigate and visualize through the low-dimensional mapped space.

3.9 Chapter Summary

In this chapter, we have developed a formal background on the different key aspects of topological data analysis. Also, we have discussed how they are relevant to visualization of a high-dimensional Pareto-optimal front. We have started with rudimentary ideas on simplicial complexes and then gradually progressed into more sophisticated concepts. We have also discussed how all these ideas are important when a visualization method must consider a number of practical aspects, e.g. neighborhood relationship and spatial navigability. In the next chapter, we will finally discuss the PaletteViz method in more detail.

CHAPTER 4

THE PALETTEVIZ METHOD

The proposed method, we call a *Palette Visualization (PaletteViz)* technique, mimics an artist's palette which is a two-dimensional plane and has regions formed with similar colors. Although there may not be any higher-dimensional simile to the palette in an artist's mind, our visualization technique is expected to map neighboring high-dimensional points close to a region in the two-dimensional palette. The two-dimensional palette allows a DM to easily navigate from one region to another, and similarity and dissimilarity in colors help DMs to have a visual picture of the proximity of points to each other. Of course, when high-dimensional data is compressed into a low-dimensional space, there will always be a loss of information. However, our technique, as will be revealed soon, focuses on maintaining as much higher-dimensional information as possible for the preferred Pareto points and does not care much if the information is lost on non-preferred points.

4.1 Proposed Palette Visualization (PaletteViz) Method

First, we need a technique to map a high-dimensional data-set into a two-dimensional space. Various methods exist in the literature for this purpose and can be used with our method, but here we use the Star-coordinate plot [20] due to its simplicity and popular use in EMO studies. Second, we need a method to maintain the structure and neighborhood information of the high-dimensional data-set into the two-dimensional space. For this purpose, we separate the points according to their centrality. Points which are closer to the boundary of the Pareto-optimal set are isolated and plotted separately. The next set of points towards the core of the Pareto set are plotted next, and so on, until the core points are plotted at the end. Such a structure-preserving plot will allow DMs to focus on boundary or core points according to their preference. Third, the points are marked in different sizes and colors by providing special emphasis to the DM-preferred functionalities so that they become visually and clearly apparent to DMs. In summary, the visualization procedure is performed in two phases:

1. Construction of depth contours and assigning the depth-equivalence class to each of the data points in $\mathcal{C} \subset \mathbb{R}^m$.
2. Representation of each depth contour in a two-dimensional palette mapping so that the neighborhood relationships and center-outward orderings are maintained as accurately as possible.

4.1.1 Computing The Depth Contours Using α-Hull

In order to find the depth contours of the data points in $\mathcal{C} \subset \mathbb{R}^m$, we have utilized the notion of *non-convex hull peeling depth* using an algorithm called 'α-shape' [33, 32]. Since the α-shape algorithm can be slow to run for large-dimensional spaces, we have modified the algorithm to make it faster at the cost of reduced accuracy. This is due to the fact that we do not need the most accurate enumeration of depth contours and some errors are tolerable[1], as we combine multiple boundary layers together to represent a few combined layers at the end. The depth contour enumeration with an approximate α-shape algorithm is presented in Algorithm 1.

The α-shape algorithm starts with a Delaunay triangulation of the high-dimensional data points $\mathcal{C}$. Then the algorithm decides which edges need to be removed to get the final non-convex hull. In the case of exact α-shape, the filtration (i.e. removal of edges in a Delaunay triangulation) is done according to all the simplexes ranging from 0 to $m - 1$ dimensions. However, we avoid such loop; instead, we assume that filtration by the removal of the longest edge in the convex-hull should give an approximate non-convex hull shape. Therefore, we set α to the length of the shortest convex-hull edge.

[1] However, an exact α-shape algorithm can also be used without any loss in visualization for a lower-dimensional space.

Algorithm 1 Enumerate all depth contours using the approximate α-shape

```
Require: A set of $n$ data points $\mathcal{C} \subset \mathbb{R}^m$
for each point $\mathbf{f}_i \in \mathcal{C}$ do
    $D(\mathbf{f}_i) \leftarrow 0$
end for
$d \leftarrow 1$
repeat
    Delaunay triangulate $\mathcal{C}$: $\{T, H\} \leftarrow \mathcal{C}$
    Set of all $m$-simplexes $T = \{\Delta_1, \Delta_2, \ldots, \Delta_k\}$
    Set of convex hull edges $H = \{e_1, e_2, \ldots, e_l\}$
    for each simplex $\Delta_i \in T$ do
        for each edge $e \in \Delta_i$ do
            flag($e$) $\leftarrow 0$
        end for
    end for
    $\alpha \leftarrow \min\{e_1, e_2, \ldots, e_l\}$
    for each simplex $\Delta_j \in T$ do
        $r \leftarrow$ circumradius of $\Delta_j$ (equation 4.2)
        if $r > \alpha$ then
            for each edge $e \in \Delta_j$ do
                if flag($e$) $\geq 0$ then
                    flag($e$) $\leftarrow$ flag($e$) $- 1$
                end if
            end for
        else
            for each edge $e \in \Delta_j$ do
                flag($e$) $\leftarrow$ flag($e$) $+ 1$
            end for
        end if
    end for
    $R \leftarrow \{\emptyset\}$
    for each simplex $\Delta_j \in T$ do
        for each edge $e \in \Delta_j$ do
            if flag($e$) $\geq 0$ then
                $R \leftarrow R \cup e$
            end if
        end for
    end for
    $\mathcal{B} \leftarrow$ find boundary points of $\mathcal{C}$ using $R$ (Algorithm 2)
    for each point $\mathbf{f}_i \in \mathcal{B}$ do
        $D(\mathbf{f}_i) \leftarrow d$
    end for
    $\mathcal{C} \leftarrow \mathcal{C} - \mathcal{B}$
    $d \leftarrow d + 1$
until $\mathcal{C} \neq \{\emptyset\}$
return $\{D(\mathbf{f}_1), D(\mathbf{f}_2), \ldots, D(\mathbf{f}_n)\}$
```

During filtration, the resulting simplicial complex loses the higher dimensional simplexes and shrinks by gradually developing cavities. These cavities may join to form tunnels, and even holes may appear. Intuitively, a piece of the polytope disappears when α becomes small enough so that a sphere with radius α, or several such spheres, can occupy its space without enclosing any of the points of $\mathcal{C}$. The resulting non-convex hull of the simplicial complex is called the α-hull of $\mathcal{C}$. In order to perform this filtration, we need to find the circumradius of the hyper-sphere of each simplicial complex; this can be achieved as follows [36]: for any collection of n points $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\} \subset \mathbb{R}^m$, we denote by $\mathbf{D}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n) \in \mathbb{R}^{m \times m}$ the matrix whose entries are the squares of the pairwise distances:

$$\mathbf{D}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n)_{ij} = ||\mathbf{f}_i - \mathbf{f}_j||^2. \tag{4.1}$$

We refer to $\mathbf{D}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n)$ as the Euclidean distance matrix (EDM). If we have EDM for all the points in an m-simplex, The circumradius r can be computed by:

$$r = \left[\frac{\det \mathbf{D}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n)}{2 \cdot \det \begin{pmatrix} 0 & 1 \\ 1 & \mathbf{D}(\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n) \end{pmatrix}} \right]^{\frac{1}{2}}. \tag{4.2}$$

Although there are other (recursive) methods to compute the exact circumradius [37], we use equation 4.2 since it gives a very close value to the exact radius and it is also fast to compute. Next, whenever $r > \alpha$, we need to exclude the corresponding edges of the enclosed simplex. Once the edges are removed from the simplex, we need to identify the vertices that reside on the boundary of the resultant hull. Such vertices can be counted using the following lemma:

Lemma 4.1. An edge e (1-simplex) resides on the boundary of the m-dimensional data points $\mathcal{C}$, if and only if it belongs to maximum of $(m-1)$ m-simplex in $T = \{\Delta_1, \Delta_2, \ldots, \Delta_k\}$, where T is a set of m-simplexes found from the triangulation of $\mathcal{C}$.

Algorithm 2 Find Boundary Points

```
Require: Set of all $m$-simplexes $T = \{\Delta_1, \Delta_2, \ldots, \Delta_k\}$ in $\mathcal{C}$
Require: A subset of edges $R$ from $T$ after filtration.
for each edge $e \in R$ do
    $\texttt{count}(e) \leftarrow [0, 0, \ldots, 0]$ s.t. $|\texttt{count}(e)| = k$
end for
for each edge $e \in R$ do
    for each $m$-simplex $\Delta_j \in T$ do
        if $e \in \Delta_j$ then
            $\texttt{count}(e)[j] \leftarrow 1$
        end if
    end for
end for
$\mathcal{B} \leftarrow \{\emptyset\}$
for each edge $e = \{\mathbf{p}, \mathbf{q}\} \in R$ do
    if $\sum_j(\texttt{count}(e)[j]) \leq (m-1)$ then
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{\mathbf{p}, \mathbf{q}\}$
    end if
end for
return $\mathcal{B}$
```

Proof 4.1. Any two m-simplexes Δ_i and Δ_j can meet with each other at any of $\{0, 1, 2, \ldots, (m-1)\}$-faces. If they meet at 0-face (i.e. vertex), then there is no common edge. If they meet at 1-face (i.e. edge), then there is only one common edge. Inductively, if they meet at an $(m-1)$-face, then there are $(m-1)$ common edges. Therefore, if two m-simplexes meet with each other by touching a maximum number of edges, then the upper bound of the number of common edges is $(m-1)$. Hence, any edge that belongs to more than $(m-1)$ m-simplexes, cannot be an edge on the boundary of $\mathcal{C}$.

Line 37 in Algorithm 1 applies lemma 4.1 and this can be achieved simply by keeping a flag on each edge. This flag keeps the count of how many simplexes a certain edge belong to. If the count is less than or equal to $(m-1)$, then we can decide that the edge belongs to the boundary of $\mathcal{C}$ and returns those vertices as boundary points of the α-hull. This operation has been implemented in Algorithm 2.

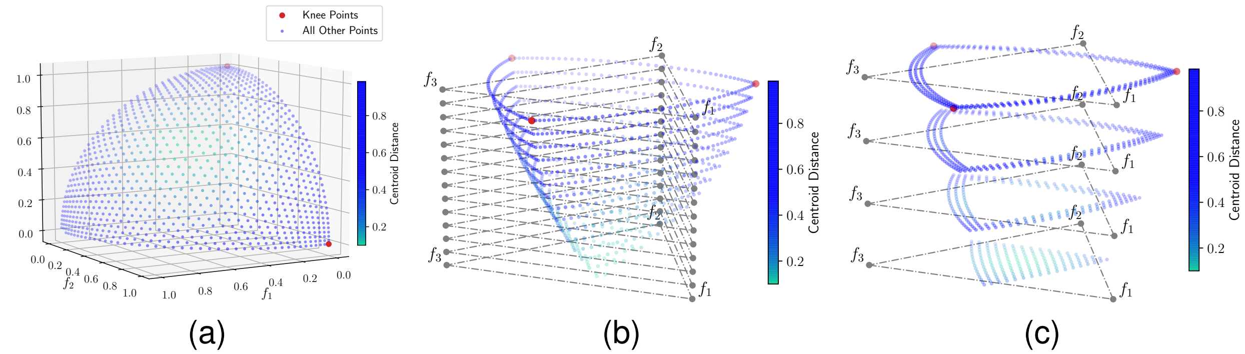


Figure 4.1: Steps of Palette visualization on a simple spherical three-dimensional Pareto-optimal front.

4.1.2 Representing Points in Two Dimensions

Once we compute the depth contours, we can consider the points within the same depth equivalence class as one *layer* of the high-dimensional data-set. The α-shape algorithm works by decomposing the point cloud in a layer-wise manner. Once this decomposition is performed, points from each layer are represented in a two-dimensional Radviz or Star-coordinate plot [20]. Although we have used Radviz/SC-plot, other mapping techniques like SNE/t-SNE [23], [38] can also be used. As there are multiple layers (depth equivalence classes) in a point cloud, our method stacks each layer along the z-axis. The x-y axes are used to represent the SC-plot[2] of each layer. The top-most two-dimensional SC-plot corresponds to the boundary layer, the next one below corresponds to the next to boundary layer and so on. The bottom SC-plot corresponds to the inner-most points (close to the central region with the highest point depth) in the original high-dimensional space. Since the third dimension (z-axis) separates different layers, we call our visualization a $2\frac{1}{2}$-dimensional representation of the points, allowing a DM with a bottom-up approach to look at the points in a lower and manageable dimension, and associate them with their true location in the original high-dimensional space. An illustration of the mapping procedure is presented in Figure 4.1. The Figure 4.1a shows how each depth equivalence class is enumerated: starting from the boundary, the outermost layer is colored dark blue and as we proceed to the central region of

[2] Readers can assume all the mapping of the depth contours are done on SC-plot, unless otherwise stated.

$\mathcal{C}$, they turn green. Figure 4.1b shows how each layer is represented using a SC-plot and how they are stacked on top of each other. The top-most SC-plot corresponds to the outermost layer and the bottom-most SC-plot corresponds to the points residing in the central region in the original space. Here we have 17 such layers. In order to reduce the visual clutter, we have merged them into four layers, as shown in Figure 4.1c.

4.1.3 Addressing MCDM Issues in the End Visualization

With a layer-wise decomposition of the points through multiple two-dimensional plots, DMs have preserved the centrality information of the points (also, their *center-outward orderings* [31]). In addition, DMs would now be interested in getting more information about three different properties of the data-set: (i) geometric properties, (ii) performance properties, and (iii) preferential properties. For this purpose, we use different color schemes and marker sizes to bring our different properties in the data-set in each layer. Different metrics for each property can be considered, but in this dissertation, we use the following three properties:

1. **Geometric (Boundary to Centroid[3])**: In addition to placing the points in different layers based on the α-shape method, points that are far away from the centroid of the data-set are painted as dark-blue and as we go toward the center of the point cloud, the color gradually turns into light-green. The most central point has a light-green shade. In this study, a Euclidean distance metric is used to measure the distance from the centroid. In some scenarios, some boundary points may be close to the centroid due to the shape and density of the data-set. Moreover, the DMs have another visual means to identify the closeness of a point to the centroid in addition to its being close to the boundary. See Figure 2.5.

2. **Performance (Constraint Boundary)**: In the presence of constraints, the closeness of a point (objective vector) to any constraint boundary may be of importance to the

[3] The centroid of the data-points $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$ is computed as their center of mass, i.e. $\bar{\mathbf{f}} = 1/n \sum_{i=1}^{n} \mathbf{f}_i$. Other centrality measures could also be used.

DM. Hence, points that are close to a constraint boundary [39] are painted in pink color and points well inside the feasible region are colored in cyan. See Figure 5.5a.

3. **Preferential (Large Trade-off)**: Points with large trade-off values are marked in a dark-red circle. A point having a trade-off θ value larger than $(\mu_\theta + 3\sigma_\theta)$ of the entire data-set is shown in dark-red color[4]. The points with a larger θ than the above threshold are marked with a bigger marker in dark-red color. See Figure 2.5. For the calculation of trade-off value θ for a point, we follow the method presented in [25].

In this study, a palette visualization plot or, a PaletteViz plot, will have multiple layers of two-dimensional SC-plots with non-dominated points painted with either item 1 or item 2 above. The points with large trade-offs (item 3), if any, are marked in dark-red color of varying size. A knee point, having a very large trade-off, will be marked with a big dark-red circle, so that it is clearly visible to the DM.

4.1.4 Calculation of Trade-off

In order to compute the trade-off value of a point, we follow the method presented in [25]. The trade-off value of a solution[5] $\mathbf{x}_i$ is first computed with respect to another non-dominated point $\mathbf{x}_j$ as the net loss in some objectives offset by the accompanying gain in other objectives as a result of substituting $\mathbf{x}_i$ with $\mathbf{x}_j$:

$$\mathcal{T}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{p=1}^{m} \max[0, f_p(\mathbf{x}_j) - f_p(\mathbf{x}_i)]}{\sum_{p=1}^{m} \max[0, f_p(\mathbf{x}_i) - f_p(\mathbf{x}_j)]}. \tag{4.3}$$

Then, the trade-off value of $\mathbf{x}_i$ is computed as the minimum $\mathcal{T}(\mathbf{x}_i, \mathbf{x}_j)$ value for all ϵ-neighboring solutions of $\mathbf{x}_i$ in the objective space:

$$\theta(\mathbf{x}_i) = \min_{\forall j: ||\mathbf{f}(\mathbf{x}_i) - \mathbf{f}(\mathbf{x_j})|| \leq \epsilon} \mathcal{T}(\mathbf{x}_i, \mathbf{x}_j). \tag{4.4}$$

[4] Here, we are assuming that a DM might only be interested in seeing the points with trade-off values within the range of $(\mu_\theta + 3\sigma_\theta)$. [5] Here by $\mathbf{x}$, we denote a solution to the MOP problem (*design variable vector*) and its corresponding *m objective function* values are $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$.

A solution with a large value of the trade-off $\mathcal{T}(\mathbf{x}_i, \mathbf{x}_j)$ within a ϵ-neighborhood of $\mathbf{x}_i$ signifies that the solution receives a bad trade-off if selected against one of its neighbors, therefore we take the minimum of all $\mathcal{T}(\mathbf{x}_i, \mathbf{x}_j)$ values.

4.2 Other Issues

4.2.1 The Number of Layers

The choice of the number of layers in a PaletteViz plot along the z-axis is DM-dependent. This may depend on the properties of the data-set as well. However, in most of our illustrations here, we use three to four layers, but a layer can be further divided into a number of sub-layers to finely distinguish centrality of the data-set. We attempt to distribute all points equally among the layers in this study. It is interesting to note that the division of layers can also be made based on the distance to the centroid or using any other metrics that are important for the DM to visualize the entire data-set in a hierarchical manner. Our methodology does not exclude any of such possibilities.

4.2.2 Clusters of Disjointed Sets

The entire Pareto-optimal front set may be constituted with a number of disconnected regions in the original high-dimensional objective space. Such an information is of great importance to DMs. Due to loss of dimensionality, most high to low dimensional mappings, including Radviz or SC-plot, fails to show a clear separation of the clusters of data-set in the two-dimensional mapped space. In such a scenario, we propose to apply a clustering algorithm on the original data-set in the high-dimensional space and then make a separate PaletteViz plot for each cluster of data points. In this study, we use an unsupervised algorithm, DBSCAN [40] and Mean-Shift [41] algorithm, for this purpose, but any other clustering method can also be used. Also, a DM may want to use PaletteViz on the entire data-set to get an idea of the relative location of clusters and then redo PaletteViz plots for each cluster to get a more detailed information. The entire procedure can be summarized in Algorithm 3.

Algorithm 3 Palette Visualization Procedure

```
Take all the points, use a suitable clustering algorithm [40] and find $n$ different clusters $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_n\}$.
For each cluster $\mathcal{C}_i$, compute trade-off value [25], constraint violation [39], distance from the centroid, etc. for each high-dimensional Pareto-optimal point within the cluster.
Use the approximate $\alpha$-shape algorithm to find the depth contour $\mathcal{B} = \{D(\mathbf{f}_1), D(\mathbf{f}_2), \ldots D(\mathbf{f}_n)\}$ of $\mathcal{C}_i$.
Using the depth equivalence classes, decompose the data points into multiple layers $\mathcal{C}_i \rightarrow \{L_1, L_2, \ldots, L_k\}$
Use Radviz to display each layer $L_i$ on a two-and-half dimensional space in which the sequence of layers is stacked along the $z$-axis.
Use the color-scheme and point sizing as discussed in Section 4.1.3 to represent different features of the Pareto-optimal points.
```

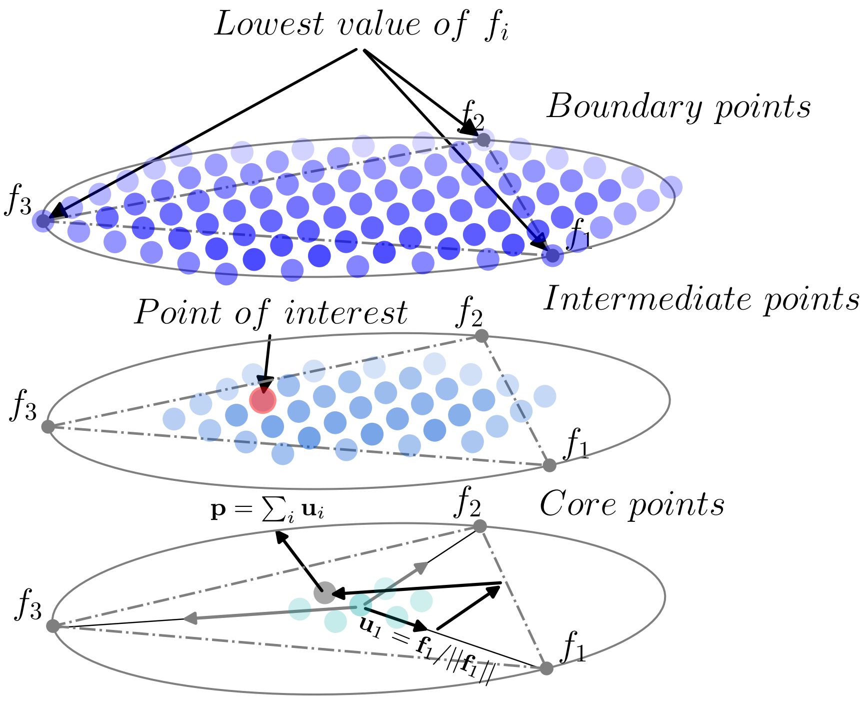


Figure 4.2: PaletteStarViz representation of the cube data set presented in Figure 2.1. The plot indicates that the red point is an interior point (middle layer), but not at the core (bottom layer). Also the point has a relatively smaller f_3 value.

If we apply PaletteViz on the cube data points in Figure 2.1a , it generates a plot presented in Figure 4.2. In this figure, the points on the outer-most depth-contour is mapped on the top of the PaletteViz layer, the points from the inner-most depth contour falls on the bottom layer of the PaletteViz. The *point of interest* appears on the middle layer of the PaletteViz which means this point is neither on the outer shell nor the inner part of the data points in the high dimensional space (in this case, it is three dimensional). The anchors correspond to the points with the lowest (i.e., the best) objective function values. The relative positions of the other points from the anchors can be understood how they are mapped in an SC-plot. For example, the position of the gray point on the bottom layer is interpreted as this is farthest from anchor f_1 – which means this point has the worst objective value in terms of f_1.

In PaletteViz , the neighborhood relationships are analogous across the vertical direction. For example, if we take an arbitrary point on the intermediate layer and imagine a vertical line passing through it, then the points on the other layers that are close to the vertical line are also close to that point on the intermediate layer.

4.3 Chapter Summary

In this chapter, we introduced and discussed the Palette visualization method, i.e. PaletteViz. We started with the basic topological concepts like *depth-contour* and *depth equivalence class* to explain how the *shape* of the data point cloud $\mathcal{C}$ can be captured in a lower dimensional space. Then we proceeded to explain the algorithm in detail. We also argued how such transformation is able to respect the topological properties of $\mathcal{C}$ through *inward-outward* relationship of the data points. We also showed how hull of the α-shape is utilized to achieve such a goal.

In the next chapter we are going to discuss the resultant plots obtained from a set of benchmark scalable Pareto-optimal fronts and how PaletteViz can provide us with interesting insights into the data points.

CHAPTER 5

RESULTS AND COMPARATIVE ANALYSIS

To demonstrate the working of our proposed palette visualization technique, we consider a number of multi- and many-objective optimization problems. For the ease of visualization, we group the original layers from the high-dimensional space in PaletteViz plots into three or four layers. Note that since none of the existing EMO visualization methods [42] consider the topological properties of the Pareto set, we have only compared PaletteViz plots with the most widely used methods, such as PCP, RadViz, and Heatmap etc. Moreover, we employ two different type of color-schemes, depending on the nature of the multi-/many-objective optimization problem:

- **Depth-Contour-Based Color Scheme:** Generally, where there is no constraint function, we color the data points according their depth-contours. The points that are closer to the central region are colored light green; as the points approach the boundary, they are increasingly colored dark blue; i.e., light green points become dark blue as they are farther away from the central region.

- **Constraint-Violation-Based Color Scheme:** If we are visualizing constrained problems, then we use the cumulative normalized constraint violation values for coloring. The points with higher normalized constraint violation values are colored in pink and as they have less constraint violation values, they turn cyan. i.e. cyan points turn pink as they get close to the one of constraint boundaries.

- **Trade-off Based Color Scheme:** Data points with higher trade-off values are colored in dark red so that they are easily distinguishable from the rest of the data points.

- **Trade-off-Based Point Sizing:** The size of the points will be increased with respect to their trade-off value $\mathcal{T}$, regardless of their colors.

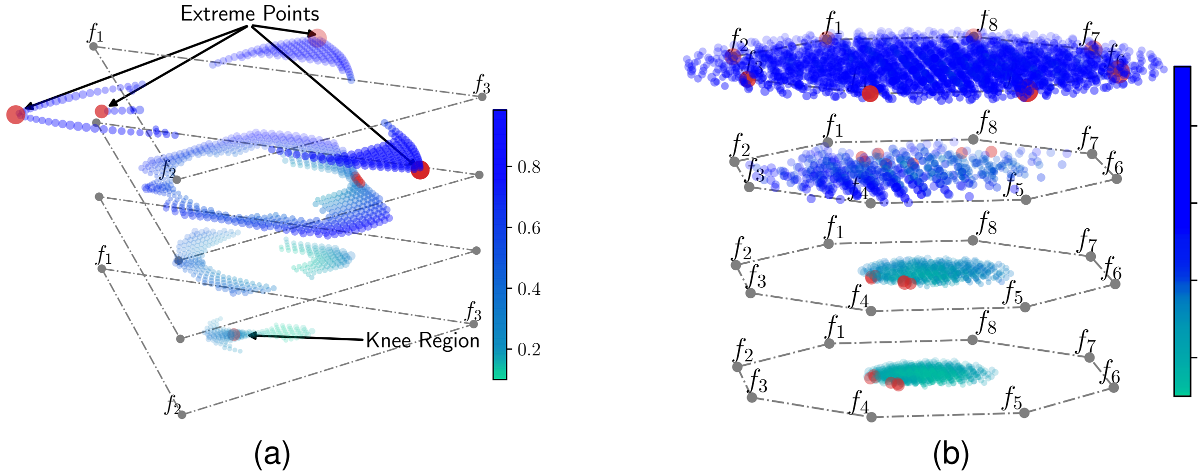


Figure 5.1: (a) PaletteViz plot for a three-objective knee problem (DEB3D1K) presented in Figure 2.5. The data-set consists of 1,143 points. (b) PaletteViz plot for the eight-objective knee problem having 3,432 non-dominated points.

5.1 Pareto-optimal Fronts with Knees

First, we use a benchmark test problem from [1], called DEBMDK. The problem has a parameter that can be tuned to introduce multiple *knees* (i.e. high trade-off points). However we choose the problem with a single knee region for an easier demonstration of our proposed palette visualization technique. We call this version of DEBMDK as "DEBMD1K" and the function definition is given in appendix A.1.1.

Results for three-, and eight-dimensional (i.e. objective) DEBMD1K problems are presented in Figures 5.1a and 5.1b, respectively. In Figure 5.1a, the bottom layer corresponds to the deepest (near core) points in the original dimension. A few knee solutions (in dark-red color) in this layer are clearly visible. Three other dark-red points on the top-most layer close to each objective-worst points indicate that a large trade-off exists at these three extreme points. Since they appear on the top-most layer showing mostly the boundary points, these high trade-off points may not represent knee points in its traditional sense, but with respect to their neighboring points, they exhibit a large trade-off. In the sense of trade-off information alone, a DM would first be interested in filtering these few large-trade-off points with more emphasis on the knee point appearing on the bottom-most layer. Division of points

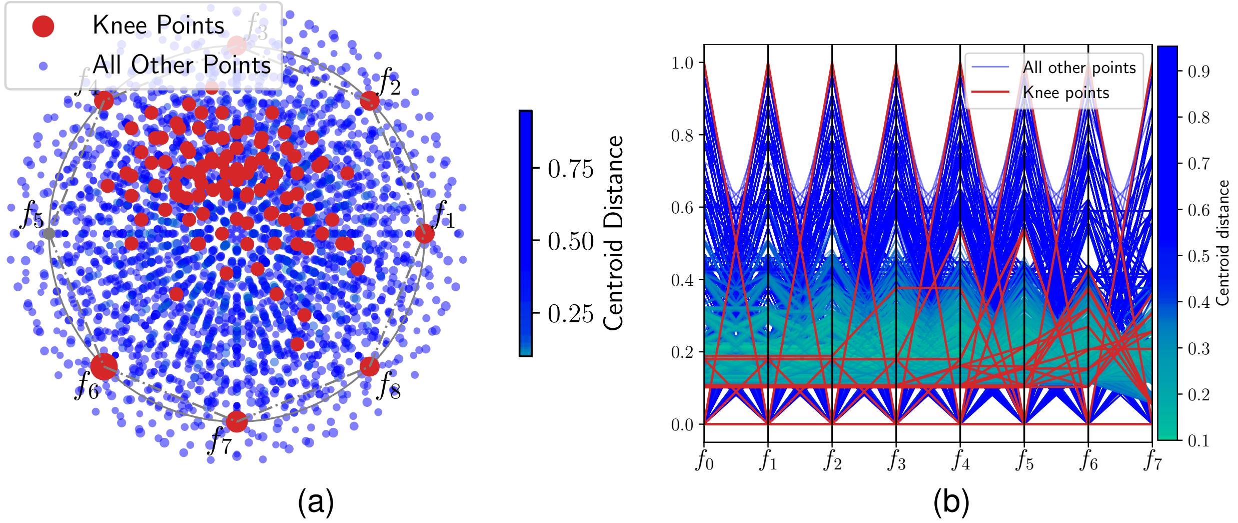


Figure 5.2: (a) An original Star-coordinate (SC) plot of the eight-objective knee problem on the same data-set used to make the PaletteViz plot in Figure 5.1b. (b) A parallel coordinate plot (PCP) of the eight-objective knee problem presented in Figure 5.1b.

into layers based on the depth-equivalence class and marking large trade-off points clearly allowed the PaletteViz technique to effectively filter out a few knee points and other large trade-off points from 1,143 non-dominated points.

Next, we apply the palette visualization to the eight-objective DEBMD1K problem in Figure 5.1b. Two knee points in the bottom-most layer are clearly identifiable from the PaletteViz plots and may be of great importance to DMs. Moreover, the existence of a few boundary points with large trade-offs is also evident from the plots. Existence of relatively few core points in the Pareto set is also evident.

For comparison, we visualize the eight-dimensional problem using the original SC-plot in Figure 5.2a. The SC-plot cannot show any boundary-core relationship among the points. Also, all points seem to be clustered around the central part of the Radviz circle, thereby not providing much information to a DM. As somewhat understandable from the colors, the boundary points cover the core points hence we cannot visualize any of the light green points on the plot. Moreover, it is also not possible to understand where exactly the knee points are located in the high-dimensional space. The knee points are colored in red and the two arbitrary boundary points are colored in blue. The rest of the points are colored in light gray. If there were no coloring applied, it is impossible to comprehend which points are on the knee or on the boundary, or what is even the structure of the high-dimensional data point cloud.

For further comparison, Figure 5.2b shows all points and marks three extreme points with a large trade-off in blue color and the knee points in red color on the parallel coordinate plot (PCP) [17]. The PCP is unable to indicate any small or large trade-off information associated with them. The information about their geometric location – near boundary or near the core – is not inferable from the PCP.

5.1.1 Pareto-optimal Front with Constrained Knees

Sometimes, a DM might want to concentrate on a certain portion of the Pareto-optimal space. For example, in the DEBMD1K problem, a DM might want to explore the high trade-off region in more detail. To simulate this scenario we introduce another multi-objective optimization problem that is adapted from the formulation of the preceding section. We introduce a constraint function that makes only the knee region of the DEBMD1K problem feasible. We call this problem CDEBMD1K and the exact formulation is presented in appendix A.1.5. An example Pareto-optimal front in three dimensions is presented in Figure 5.3a.

Here we adopt the color scheme based on the cumulative normalized constraint violation value (i.e. pink to cyan). In the figure we see the knee region has a number of larger points in red while the rest of the data points are colored in cyan to pink. The pink points are closer to the constraint function. A PaletteViz plot for this data set is presented in Figure 5.3b. In Figure 5.4. we demonstrate PaletteViz plots for a four-dimensional CDEBMD1K problem. Observant readers can notice an interesting fact about this plot – the high trade-off values are closer to anchor f_2, which means these points have smaller values in objective f_2 and larger values in the other three objectives, i.e. f_1, f_2 and f_3. The points in the bottom layer have mostly cyan color, therefore it means those solutions are farther from the constraint function, which makes these points more interesting to the DM.

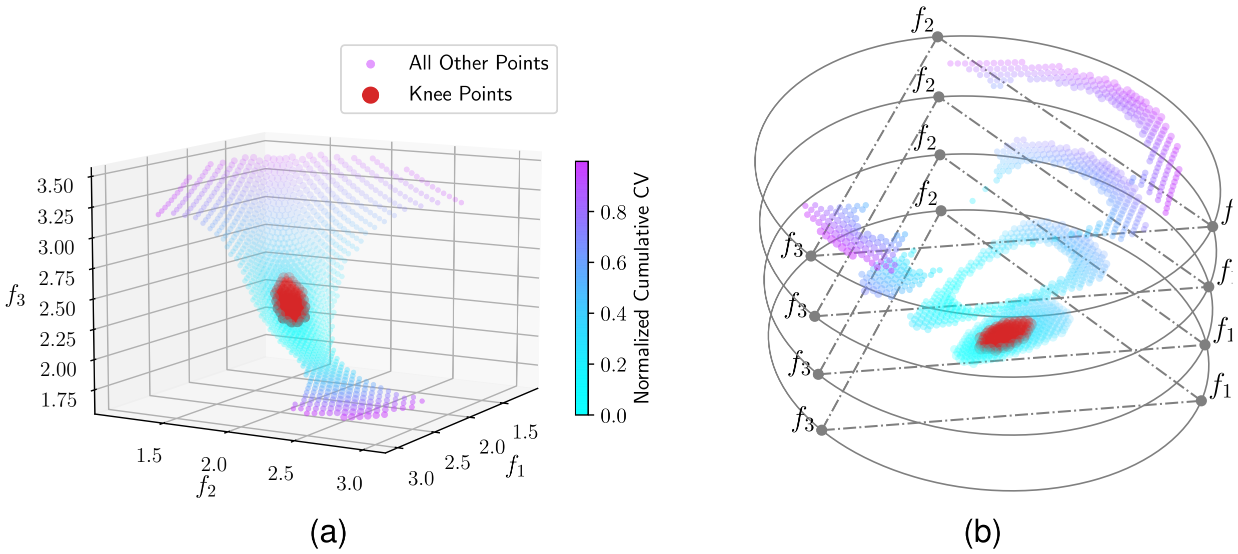


Figure 5.3: (a) A 3D scatter plot for the CDEBM1K Pareto-optimal front (see appendix A.1.5) with 1,049 data points. (b) A PaletteViz plot for the three dimensional CDEBM1K problem.

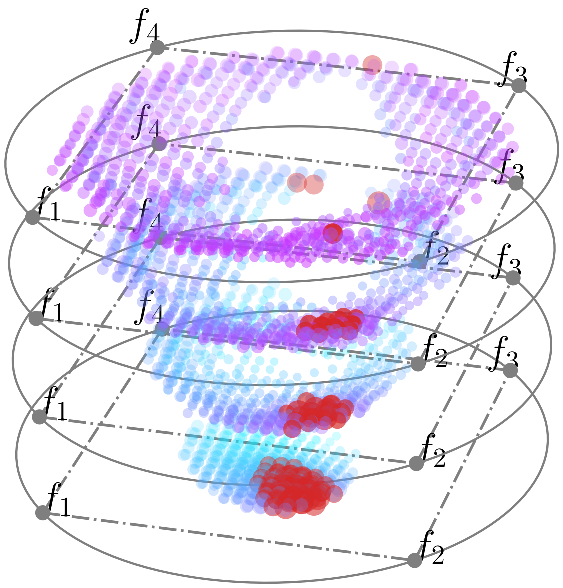


Figure 5.4: A PaletteViz plot for four-dimensional CDEMBM1K problem with 2,042 data points.

5.2 Pareto-optimal Fronts with Isolated Regions

Here we are interested to see if the proposed palette visualization method can address the topological features of high-dimensional points. In order to simulate this scenario, we modify the DTLZ2 problem [43] by slicing the Pareto set using two constraint functions in such a

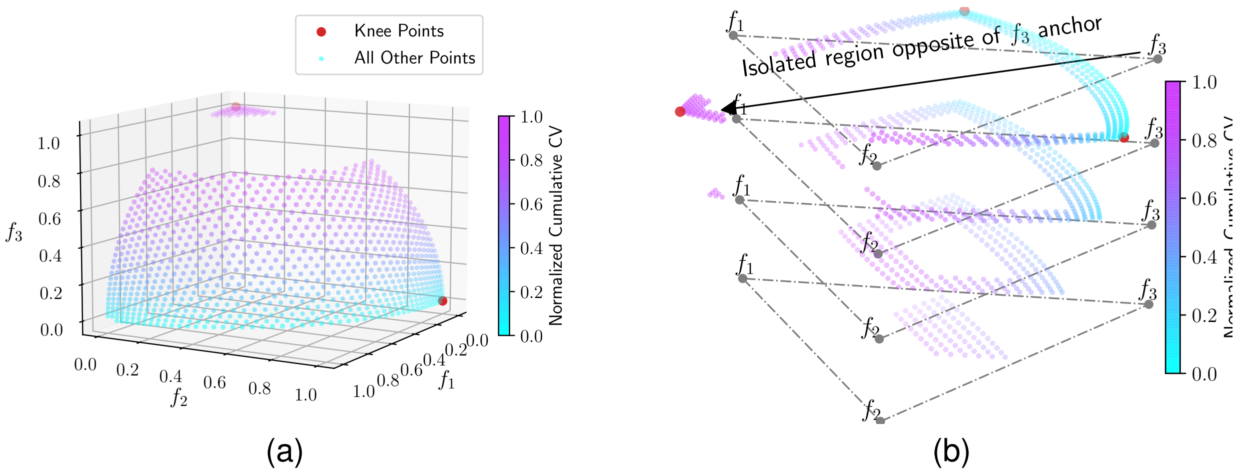


Figure 5.5: PaletteViz plot for the Pareto-optimal front of C0DLTLZ2 problem (see appendix A.1.2) in three dimensions.

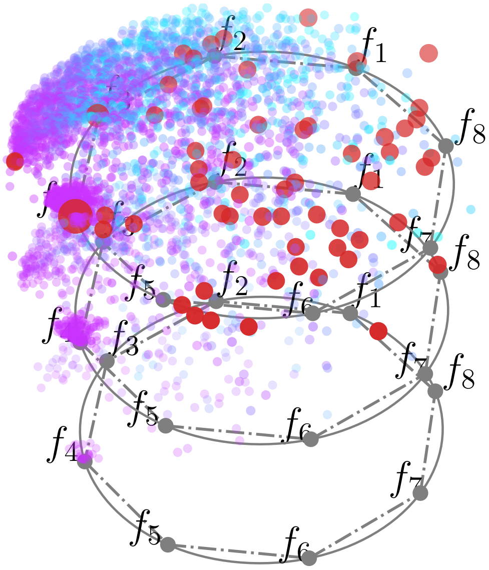


Figure 5.6: PaletteViz plot for the Pareto-optimal front of C0DTLZ2 problem in eight dimensions.

way that two clusters are created (see appendix A.1.2). We call this problem C0DTLZ2. A three-dimensional scatter plot is shown in Figure 5.5a. The isolated region opposite to the f_3 anchor (i.e. the maximum of f_3 in the original dimension) is clearly identifiable. There are 983 data points in this Pareto set.

Figure 5.6 demonstrates a PaletteViz plot for the isolated Pareto set in eight dimensions. The isolated region opposite to the f_8 anchor (i.e. the maximum of f_8 in the original dimension) is clearly identifiable. There are 4,005 points in this data set. Here we can see one cluster has more points than the other. Thus the smaller cluster creates an isolated region in the objective space. When an EMO algorithm is applied to this problem, it is expected that a few solutions will be found in the smaller region and a DM may be highly interested in analyzing at least one solution from this place, as such a point comes with best choices on f_1 and f_2.

The PaletteViz plots show two clearly distinct regions, one having a few points (opposite of f_3 and f_8 anchor points) and the other having the majority of its points in the middle. This demonstrates how the cluster information in high-dimensional space is carried down to the PaletteViz plots. It is also interesting to note that the extreme point on the isolated region also has a large trade-off point, marked in dark-red color. The fact that this point appears on the top-most layer signifies that this large trade-off point is a boundary point. Also, the absence of large trade-off points in bottom layers indicates that the interior of the larger cluster region has no knee-like point for this problem for the DM to have an obvious selection of an interior point. Certain small-sized dark-red circles in the second layer for the eight-objective problem come from the lack of an adequate number of neighboring points to decrease their trade-off values.

Along with all the above information, the capability of visually filtering out points from 4,005 to a handful of large trade-off points and a few isolated points is a tremendous advantage derived from PaletteViz.

5.3 Pareto-optimal Fronts with Differing Dimensions

In many real-world problems, a Pareto-optimal front can consist of clusters of data points in different dimensions. For example, a Pareto set may consist of a mix of spherical and planar spaces, and even a part of the data point cloud $\mathcal{C}$ can be composed of lower-dimensional components. In order to see how PaletteViz technique works on such a scenario, we consider DTLZ8 problem [43], where the Pareto set consists of a $(m-1)$-dimensional hyper-plane and

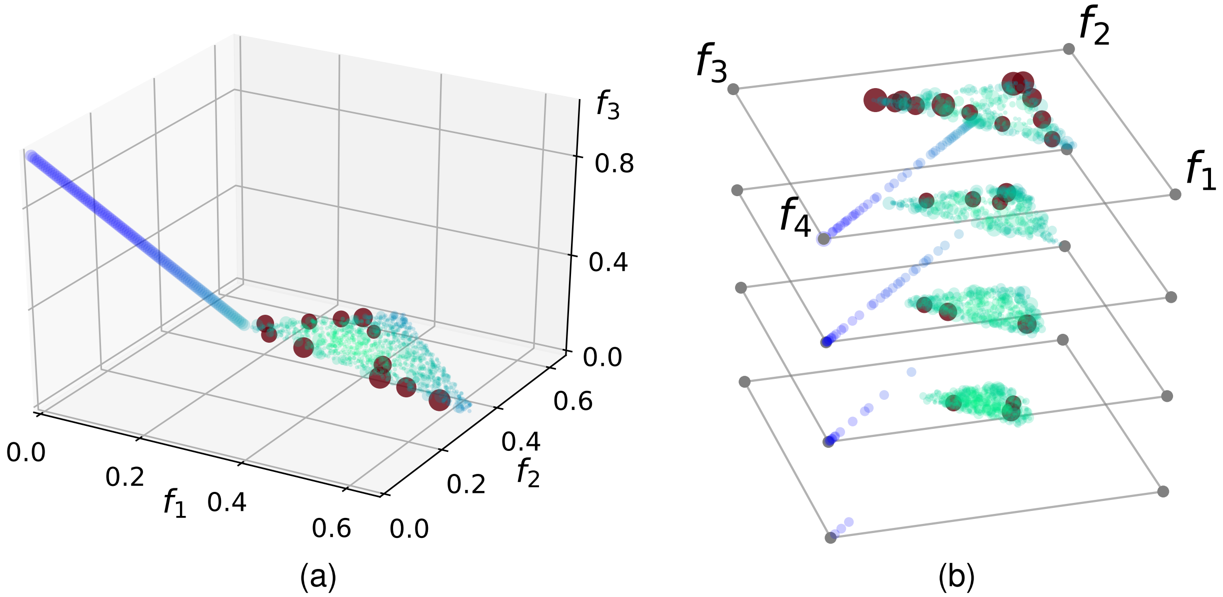


Figure 5.7: Three-objective Pareto set (2,105 points) for the DTLZ8 problem (see appendix A.1.4).

a one-dimensional straight line. The corresponding three-dimensional Pareto set is presented in Figure 5.7a. The Pareto set consists of two geometrically different point clouds – a straight line is connected with an $(m-1)$-dimensional hyper-plane. The PaletteViz plot of the four-objective DTLZ8 Pareto set contains 2,105 points. An analytical expression for this problem is described in appendix A.1.4.

The following decision-making aids can be obtained: (i) Data set consists of a straight line and a four-dimensional surface, (ii) a few exceptionally high trade-off points exist; most of them are near the boundary, (iii) no large trade-off point exists on the straight-line part of the set. The PaletteVizof a four-objective version of the problem is presented in Figure 5.7b. In the next figures (Figure 5.8a and Figure 5.8b), we demonstrate the results of PaletteViz applied to six- and eight-dimensional DTLZ8 problems. Here we show the results with color coding derived from the cumulative normalized constraint violation values. Due to the number of constraint functions bounding the (six- and eight-dimensional) hyperplanes, most points are – one or another way – closer to one of the constraint functions. As a result, almost all of them are pink. However, there are a small number of robust solutions (cyan points) that can be seen near the meeting point of the line and the hyperplane.

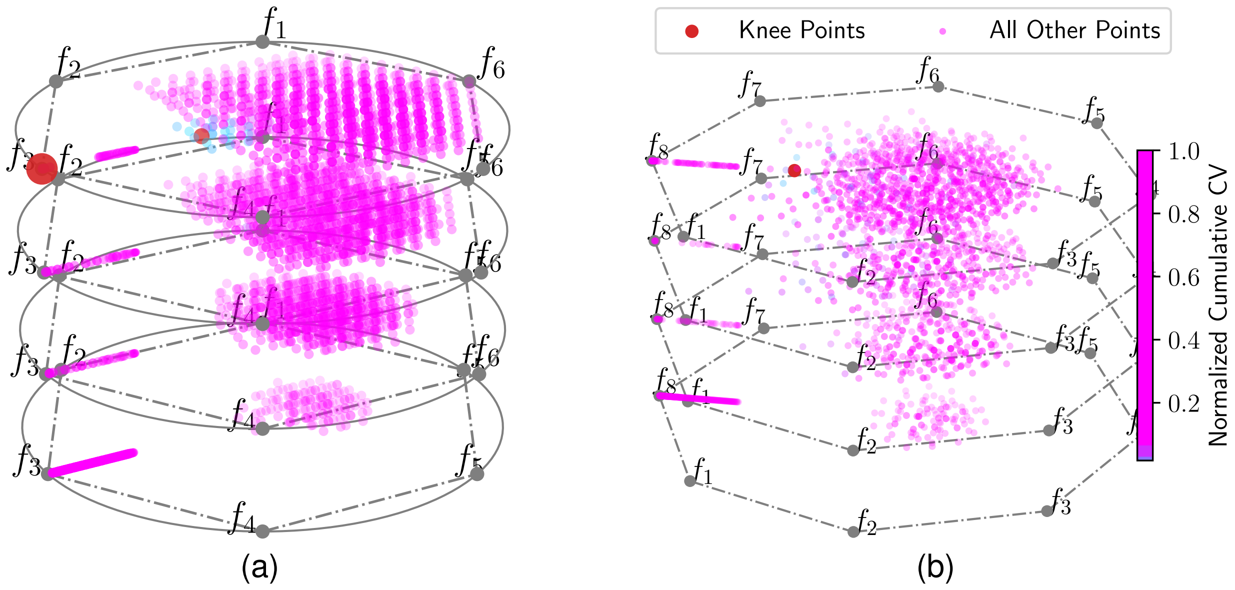


Figure 5.8: (a) PaletteViz for six objective DTLZ8 problem (3,535 data points). (b) PaletteViz for eight-objective DTLZ8 problem (2,277 data points). For eight dimensional DTLZ8, we use Radviz for the mapping of the depth contours.

The PaletteViz plot allows a DM to look at only about 10 12 robust points (marked in cyan color) from a set of 3,535 and 2,277 non-dominated points. If the DM prefers high trade-off points, only a few red points can be further looked into, however if the DM prefers a point which lies at the core of the data-set (with a good compromise of all four objectives) then the couple of pink points in Layer 3 (from top) can be further looked into.

5.3.1 Disjoint Pareto-optimal Fronts

As we have discussed before, in the presence of a Pareto set having multiple disjoint clusters in the original high-dimensional objective space, a DM may be interested in the boundary points of each cluster. To illustrate this scenario, we have used the palette visualization on three- and five-objective C2DTLZ2 problem [39]. The Pareto set consists of multiple clusters that are generated by applying $(m+1)$ constraints on the DTLZ2 [43] problem. An analytical expression for this problem is described in appendix A.1.3.

A three-dimensional scatter plot of C2DTLZ2 is presented in Figure 5.9a. Three extreme points have a large trade-off, due to the concavity of the Pareto-optimal surface. Figure 5.9b

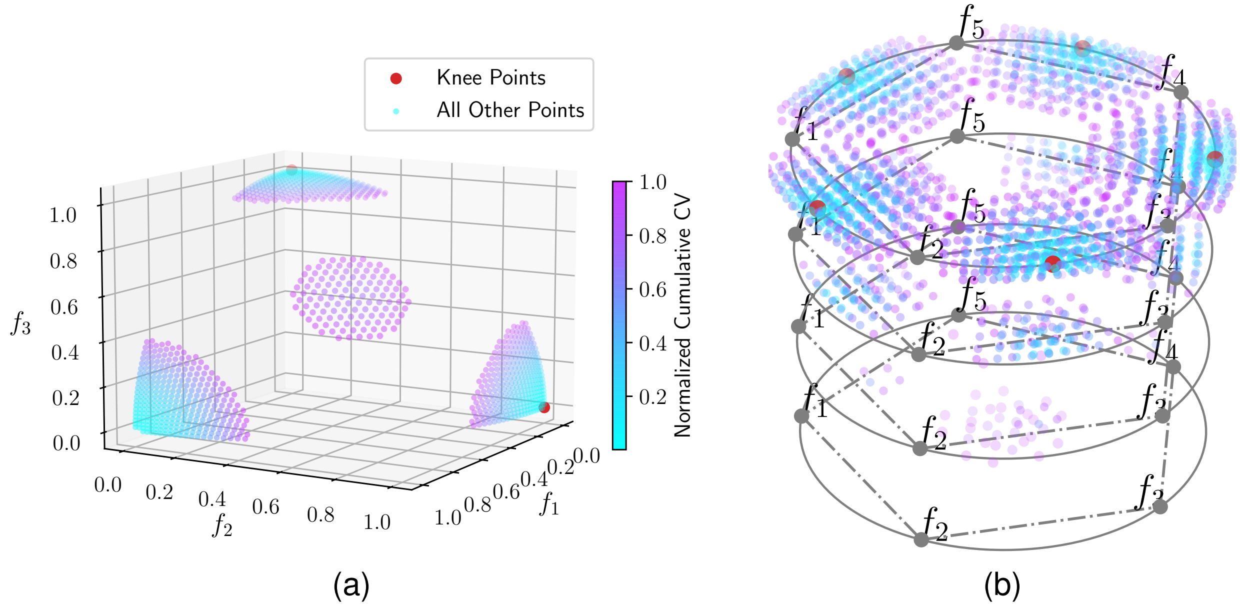


Figure 5.9: (a) Pareto set for the three-objective C2DTLZ2 problem (see appendix A.1.3). There are four isolated regions on the data-set. The data-set contains 1,036 points. (b) PaletteViz plot of the five-objective C2DTLZ2 problem indicating five extreme and one central cluster in the original data-set containing 2,280 points.

shows the PaletteViz plot for a five-objective C2DTLZ2 problem. Six clusters and their relative locations are noticeable from the PaletteViz plot. The pink points are closer to a constraint boundary and cyan colored points are well inside the feasible region. It is interesting to note that the top-most layer has mixed pink and cyan-colored, providing vital information to a DM. Not all boundary points are on constraint surfaces. If a boundary point near the constraint surface is desired, the DM can only analyze the pink points further. Otherwise, if a core point near the constraint surface is desired, a pink point from one of the bottom layers can be chosen.

In order to compare PaletteViz plot with another existing visualization method, we plot the same data-set for the five-objective C2DTLZ2 problem in a heatmap [11] visualization in Figure 5.10a. For this particular application, heatmap does not provide much information about six clusters that the data possess and nearness of any point to a constraint boundary. All the points are sorted in descending order with respect to f_1. The cluster closer to extreme f_1 value has a large difference in f_1 values from other clusters. For the curious reader, we also include a PCP for the same problem, which is present in Figure 5.10b. Here we see PCP

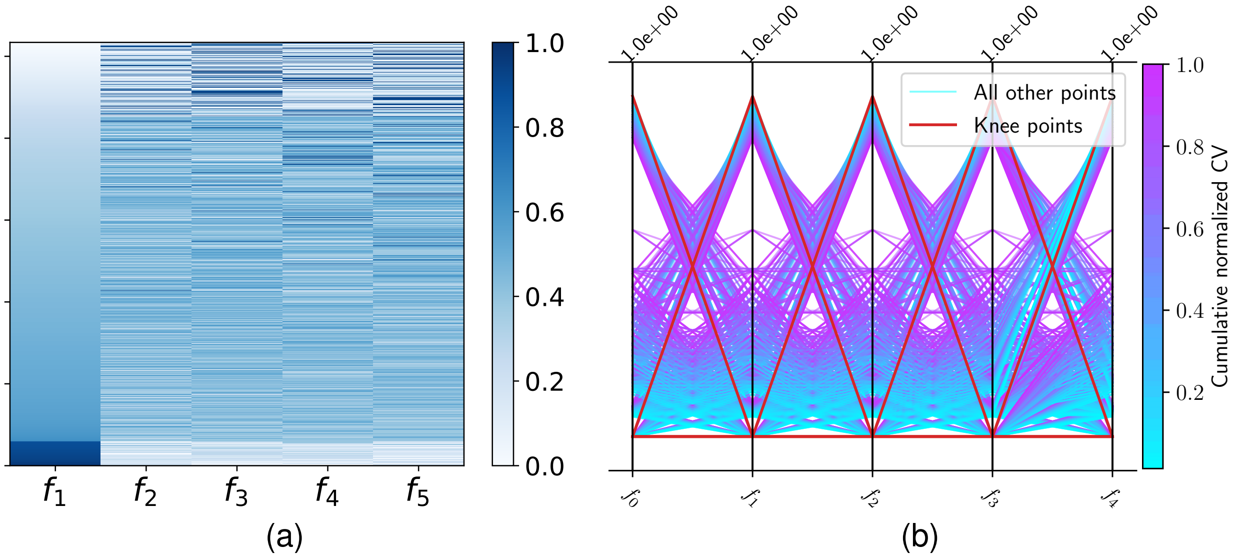


Figure 5.10: Heatmap and PCP for five-objective C2DTLZ2 problem.

neither gives any useful insight about any aspect of the topology of the data point cloud, nor the trade-off information. It's not even possible to infer there are multiple clusters of points in $\mathcal{C}$, unless otherwise such information is available beforehand.

5.4 PaletteViz for Engineering Design Problems

So far, we have seen the proposed palette visualization technique can address different scenarios that a DM may come across during a decision-making process. Next, we apply PaletteViz on two engineering design problems.

5.4.1 Vehicle Crash-Worthiness Problem

We apply our method to the vehicle crash-worthiness problem from [2]. This is a three-objective optimization problem, however the Pareto set in this problem is observed to have two separate clusters, as shown in Figure 3.9a.

Although both clusters together can be presented in a single PaletteViz plot (Figure 5.11a), here we show that more detailed information of each cluster can be observed from their individual plots. The palette visualization of the top cluster (cluster A) is presented in Figure 5.11b and that of the bottom cluster (cluster B) is presented in Figure 5.11c.

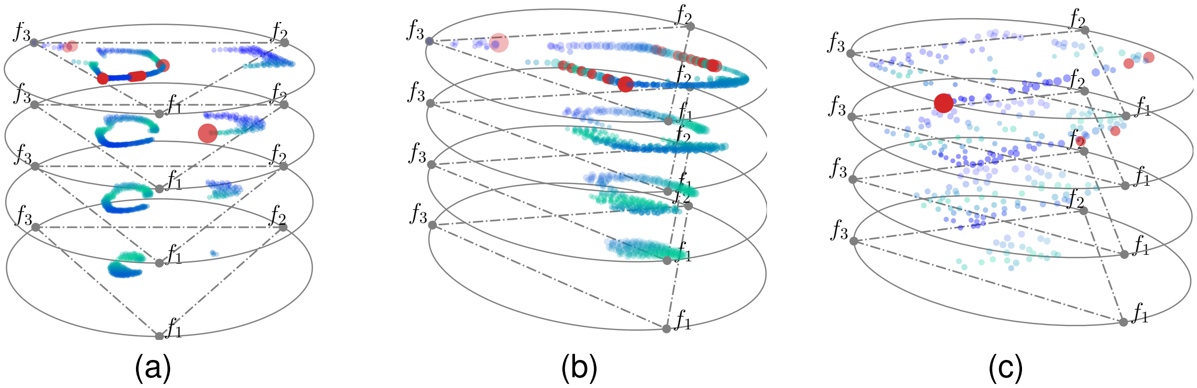


Figure 5.11: (a) PaletteViz plot of the Pareto-optimal front found from vehicle crash-worthiness problem [2]. (b) PaletteViz plot of the cluster A (1,053 points) (ref. Figure 3.9a). (c) PaletteViz plot of the cluster B of the same Pareto-optimal front. (274 points)

In Figure 5.11b, there is no outlier but one point with a large trade-off value. Since the bottom layer contains points far from the boundary, they can be considered as preferred candidate solutions in terms of robust design. In Figure 5.11c, one point with a large trade-off value is observed and can be considered as a preferred choice.

In this case, the points are colored from blue to green based on smallest to largest Euclidean distance from the centroid of each cluster. It is clear that some boundary points (top layer) are not far from the centroid and some points are, however, in the core of cluster A. Most points are closer to the centroid, but interestingly there are some points which are away from the centroid as well. Cluster B has most of its points closer to the centroid, as shown by the green colored points, and only a few points on the boundary are away from the centroid of cluster B. Interestingly there are one or two points having a large trade-off in either cluster. Such a functional analysis of points through our proposed visualization technique demonstrates its usefulness in real-world problems for filtering only a few handfuls of points from an entire set (1,327 points, in this case) of non-dominated solutions.

5.4.2 10-Dimensional General Aviation Aircraft (GAA) Design

Finally, we apply our proposed palette visualization technique on an engineering design problem having ten objectives and 16 constraints [44]. Figure 5.12 clearly shows that there

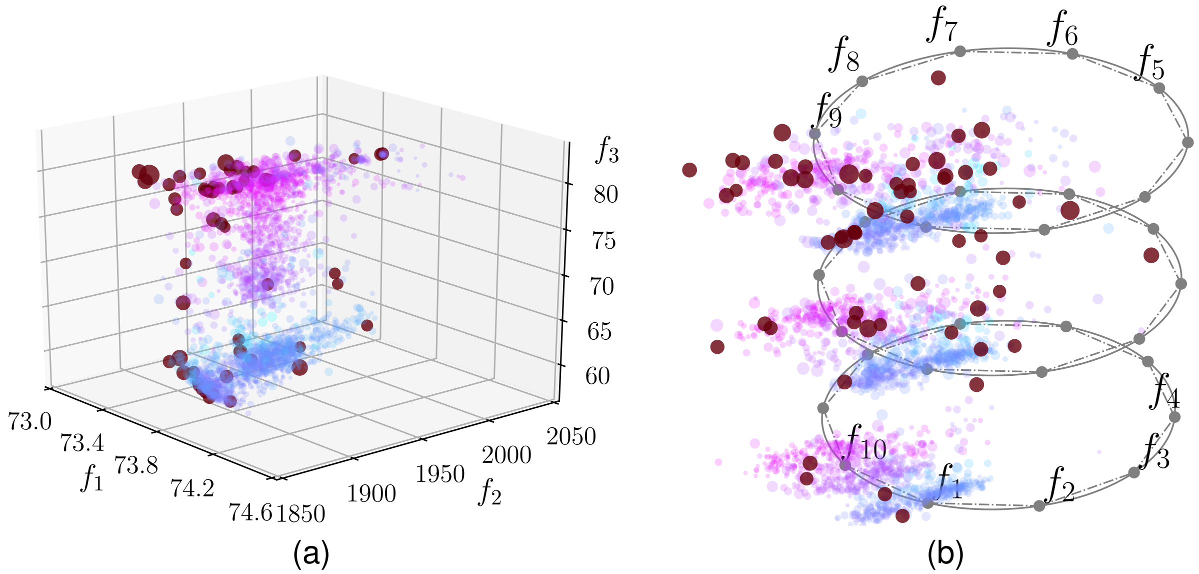


Figure 5.12: (a) Ten-dimensional non-dominated data points for the GAA design problem plotted in the first three-objective space. There are 3,112 points in this data-set. (b) PaletteViz plot for the 10-dimensional GAA design problem.

are two disjoint clusters in the original ten-dimensional Pareto set, which is visible if we plot the data points using only the first three objectives. In these plots, points are colored based on their proximity to constraint surfaces. Interestingly, one of the clusters (with mostly pink points) is close to certain constraints and another (with mostly cyan points) is far from the constraint boundaries. The corresponding palette visualization is presented in Figure 5.12b.

Each cluster has a number of dark-red points indicating a large trade-off that these points possess. The large trade-off can be an artifact of relatively fewer points for a ten-dimensional front representation. Finding more points near each of these large trade-off points using reference-point-based EMO methods [5] may be a way to estimate the real trade-off of these points. However, even with a small density of points, the DM can start analyzing the large number of trade-off points in one of the two clusters, depending on whether the DM prefers solutions close to constraint surfaces or not.

There is another aspect of this problem worth pointing out here: despite the points being normalized before making the PaletteViz plot, the density of the data-set seems to be concentrated towards relatively larger values of f_5–f_7 and on the smaller sides of f_1–f_3 and f_9–f_{10}.

5.5 Comparative Analysis

In order to correctly assess the efficacy of PaletteViz , we need to compare it with other methods. In the preceding sections, we have seen a number of examples where PaletteViz has been compared to other commonly used visualization techniques like PCP, Radviz and Heatmap. However, instead of visual comparison, in this section we will conduct a functional comparison of different visualization techniques. The evaluation criteria that we have used to address the following questions:

- **Number of Objectives:** How many objectives (i.e. dimension) can be visualized by a given method.
- **Trade-off Visualization:** Can a particular visualization method represent trade-off solutions in a useful manner?
- **Topological Information:** Can a visualization method capture topological properties of the underlying space of $\mathcal{C}$?
- **Algorithm Convergence:** Can a visualization method provide information about algorithmic convergence, while a visualization is usually used for identifying the Pareto-optimal front for decision-making purposes.
- **Bijective Mapping:** Since our concern is high-dimensional data visualization and all visualization techniques employ one or another form of mapping of high-dimensional vectors to a lower-dimensional space, can a visualization algorithm produce a bijective mapping[1]?
- **Neighborhood Relationship:** Can a visualization technique maintain the neighborhood relationship among the data points?

[1] In fact, total bijective mapping is not theoretically possible; therefore we would like to ask to what extent a particular visualization method can produce bijective mapping.

Visualization Methods	Evaluation Criteria						
	$m > 4$ Objectives	Trade-off Visualization	Provides Topological Information	Algorithm Convergence	Provides Bijective Mapping	Maintains Neighborhood Relationship	Generates Single Plot
PCP [17]	Yes	No	No	Yes	No	No	Yes
Radviz [20]	Yes	Somewhat	No	No	No	Somewhat	Yes
Star Coordinate [20]	Yes	Somewhat	No	No	No	Somewhat	Yes
Heatmap [19]	Yes	No	No	Yes	No	No	Yes
Scatter Plot Matrix	Yes	Yes	Yes	Yes	Yes	Yes	No
t-SNE [23]	Yes	Yes	No	No	No	Somewhat	Yes
PaletteViz	Yes	Yes	Somewhat	No	Somewhat	Somewhat	Yes
Prosection Method [13]	No	Yes	No	Yes	No	Somewhat	Yes

Table 5.1: Comparison chart for different popular MCDM visualization methods.

- **Number of Plots:** How many plots must be generated to achieve the visualization goal for a given method?

A summary of comparisons is presented in Table 5.1. Here we can see that there are varying degrees of capabilities in each of the visualization methods. It's not a surprise since the comparison of different visualizations is not a straightforward undertaking, mainly because visualization methods are developed with a specific use case in mind. However, in some aspects, PaletteViz seems to be quite useful when it comes to versatility. Interestingly, the scatter plot matrix seems to be the most effective method. But unfortunately, it produces $\binom{m}{2}$ number of plots when the data points are m-dimensional – which produces a substantial cognitive burden. On the other hand, the prosection method looks quite promising in terms of its capability, however this method can't handle more than four objectives.

5.6 Chapter Summary

In this chapter, we have presented a number of examples where PaletteViz has been applied to visualize Pareto fronts of different benchmark Multi-objective Optimization Problems (MOPs). We have shown that PaletteViz can capture some important visual cues so that a DM can make effective data analytics to filter out a few Pareto-optimal points from a set of thousands of data points. With the aid of PaletteViz, the decision-making process can now incorporate topological information about the underlying space of the high-dimensional Pareto-optimal front. In the next chapter, we will discuss how PaletteViz can be applied to a number of real-world MCDM applications and how visual analytics can help with canonical interactive MCDM approaches like NIMBUS [15] and *Pareto-race* [16] while choosing a single preferred solution.

CHAPTER 6

APPLICATIONS IN MULTI-CRITERIA DECISION MAKING (MCDM) AND DATA ANALYTICS

For the past couple of decades, most of the literature in Evolutionary Multi-criterion Optimization (EMO) is mainly concentrated in designing and evaluating algorithms for their ability to converge and find an entire distribution of non-dominated solutions across the efficient front of a multi- or a many-objective optimization problem. However, finding a set of well-distributed Pareto-optimal solutions is just *a part* of the many-objective problem solving; a successful application of an EMO algorithm ends by suitably choosing a single preferred solution from the final non-dominated. However, this final step involves decision-makers in the practicing domain and their preference information. Interestingly, the field of multi-criterion decision-making (MCDM) aptly focuses on the decision-making aspects from preference information and has proposed a plethora of different approaches for arriving at a single best (or pragmatic) solution. The MCDM literature proposes three general ways of integrating many-objective optimization with decision-making:

1. ***A priori* approach**: In this case, trade-off decision-making is conducted first and then multiple objectives are converted into a single-objective problem, which can then be solved using a single-objective optimization algorithm [8]. This approach entirely depends on the scalarization method used and can be argued to be ineffective, given the fact that such an approach does not consider adequate information *a priori* to provide the informed preference information useful for scalarizing the objectives.

2. ***A posteriori* approach**: In this scenario, the optimization is performed first to find a set of Pareto-optimal solutions[1], followed by a decision-making approach to choose a single solution [45]. This allows more informed decision-making, as the knowledge of a number of solutions (the range of objective values, their variation across the set, etc.)

[1] or non-dominated solutions.

is available before any preference information is expected. However, for many-objective optimization problems, the *a posteriori* approach requires a large number of Pareto-optimal solutions to be found and processed, making the approach less applicable in practice for decision-making purposes.

3. **Interactive approach**: In this case, decision-making and optimization are performed in an integrated manner, so that preference information can be integrated and utilized continually on already solved Pareto-optimal solutions. The search is thus focused more towards preferred solutions, rather than focusing on finding the entire Pareto-optimal set [46, 47].

In this section, we show how PaletteViz can aid in the MCDM scenarios discussed above. We call this proposed approach *Visual MCDM and Analysis (VMCDA)*. The idea is to follow one of the three approaches described above while taking help from the information provided by the PaletteViz (or any other) visualization method. From the practical point of view, VMCDA falls in the third category.

6.1 Existing EMO-MCDM Approaches

We confine our formulation only to *a posteriori* and interactive approaches, as an Evolutionary Mult-objective Optimization (EMO) method is usually not required for solving a scalarized single-objective optimization problem. Moreover, the latter approach produces a single final solution. A few *a posteriori* EMO-MCDM approaches can also be found in the literature, but instead of finding a single solution, they seek to find a biased set of Pareto-optimal solutions near the preferred part of the Pareto-optimal front. In the reference-point-based NSGA-II (R-NSGA-II, [48]), only points *close* to the supplied reference or aspiration points were found. In [49], the authors used a reference-direction-based MCDM approach and found only the projected Pareto-optimal points that are closest to a supplied reference direction in the objective space, denoting a trade-off line. In another study [50], the authors implemented a so-called *light-beam search* approach from the MCDM literature with multiple simultaneous light-beams to discover different preferred parts of the Pareto-optimal front. While not many studies have been devoted to interactive approaches, the preference-information-based EMO approach [46] and Branke's approach [47] are worth mentioning.

Besides generic scalarization methods, such as weighted-sum, epsilon-constraint, normal constraint, achievement scalarization function (ASF), and cone domination approaches [8], there exist specific step-by-step MCDM procedures for combining many-objective optimization and decision-making together. These procedures suggest different scalarization methods at every iteration, depending on the progress and properties of the current Pareto-optimal solution. The scalarized problem is then solved using a single-objective optimization algorithm and the new solution is analyzed by the decision-makers to propose a new scalarized problem. Often, these methods start with an arbitrary Pareto-optimal solution as the initial point (or with a solution obtained by the weighted-sum approach with equal weights). We demonstrate our case with two specific MCDM procedures here; the first is known as the *NIMBUS* method [15] and the next is the canonical *Pareto-race* method.

6.2 PalleteViz-NIMBUS Method

The NIMBUS method [15] starts with a set of Pareto-optimal solutions (can be obtained by solving with a weighted-sum approach with equal weight for each objective, for example) or a user-supplied solution $\mathbf{x}^{(0)}$. Then, new and improved preferred solutions ($\mathbf{x}^{(t)}$, $t = 1, 2, \ldots$) are found by solving the following problem (slightly modified from [15]) at every iteration t:

$$\begin{aligned}
\text{Minimize} \quad & \max\left[\max_{i\in I^{<}} \frac{f_i(\mathbf{x}) - z_i^*}{z_i^{\text{nad}} - z_i^{**}},\ \max_{j\in I^{\leq}} \frac{f_j(\mathbf{x}) - \hat{z}_j}{z_j^{\text{nad}} - z_j^{**}}\right] + \rho \sum_{i=1}^{M} \frac{f_i(\mathbf{x})}{z_i^{\text{nad}} - z_i^{**}}, \\
\text{Subject to} \quad & f_i(\mathbf{x}) \leq f_i(\mathbf{x}^{(t)}), \quad \text{for } i \in I^{<} \cup I^{\leq} \cup I^{=}, \\
& f_i(\mathbf{x}) \geq \hat{z}_i, \quad \text{for } i \in I^{\leq}, \\
& f_i(\mathbf{x}) \leq \epsilon_i, \quad \text{for } i \in I^{\geq}, \\
& \mathbf{x} \in \mathcal{S},
\end{aligned} \tag{6.1}$$

where z_i^*, z_i^{**} and z_i^{nad} are the ideal, utopia and nadir point value of the i-th objective, respectively, $\mathcal{S}$ is the feasible variable space, and ρ is a small number (usually, 0.001). Different indices and ϵ_i values are used to direct the search process and they are described as follows:

- The index set $i^<$ indicates the objectives (i) whose values are desired to be improved (that is, decreased for minimization objectives) from the current level ($f_i(\mathbf{x}^{(t)})$).
- The index set $i^\leq$ indicates the objectives (i) whose values are desired to be improved till some desired *aspiration* level ($\hat{z}_i$), that is, $f_i(\mathbf{x}) > \hat{z}_i$.
- The index set $i^=$ indicates the objectives (i) whose values are acceptable in the current solution $\mathbf{x}^{(t)}$.
- The index set $i^\geq$ indicates the objectives (i) whose values may be impaired (i.e., increased) till some *upper* bound (ϵ_i), that is, $f_i(\mathbf{x}) < \epsilon_i$.
- The index set $i^{<>}$ indicates the objectives (i) whose values are temporarily allowed to change freely.

We integrate the above iterative NIMBUS method with our PaletteViz visualization approach in the following step-by-step procedure. Notice that instead of optimizing problem 6.1 in each iteration, we choose the best solution from the EMO-obtained non-dominated set by solving problem (6.1).

Step 1: Apply an EMO algorithm and present all n non-dominated solutions ($\mathcal{F} = \cup_{k=1}^{n}\mathbf{x}^{(k)}$) (the non-dominated set) in a PaletteViz plot for decision-making using the NIMBUS procedure. Identify ideal $\mathbf{z}^*$ and nadir point $\mathbf{z}^{\mathrm{nad}}$ by choosing the minimum and maximum objective values from the non-dominated front $\mathcal{F}$, as follows:$z_i^* = \min_{\mathbf{x}\in\mathcal{F}} f_i(\mathbf{x})$, and $z_i^{\mathrm{nad}} = \max_{\mathbf{x}\in\mathcal{F}} f_i(\mathbf{x})$. Then, compute utopia vector as $z_i^{**} = z_i^* - \delta_i$, where δ_i is a small positive number (e.g. 0.001). Make a PaletteViz plot with the non-dominated set and investigate for any interesting points to include in an archive A.

Step 2: Choose a specific starting non-dominated point ($\mathbf{x}^{(0)}$ and its objective vector $\mathbf{f}^{(0)}$, we call this objective vector $\mathbf{f}_p^{(0)}$), either by visual decision-making (using PaletteViz), or by using a standard scalarizing function method with an equal importance to each objective. Set iteration counter $t = 0$. If the Decision Maker (DM) likes the solution, then it is included in A.

Step 3: Ask the decision-maker (DM) to classify all objective functions at $\mathbf{x}^{(0)}$ and specify aspiration levels ($\hat{z}_i$) and upper bounds (ϵ_i) for respective objective functions in the sets $i^{\leq}$ and $i^{\geq}$, respectively.

Step 4: Select the best solution $\mathbf{x}^{(t+1)}$ from the non-dominated set by restricting the min and max operations in problem (6.1) within the non-dominated set $\mathbf{x} \in \mathcal{F}$.

Step 5: Ask the decision-maker to select the maximum number of alternative solutions (e.g. $1 \leq K \leq 4$) to be chosen uniformly along the line joining the two objective vectors $\mathbf{f}(\mathbf{x}^{(t)})$ and $\mathbf{f}(\mathbf{x}^{(t+1)})$ in the M-dimensional space and finding nearest points in the PaletteViz plot. Present the solutions found to the DM.

Step 6: If the decision-maker wants to save one or more of the new solutions to the archive A, include it/them in A for further analysis.

Step 7: Ask the decision-maker to choose the most preferred solution among the new and/or the intermediate solutions or the solutions in A. Denote it as the current solution $\mathbf{x}^{(t+1)}$ with objective vector $\mathbf{f}_p^{(t+1)}$. If the decision maker wants to continue, set $t \leftarrow t+1$ and go to Step 3. Otherwise, report the archive A as the chosen set of preferred solutions.

6.2.1 Finding Intermediate Solutions in Step 5

Consider Figure 6.1, in which a part of the non-dominated set in the objective space is shown in circles. Points $a = \mathbf{f}(\mathbf{x}^{(t)})$ and $b = \mathbf{f}(\mathbf{x}^{(t+1)})$ are also shown. For three ($K = 3$) uniformly distributed intermediate points along the line $\overline{ab}$, the respective non-dominated points are

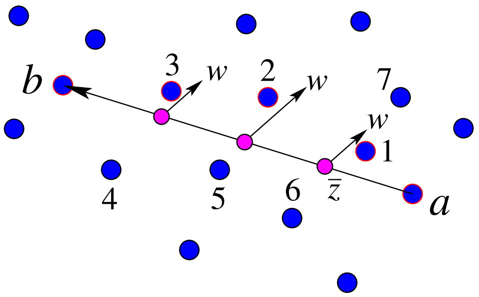


Figure 6.1: Procedure for choosing intermediate points.

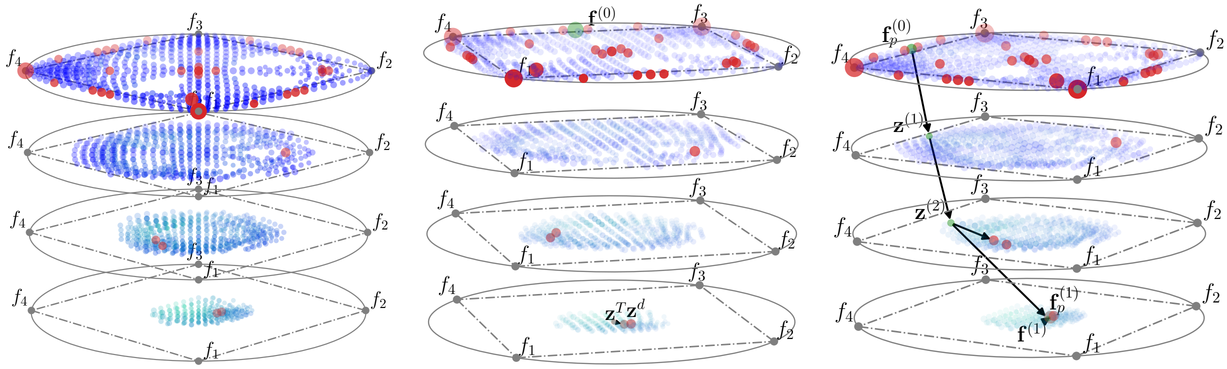
identified by applying the minimum achievement scalarizing function (ASF) [8] value with equal weights. The chosen intermediate point $\bar{\mathbf{z}}$ is as follows:

$$\mathbf{x}_{\bar{\mathbf{z}}} = \text{argmin}_{\mathbf{x}\in\mathcal{F}}\text{max}_{i=1}^{M}\left(\frac{f_i(\mathbf{x}) - \bar{z}_i}{z_i^{\text{nad}} - z_i^*}\right). \tag{6.2}$$

In the Figure 6.1, the chosen objective points are marked as 1, 2 and 3 for the equal weight vector shown. Alternatively, the normalized Euclidean distance can also be used to choose the nearest non-dominated points for the intermediate points. These alternative points are then visualized on the PaletteViz plot to enable a better understanding of their properties.

6.3 Example PaletteViz-NIMBUS Iterations

The proposed *a posteriori* approach makes a computationally faster and more visually intuitive decision-making process than the original NIMBUS procedure. Moreover, since the preferred solutions are shown through an effective visualization method which can readily provide information about the functionality of the points in their original high-dimensional space, such as their closeness to a constraint boundary, closeness to the core of the set, their trade-off value compared to neighboring solutions, their robustness against uncertainties of the problem, and other properties of relevance to the DMs, we argue that the choice of a few preferred solutions in Step 5 will be relatively easier than the original approach proposed with a PCP or other visualization methods. We illustrate the working of the PaletteViz-NIMBUS method in this section. We consider two test problems and one engineering design problem.



(a) All Pareto-optimal points. (b) Iteration 0. (c) Iteration 1.

Figure 6.2: PaletteViz-NIMBUS on DEBMD1K problem.

6.3.1 Four-objective DEBMD1K (*Knee*) Problem

First, we choose a 4-objective problem [1] having knee-like (i.e. high-tradeoff) points at the core of the Pareto front. Figure 6.2a shows 2,000 non-dominated points obtained using NSGA-III [51] with standard parameter settings. Two red points on the bottom layer (representing the core of the front) are clearly visible. The ideal and nadir points of the non-dominated points are as follows:

$$\mathbf{z}^* = [0, 0, 0, 0], \quad \mathbf{z}^{\text{nad}} = [9.5, 9.5, 9.5, 9.5].$$

Let us say, the DM is interested in finding a high trade-off solution near the core with equal importance to all four objectives. A search for such a point with $\mathbf{w}^d = [1, 1, 1, 1]/\sqrt{4}$ with ASF minimization approach with ideal point and equal weight vector finds the following target vector from the non-dominated set: $\mathbf{z}^d = [1.5068, 1.5068, 1.8081, 1.8081]$. To find a high trade-off point near this ASF solution, we borrow the following definition of a trade-off (sacrifice to gain in the neighborhood) of a point $\mathbf{x}$ [25]:

$$\mathcal{T}(\mathbf{x}) = -\frac{\min_{\mathbf{z}\in B(\mathbf{x})} \min(f(\mathbf{x}) - f(\mathbf{z}), 0)}{\max_{\mathbf{z}\in B(\mathbf{x})} \max(f(\mathbf{x}) - f(\mathbf{z}), 0)}, \tag{6.3}$$

which computes the maximum loss versus maximum gain for a non-dominated point ($\mathbf{x}$) in moving to any neighboring point within set $B(\mathbf{x})$. Here, we define the nearest P (=20) points as the neighborhood size. The higher the trade-off value, the more preferred is the solution.

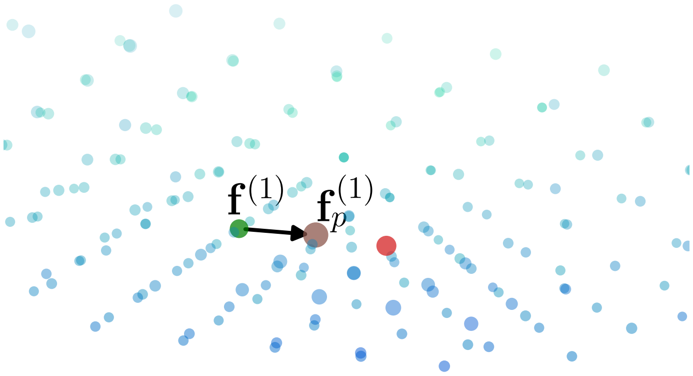


Figure 6.3: $\mathbf{f}_p^{(1)}$ has a higher trade-off than $\mathbf{f}^{(1)}$ (a part of bottom-most PaletteViz plot).

With this definition, we are able to find the following target solution ($\mathbf{z}^T$) which has an improved trade-off of 0.3241 from 0.0795 for $\mathbf{z}^d$: $\mathbf{z}^T = [1.0470, 1.0470, 1.5705, 2.0940]$. It is one of the high trade-off solutions at the core of the Pareto front, as shown in Figure 6.2b.

To begin the NIMBUS procedure, let us say, we start our PaletteViz-NIMBUS procedure with a solution that has a large weight for the first two objectives ($w_i = 0.5$ for $i = 1, 2$ and $w_i = 0.01$ for $i \geq 3$) resulting: $\mathbf{f}_p^{(0)} = \mathbf{f}^{(0)} = [5.1854, 5.1854, 0, 0]$. Using the principle set out by the DM, in iteration 1, the DM provides the following classification of objectives:

$$I^{<} = \emptyset, \quad I^{\leq} = \{1, 2\}, \quad I^{=} = \emptyset, \quad I^{\geq} = \{3, 4\}, \quad I^{<>} = \emptyset,$$

with $\hat{z}_1 = \hat{z}_2 = 1.0$ and $\epsilon_3 = \epsilon_4 = 2.0$. Using Step 4, the obtained solution from non-dominated set is

$$\mathbf{f}^{(1)} = [1.1140, 1.1140, 1.9496, 1.9496],$$

with a trade-off value $\mathcal{T} = 0.0906$. However, the PaletteViz plot in Figure 6.3 helps to discover that there is a better trade-off solution ($\mathcal{T} = 0.3241$) close to it. The DM chooses this solution (we refer as $\mathbf{f}_p^{(1)}$): $\mathbf{f}^{(1)} = [1.0470, 1.0470, 1.5705, 2.0940]$.

Thus, the DM chooses this higher trade-off solution $\mathbf{f}^{(1)}$ and continue with the NIMBUS method. In Step 5, the DM plans to find three ($K = 2$) intermediate points between $\mathbf{f}^{(0)}$ and $\mathbf{f}^{(1)}$, marked as $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$. These two points are visualized using PaletteViz in Figure 6.2c. The DM observes that $\mathbf{z}^{(1)}$ has no high trade-off solution (red point) near it, but $\mathbf{z}^{(2)}$ has such a point. Thus, solution $\mathbf{z}^{(1)}$ is retained and $\mathbf{z}^{(2)}$ is replaced with its nearest high trade-off solution obtained from the PaletteViz plot:

$$\begin{aligned}
\bar{\mathbf{z}}^{(1)} &= [4.9216, 4.9216, 0.4922, 0.4922], \quad \mathcal{T}(\bar{\mathbf{z}}^{(1)}) = 0.1725, \\
\bar{\mathbf{z}}^{(2)} &= [2.384, 2.384, 3.9734, 0], \quad \mathcal{T}(\bar{\mathbf{z}}^{(2)}) = 0.3081 \text{ from } \mathcal{T}(\mathbf{z}^{(2)}) = 0.2196.
\end{aligned}$$

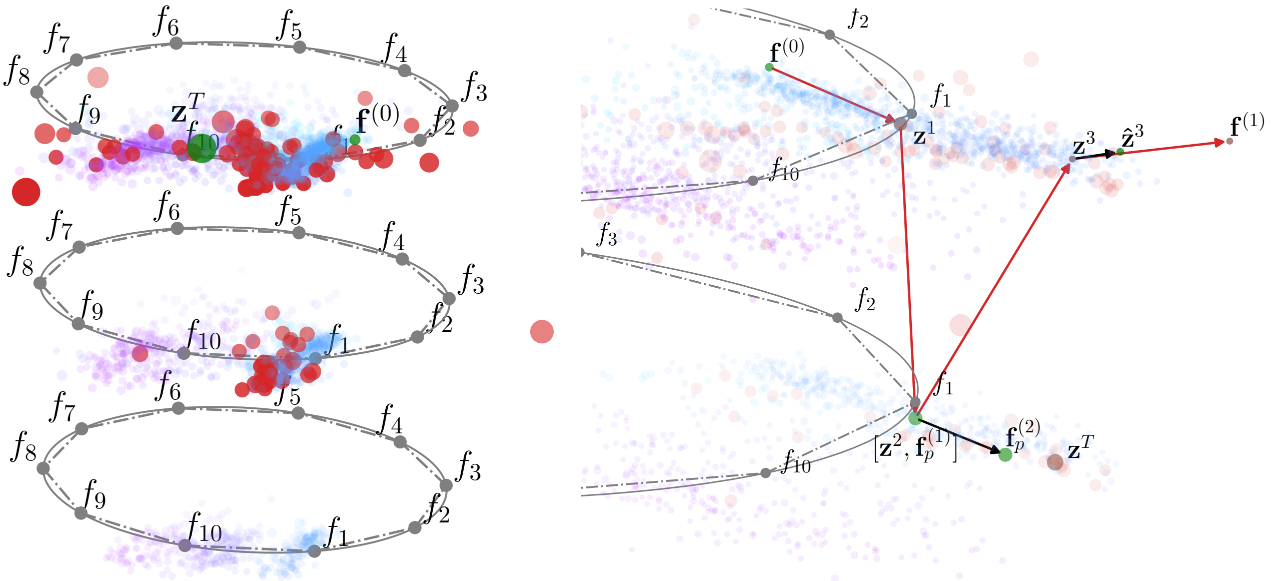
Since not all solutions are very interesting, instead of storing points in the archive in Step 6, we move to Step 7 and choose the best solution among $\mathbf{f}^{(1)}$ and the above two solutions based on the closest point to $\mathbf{z}^d$: $\mathbf{f}_p^{(1)} = [1.0470, 1.0470, 1.5705, 2.0940], \quad \mathcal{T}(\mathbf{f}_p^{(1)}) = 0.3241$. Note that this solution is identical to $\mathbf{z}^T$ with a trade-off of 0.3241. Note also how the above PaletteViz allows the DM to pick several large trade-off solutions in the neighborhood of NIMBUS-obtained solutions and also they follow DM's preferences.

6.3.2 10-Objective General Aviation Aircraft (GAA) Design Problem

Finally, we choose an engineering design problem (GAA) having 10 objectives and 16 constraints [44]. Figure 6.4a shows a PaletteViz plot of 3,000 non-dominated points obtained using NSGA-III procedure (Step 1). It is interesting to note that the boundary layer has many high trade-off points and the points in the core include no high trade-off points. Moreover, the blue color indicates that the points are away from constraints and pink points are close to at least one of the constraint boundaries. The ideal and nadir points of the non-dominated set are:

$$\begin{aligned}
\mathbf{z}^* &= [73.25, 1882.74, 57.70, 1.86, 376.36, 42206.65, -2547.43, -16.79, -202.90, 0.00], \\
\mathbf{z}^{\text{nad}} &= [74.44, 2038.54, 82.54, 2.00, 500.00, 44945.04, -1999.96, -14.46, -186.15, 3.70].
\end{aligned}$$

In Step 2, a close investigation of PaletteViz plot clearly indicates that there are three outlier points with a large trade-off $\mathcal{T}$ value (shown in the following), marked as $A^{(1)}$, $A^{(2)}$, and



(a) All non-dominated points.

(b) Iterations 1 and 2.

Figure 6.4: PaletteViz-NIMBUS method illustrated, at iteration 1, the exploration path moves as follows: $\mathbf{f}^{(0)} \rightarrow \mathbf{z}^{(1)} \rightarrow \mathbf{z}^{(2)} \rightarrow \mathbf{z}^{(3)} \rightarrow \mathbf{f}^{(1)}$. Chosen solution is $\mathbf{f}_p^{(1)} = \mathbf{z}^{(2)}$. $\mathbf{f}_p^{(2)}$ is the final solution of Iteration 2 obtained from $\mathbf{f}_p^{(1)}$. All iterations proceed towards targeted solution $\mathbf{z}^T$.

$A^{(3)}$) in Figure 6.4b:

$$A = \begin{pmatrix} [73.34, 1889.03, 80.00, 2.00, 450.32, 42691.33, -2024.06, -16.79, -202.62, 0.65], & \mathcal{T} = 0.95 \\ [73.79, 2010.62, 80.63, 2.00, 406.64, 44296.90, -2331.33, -15.86, -187.13, 2.32], & \mathcal{T} = 0.58 \\ [73.75, 1901.27, 59.51, 1.87, 432.80, 42786.86, -2000.00, -14.96, -198.91, 0.92], & \mathcal{T} = 0.71 \end{pmatrix}$$

The respective pseudo-weight vectors [52] for these points are computed below:

$$\widetilde{\mathbf{w}} = \begin{pmatrix} [\mathbf{0.153}, \mathbf{0.157}, 0.017, 0.004, 0.066, \mathbf{0.135}, 0.007, \mathbf{0.164}, \mathbf{0.161}, \mathbf{0.135}], & d_{\mathbf{w}} = 0.143 \\ [\mathbf{0.158}, 0.052, 0.022, 0.000, \mathbf{0.220}, 0.069, \mathbf{0.176}, \mathbf{0.175}, \ 0.017, \ 0.109], & d_{\mathbf{w}} = 0.306 \\ [0.092, \mathbf{0.139}, \mathbf{0.146}, \mathbf{0.142}, 0.085, 0.124, 0.000, \ 0.034, \ 0.120, \ 0.118], & d_{\mathbf{w}} = 0.157 \end{pmatrix}$$

.

The first solution $A^{(1)}$) emphasizes six of the 10 objectives (according to a pseudo-weight half of the standard deviation above their mean), while second solution emphasizes four objectives and third solution emphasizes only three objectives. Importantly, since there are no solutions close to them, it means that no other solution in the non-dominated set comes close to emphasizing the objectives in the above way, thereby making these solutions interesting for the DM to consider further. These three points are added to the archive A for further analysis.

Let us assume that the DM's goal for this problem is to reach a solution which emphasizes objectives f_1 to f_{10} in a linearly increasing manner. The respective non-dominated point in the set is

$$\mathbf{z}^T = [73.48, 1894.34, 60.28, 1.99, 450.00, 42713.74, -2088.43, \; -16.02, -201.02, 0.71],$$

with a trade-off $\mathcal{T} = 0.5638$. Although the DM would like to get close to this solution, the DM wanted to use PaletteViz aid and the flexibility of NIMBUS method to arrive there.

We start the NIMBUS method with an initial neutral point, obtained by minimizing ASF with ideal point and an equal weight vector:

$$\mathbf{f}_p^{(0)}{=}\,\mathbf{f}^{(0)}{=}\,[73.82, 1895.35, 59.33, 1.97, 450.00, 42565.30, -2145.27, -15.33, -197.96, 1.08].$$

This solution is marked in Figure 6.4b and has a pseudo-weight vector:

$$\widetilde{\mathbf{w}}^T = [0.126, 0.143, 0.139, 0.011, 0.063, 0.126, 0.025, 0.104, 0.138, 0.125].$$

The DM is interested in finding a solution whose pseudo-weight vector comes closest to this vector. The solution $\mathbf{f}^{(0)}$ has a distance $d_{\mathbf{w}} = \|\widetilde{\mathbf{w}}_{\mathbf{f}_p^{(0)}}, \widetilde{\mathbf{w}}^T\|_2 = 0.0752$. Notice that the three archive solutions have very different pseudo-weight vector than $\widetilde{\mathbf{w}}^T$. With the linear importance in mind, DM wants to improve f_{10} by sacrificing f_1 in the first iteration, as follows:

$$I^{<} = \{10\}, \quad I^{\leq} = \emptyset, \quad I^{=} = \emptyset, \quad I^{\geq} = \{1\}, \quad I^{<>} = \{2\text{--}9\}, \quad \epsilon_1 = 74.$$

The respective solution of problem (6.1), found in Step 4, is

$$\mathbf{f}^{(1)}{=}\,[73.50, 1945.10, 62.37, 1.99, 450.00, 43438.32, -2000.30, -15.29, -195.53, 0.00],$$

with $\mathcal{T} = 0.0834$. In Step 5, $K = 3$ intermediate points $\mathbf{z}^{(1)}$ to $\mathbf{z}^{(3)}$) are chosen and are presented in the PaletteViz plot in Figure 6.4b. Notice that while most points lie on the boundary layer, point $\mathbf{z}^{(2)}$ lies in the middle layer. By analyzing the plot, it is observed that following high trade-off points are close to these intermediate points:

$$\hat{\mathbf{z}} = \begin{pmatrix} [73.71, 1883.91, 59.11, 1.98, 456.38, 42550.94, -2145.49, -15.67, -200.52, 0.99], & \mathcal{T} = 0.36 \\ [73.77, 1885.33, 58.83, 1.98, 450.15, 42563.51, -2100.18, -15.81, -200.58, 0.75], & \mathcal{T} = 0.40 \\ [73.71, 1922.04, 59.07, 1.98, 449.96, 43237.92, -2000.00, -15.80, -199.65, 0.28], & \mathcal{T} = 0.13 \end{pmatrix}.$$

Let us say that the DM chooses the solution $\mathbf{z}^{(2)}$:

$$\mathbf{f}_p^{(1)} = [73.77, 1885.33, 58.83, 1.98, 450.15, 42563.51, -2100.18, -15.81, -200.58, 0.75],$$

with $\mathcal{T} = 0.3998$. Note that for this solution

$$\|\widetilde{\mathbf{w}}_{\mathbf{f}_p^{(1)}}, \widetilde{\mathbf{w}}^T\|_2 = 0.0443,$$

indicating that the DM has found a solution with a more similar pseudo-weight vector to target than $\mathbf{f}_p^{(0)}$. In the second iteration, a new classification is provided based on the above solution:

$$I^{<} = \emptyset,\ I^{\leq} = \{10\},\ I^{=} = \emptyset,\ I^{\geq} = \{6\},\ I^{<>} = \{1-5, 7-9\},\ \hat{z}_{10} = 0.7,\ \epsilon_6 = 42715.$$

Following the PaletteViz-NIMBUS method, we obtain a new solution:

$$\mathbf{f}_p^{(2)} = [73.74, 1886.47, 58.30, 2.00, 450.00, 42674.91, -2055.81, -16.05, -201.58, 0.66],$$

with $\mathcal{T} = 0.3849$. This solution has a smaller $\|\widetilde{\mathbf{w}}_{\mathbf{f}_p^{(2)}}, \widetilde{\mathbf{w}}^T\|_2 = 0.0419$ compared to $\mathbf{f}_p^{(1)}$. Figure 6.4b also shows that this solution is closer to the desired target, thereby terminating the decision-making process.

6.4 An Example MCDA using Pareto Race

As a proof of concept of utilizing the proposed PaletteViz method in aiding Multi-criterion Decision-Making and Analysis (MCDA) by decision-makers (DMs), we demonstrate its use in following another interactive MCDM method – the *Pareto-race* [16] method. Pareto-race is an interactive MCDA technique, in which a DM provides an m-dimensional direction vector in the objective space (usually, starting from the *nadir* point and moving towards the *ideal/utopia* point). Points $Z = \{\mathbf{z}^{\text{nad}}, \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}^*\}$ are created on the direction of the vector to simulate the movement along the line. For every intermediate point $\mathbf{z}$ on the line, the points with the best achievement scalarization function (ASF) [4] are selected from the Pareto-optimal front:

$$\mathrm{S}_{\mathbf{z}}(\mathbf{f}(x), \mathbf{w}) = \max_i \{w_i \left(f_i(x) - z_i\right)\} + \rho \sum_i w_i \left(f_i(x) - z_i\right) \tag{6.4}$$

In all of our experiments, we assume the weight vectors are identical, i.e. $\forall i, j : \sum_i w_i = 1, w_i = w_j$. ρ is a very small number (we set to 0.001) to avoid the weakly non-dominated points. A DM can then analyze the sequence of points and increase or decrease their speed of movement or steer in a different direction. The process is expected to lead DMs to reach to a compromise solution. Coupled with our visualization method, now the Pareto-race method can guide the DM through the high-dimensional space in an intuitive manner and help in identifying the location of the chosen trade-off point – whether the point is lying on boundary or core, closer to a constraint boundary, closer to a high trade-off point, on a disconnected region in the Pareto-optimal front, or on a specific/desired part of the front, etc.

6.4.1 A Case for DTLZ8 Problem

In many real-world problems, a Pareto-front can consist of clusters of data points having different dimensions. For example, a Pareto-front may consist of a mix of spherical and planar manifolds, and even can have one of its part made of a manifold that spans a lower dimension, etc. In order to test how the proposed PaletteViz method works[2] on such a scenario, for example, we consider the DTLZ8 problem [43], where the Pareto-front consists of a $(m-1)$-dimensional hyper-plane and a one-dimensional straight line. The corresponding three-dimensional Pareto-optimal front is presented in Fig. 6.5a. The PaletteViz of a four-

[2] In this example we apply star-coordinate plot [21] for the mapping of each depth contour on the 2D planes.

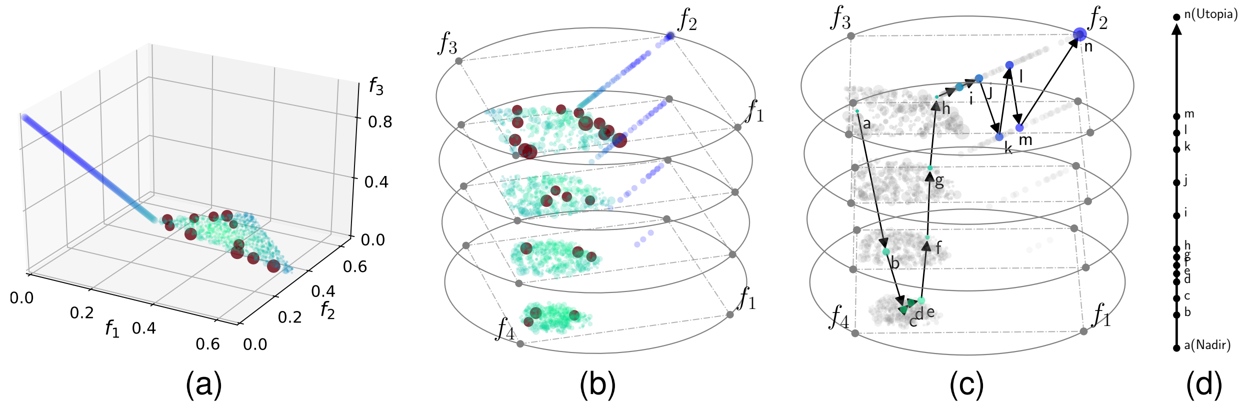


Figure 6.5: (a) A three-objective Pareto-optimal data-set (1,038 points) for the DTLZ8 problem. (b) A PaletteViz plot of the four-objective DTLZ8 Pareto-optimal front (2,105 points). (c) A *Pareto-race* like solution exploration steps. (d) A *traversal scheme* defines how an individual jump was made along a straight line from Nadir to Utopia point.

objective version of the same problem is presented in Fig. 6.5b. The following decision-making information can be gathered from the plot: (i) the front consists of a straight line and a four-dimensional surface, (ii) a few exceptionally high trade-off points exist, most of the high trade-off points lie near the boundary of the four-dimensional front, and (iii) no large trade-off point exists on the straight-line part of the set.

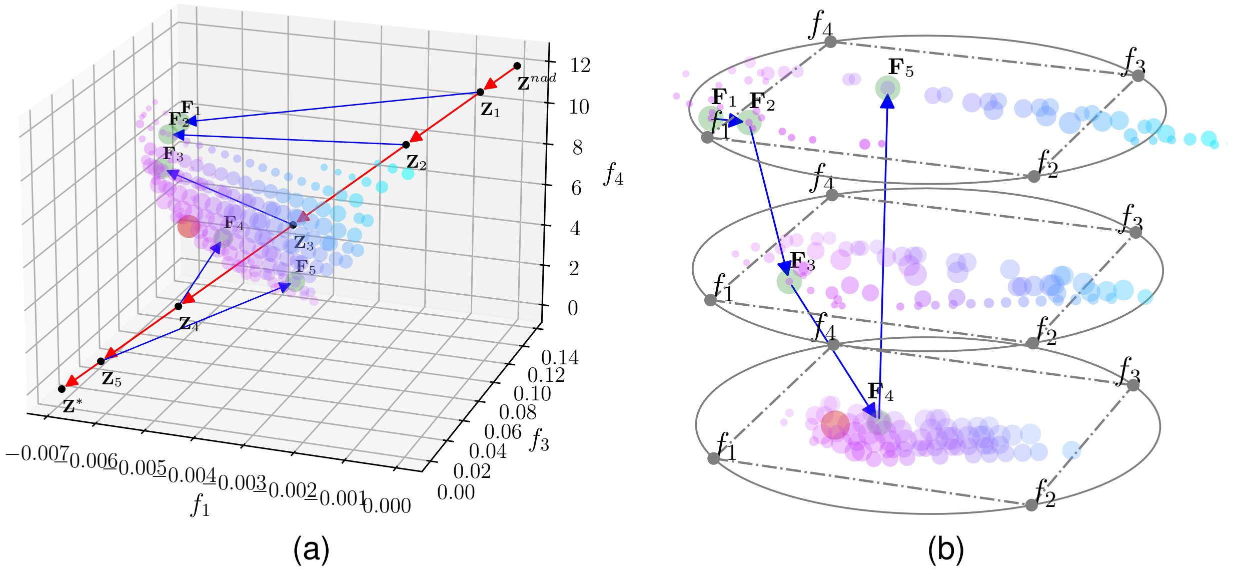
In this MCDA process, the DM starts from the *nadir* point and moves towards the ideal point in the four-dimensional objective space with a uniform step. Points a and b represent the first two points on the adjoining vector and their respective non-dominated points are shown on the PaletteViz plot. It can be seen that point a lies on the top-most layer indicating that it is a boundary point on the four-dimensional trade-off frontier. The point b moves into the innermost layer. Now the DM decides whether that point b is worth analyzing, as there might be a preference for a point closer to the core of the frontier. Therefore, the DM slows down in the same direction and finds c and d. To investigate further, the DM keeps reducing the speed and finds points e and f. However, the respective trade-off points have left the inner layers and are moving towards the boundary layer. The DM notices that a further movement along the direction reveals points with higher trade-off that are on the two outer most layers, before converging to point n which is the respective ASF point for the ideal point. Thus, the entire journey from *nadir* to ideal point provides some useful information in terms of objective selection of preferred points. As in the previous examples, the data points can be marked in a constraint-based coloring approach (pink vs. cyan) so that every ASF point can be visualized for its closeness to constraint boundary, knee point, isolated region, etc. This manner of visual analytics is not possible in the other existing visualization methods.

6.4.2 A Case for Four-Objective Portfolio Optimization Problem

Multi/Many-objective optimization techniques are widely used in financial decision making and project management [53]. In this case, we demonstrate a simple case for a stock portfolio optimization problem.

Let us assume we have stock quotes from n_q number of symbols[3] and their normalized closing prices within a certain date range. The expected return from each stock $\bar{\mu}_r = \{r_1, r_2, \ldots, r_{n_q}\}$ in the portfolio where μ_i is the mean return from asset i. We need to find a vector of weights $\mathbf{w} = \{w_1, w_2, \ldots, w_{n_q}\}$ for asset allocation where w_i $(0 \leq w_i \leq 1)$ denotes the proportion of total investment in stock i in the portfolio, where $\sum_i w_i = 1$. The objective of this problem is to maximize the expected return, which is defined as the weighted average of individual return $p(\bar{\mu}_r, \mathbf{w}) = \sum_i^{nq} \mu_i w_i$. If the variance-covariance matrix of $\bar{\mu}_r$ is $\Sigma_{\bar{\mu}_r}$, then the expected risk of the portfolio is $r(\Sigma_{\bar{\mu}_r}, \mathbf{w}) = \sum_i^{nq} \sum_j^{nq} w_i w_j \sigma_{i,j}$. Instead of the expected risk, we also minimize so-called *Value-at-Risk (VaR)* and *Conditional Value-at-Risk (CVaR)*. More details on VaR and CVaR can be found in [54]. In order to maintain the diversity in the investment, we also add another objective that minimizes the amount of asset allocation that does not go below a certain asset allocation threshold $1/n_q$. Therefore,

[3] In our case, the total number of stocks is 12, i.e. $n_q = 12$.



(a)

(b)

Figure 6.6: (a) A 4-objective Pareto-optimal front (118 points) for the portfolio optimization problem presented in the first three dimensions $\{f_1, f_2, f_3\}$. (b) A PaletteViz plot of the same Pareto-optimal front. A *Pareto-race* solution exploration scheme presented with red arrows along the direction vector $\mathbf{v} = \mathbf{z}^* - \mathbf{z}^{\text{nad}}$ and each discovery of solution is presented with blue arrows.

the overall corresponding multi-objective optimization problem is defined as follows:

$$
\begin{aligned}
&\min f_1(\mathbf{w}) = -p(\bar{\mu}_r, \mathbf{w}) \\
&\min f_2(\mathbf{w}) = \text{VaR}(\alpha, p(\bar{\mu}_r, \mathbf{w}), r(\Sigma_{\bar{\mu}_r}, \mathbf{w})) \\
&\min f_3(\mathbf{w}) = \text{CVaR}(\alpha, p(\bar{\mu}_r, \mathbf{w}), r(\Sigma_{\bar{\mu}_r}, \mathbf{w})) \qquad (6.5) \\
&\min f_4(\mathbf{w}) = \min \sum_i^n \phi(w_i) \text{ where } \phi(w_i) = \begin{cases} 1, & \text{if } w_i < 1/n_q \\ 0, & \text{otherwise} \end{cases} \\
&\qquad \text{s.t. } \sum_i^{n_q} w_i = 1, \quad 0 \le w_i \le 1, \quad n = 12, \quad \alpha = 0.05
\end{aligned}
$$

We solve this problem using the NSGA-III algorithm [51] with 12,341 population members over 2,000 generations. After the run, we have found only 188 feasible Pareto-optimal solutions that are shown in Fig. 6.6a. Interestingly, this front has only one high-tradeoff solution which is denoted with a large red circle in the plot. The scatter plot in Fig. 6.6a is presented with objectives $\{f_1, f_3, f_4\}$ for visual convenience. The Pareto-race traversal is shown in a red arrow starting from the *nadir* point Z^{nad} and ending at ideal point Z^*. Each intermediate point on the line is presented with $\mathbf{z}_i$ (black filled circles) and the corresponding points on the Pareto-front found by solving the equation (6.4) are denoted with $\mathbf{F}_i$ (large transparent green circles). We denote the search direction from the reference points to their corresponding $\mathbf{F}_i$ points with blue arrows. The PaletteViz plot is given in Fig. 6.6b.

In this scenario, the we start from $\mathbf{z}^{\text{nad}}$. After making small step along the direction vector $\mathbf{v} = \mathbf{z}^* - \mathbf{z}^{\text{nad}}$, we arrive at $\mathbf{z}_1$, solve equation (6.4) and discover the point $\mathbf{F}_1$. At this stage, we can sees that $\mathbf{F}_1$ is close to the boundary, so a longer step is made and it leads us to the point $\mathbf{z}_2$. The corresponding ASF point from this position is $\mathbf{F}_2$. As it is still closer to the boundary, we make a bigger jump and find $\mathbf{F}_3$. Since $\mathbf{F}_1$, $\mathbf{F}_2$ and $\mathbf{F}_3$ are situated on a closer neighborhood, next we apply the same procedure to find $\mathbf{F}_4$. We can see that this point is very close to the high-tradeoff point and also farther away from the boundary with comparatively smaller magnitude of constraint violation. This point is saved for further analysis. Next, we make another step forward and discover $\mathbf{F}_5$ (from $\mathbf{z}_5$, by solving (6.4)). Here we see that $\mathbf{F}_5$ is again close to the boundary. Therefore, we settle with point $\mathbf{F}_4$[4]. We can also use the PaletteViz to see where the preferred point is located and based on that information, we can change the direction vector to get a better look. The entire traversal in the PaletteViz plot shows how we move from boundary to the core and then again we move back to the boundary of the Pareto-optimal front. This kind of functionally-decomposed representation of Pareto-optimal front can help the DM to conduct a better a decision-making task through a visual analysis.

6.5 Visualization of $\mathbf{H}_{m-1}$ Homology Group Boundary of An m-dimensional Pareto-optimal Front ($\mathcal{PF}$)

Besides multi-criterion decision-making applications, the PaletteViz can also be used for Topological Data Analysis (TDA) purposes. As we have already discussed somewhat in detail about this topic in Chapter 3, the readers can speculate that PaletteViz method may

[4] Or, we can restart the entire procedure to explore other points in the vicinity of $\mathbf{F}_4$.

be useful in TDA domain as well. Here we demonstrate an example proof of concept for such an application. PaletteViz method can also be used to visualize the boundary points of an $\mathbf{H}_{m-1}$ homology group in an m-dimensional Pareto-optimal front $(\mathcal{PF})^5$. To demonstrate this scenario, we device another multi-objective optimization problem where upon solving, there will be a *hole* on the Pareto-optimal surface. We achieve this scenario by introducing a constraint function in DTLZ2 problem where all the points inside the *hole* are infeasible. The exact formulation of this function is presented in Appendix A.1.6.

The boundary points of homology groups are important in high-dimensional MCDM, since a homology group appears in $\mathcal{C}$ if the Pareto-optimal front has a constraint function that is disconnected or composed of multiple dissimilar dimensional spaces. A more specific example could be identification and visual analysis of points lying on a homology boundary that are extremely close to the constraint boundary as well. In such a case, a DM might want avoid those points since they have higher probability of being less robust.

6.5.1 Computing the Boundary Points of $\mathbf{H}_{m-1}$

The first step of the algorithm is to construct the persistence diagram $\mathsf{Dgm}(\mathcal{C})$ from $\mathcal{C}$, which can be done by solving the Smith-normal form found from the Vietoris-Rips complex of the data points (please refer to Section 2.1). Once this is done, we will also have the r_j, t_{b_j} and t_{d_j} values of the persistence diagram Dgm for all homologies up to $\mathbf{H}_{m-2}$.

To find the chain complex of $\mathbf{H}_{m-2}$, we locate the highest off-diagonal value in the $\mathsf{Dgm}(\mathcal{C})$ for $\mathbf{H}_{m-2}$, which basically will give us the death time for the last $(m-2)$-dimensional hole that has been eliminated. The second-highest off-diagonal value in $\mathsf{Dgm}(\mathcal{C})$ is the birth time for the same complex. Therefore, we collect all the representative cocyles [28] that are generated at the mid-point of highest and the second-highest off-diagonal values in $\mathsf{Dgm}(\mathcal{C})$. We then use that value as threshold to find all the representative cocyles. The cocycle is a collection of 1-simplices that, if removed, would break the cycle in $\mathcal{C}$.

[5] In an m-dimensional Pareto-optimal space, $\mathbf{H}_m$ homology groups will never appear. Please refer to Section 3.3 for a formal proof.

Algorithm 4 Find Boundary Points of $\mathbf{H}_{m-1}$

```
Require: Data points $\mathcal{C}$, persistence diagram $\mathsf{Dgm}(\mathcal{C})$, a distance matrix $\mathbf{M}$ of $\mathcal{C}$, where
    $|\mathbf{M}| = n \times n$ and $n = |\mathcal{C}|$
Find highest off-diagonal value $[t_{b_j}, t_{d_j}, r_j]$ and the second highest off diagonal value
    $[t_{b_{j-1}}, t_{d_{j-1}}, r_{j-1}]$ in $\mathsf{Dgm}(\mathcal{C})$
Find the mid point radius $\epsilon = \frac{1}{2}(r_j + r_{j-1})$
Construct *Vietoris-Rips* complex $\mathcal{R}_\epsilon(\mathcal{C})$ over $\mathbf{M}$
$L \leftarrow \emptyset$
for each $\sigma_i \in \mathcal{R}_\epsilon(\mathcal{C})$ do
    if $L = \emptyset$ then
        $S \leftarrow \emptyset$
        $S \leftarrow S \cup \sigma_i$
        $L \leftarrow L \cup S$
    else
        for each $S \in L$ do
            for each $\sigma_j \in S$ do
                if $\sigma_i \cap \sigma_j \neq \emptyset$ then
                    $S \leftarrow S \cup \sigma_i$
                    break
                end if
            end for
        end for
    end if
end for
$\mathcal{B}_s \leftarrow \emptyset$
for each $S \in L$ do
    Apply algorithm 2 (ref. Section 4.1.1) to find all the boundary points $\mathcal{B}$ of $S$
    $\mathcal{B}_s \leftarrow \mathcal{B}_s \cup \mathcal{B}$
end for
return $\mathcal{B}_s$
```

Now in order to find all the points that bounds all the cocylces, we go through each of them and apply a similar procedure to Algorithm 2 to construct a chain of vertices that corresponds to the boundary of $\mathbf{H}_{m-1}$. A pseudocode for this procedure is presented in Algorithm 4.

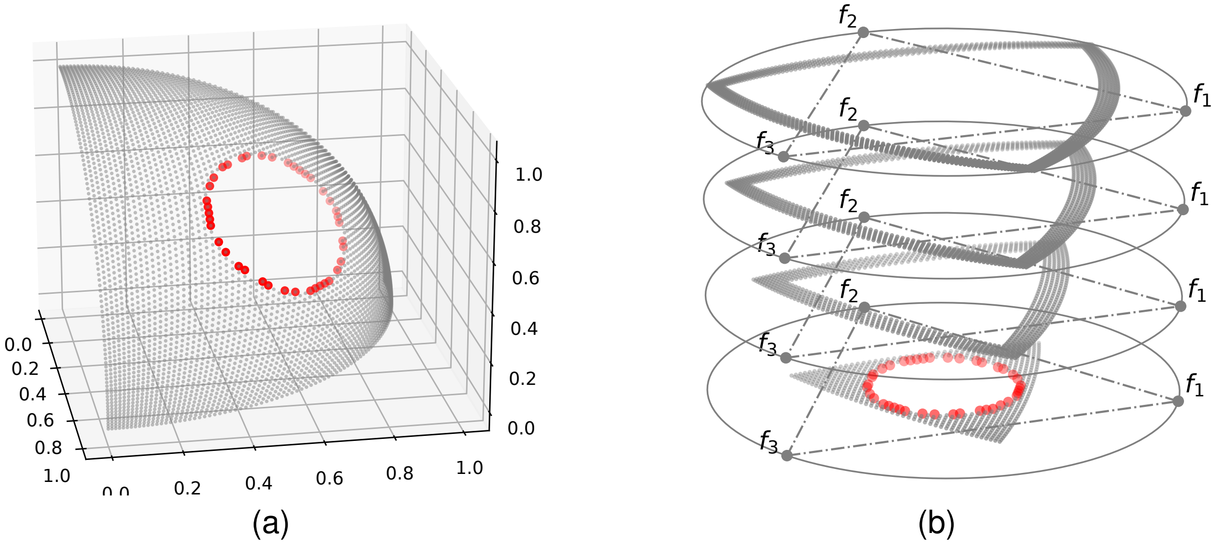


Figure 6.7: An example $\mathbf{H}_{m-1}$ boundary points visualization (a) Scatter plot visualization of a three-dimensional spherical $\mathcal{PF}$ with the $\mathbf{H}_1$ homology group. (b) A PaletteViz visualization of the same $\mathcal{PF}$.

6.5.2 Visualization Results

Once the boundary points are identified, we can visualize them with a conspicuous color coding. In our case, we use red color for the boundary points and keep the rest of the points in gray. Although PaletteViz is not required for an efficient visualization of three-dimensional points, we include this example since it is helpful to visualize the boundary points of an apparent hole in the $\mathcal{PF}$. Figure 6.7a is the scatter plot of the spherical $\mathcal{PF}$. In Figure 6.7b, we can see how the boundary of the $\mathbf{H}_2$ homology group (points in red) appears in the bottom layer of the PalettViz plot. The red points represent the Pareto-optimal solutions that are very close to the constraint function.

Figure 6.8a is the $\mathcal{PF}$ in four objectives. Here we can see that points surrounding the $\mathbf{H}_2$ homology group are highlighted in red. Since $\mathcal{C}$ is in four-dimensional space, the hole created by the infeasible region is bounded by a chain of three-dimensional simplices. As a result, the boundary points appear on multiple layers of PaletteViz (in this case, the last two layers). It should be noted that the grey points surrounded by the red points in the last two layers are actually outside of the $\mathbf{H}_3$ homology group. Due to the fact that PaletteViz is collapsing two

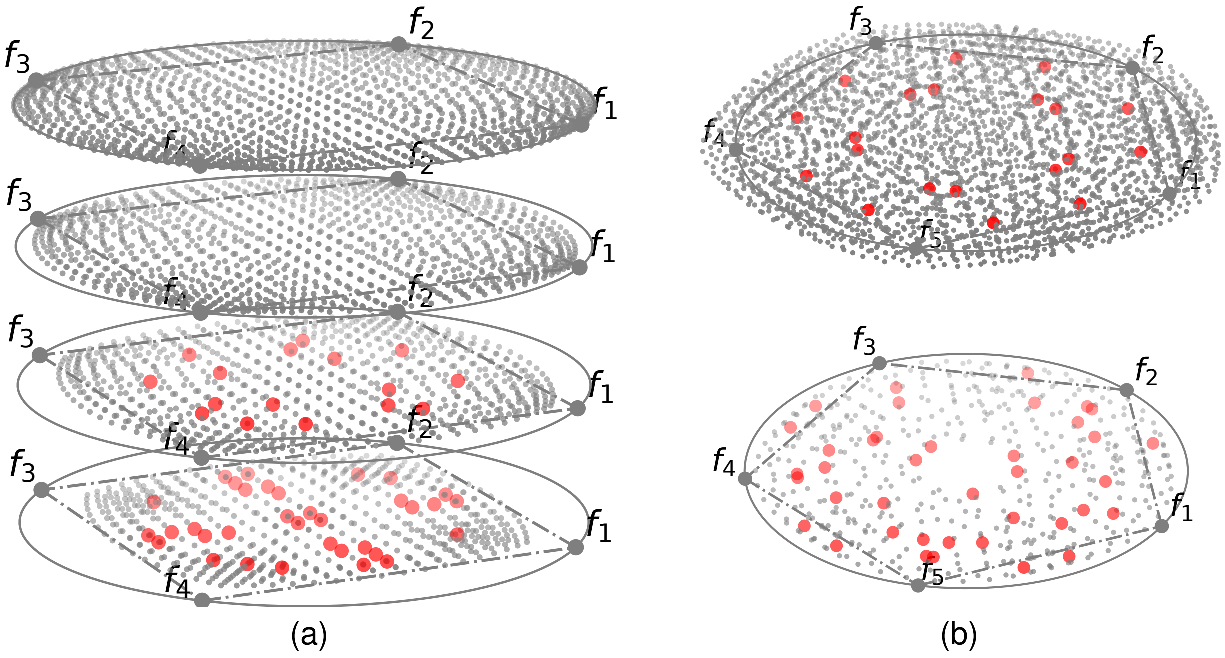


Figure 6.8: An example $\mathbf{H}_{m-1}$ boundary points visualization (a) A PaletteViz visualization of a four-dimensional spherical $\mathcal{PF}$ with the $\mathbf{H}_3$ homology group. (b) A PaletteViz visualization of a five-dimensional $\mathcal{PF}$ with the $\mathbf{H}_4$ homology group.

dimensions (four to two), the points outside of the $\mathbf{H}_3$ homology group are mapped inside in the last two layers. This is one limitation of PaletteViz . However, using these plots, we can at least identify other interesting insights about the data set. For example, looking at Figure 6.8a, we can say that the four-dimensional $\mathcal{PF}$ has a three-dimensional hole and that is located in the central region of $\mathcal{C}$.

In the end, we apply our method on a five-dimensional problem, which is shown in Figure 6.8b. Similarly, the points surrounding the $\mathbf{H}_4$ homology group are highlighted in red. Since $\mathcal{C}$ is in five dimensional space, the hole created by the infeasible region is bounded by a chain of four-dimensional simplices. As a result, the boundary points are found on both layers of PaletteViz. Due to the high-dimensionality of the space, it is difficult to find enough points on the Pareto-optimal surface in *general positions* [33]. As a result, most of the *Vietoris-Rips* complexes are coplanar. Consequently, the α-hull algorithm produces only two level sets. However, still we can see the boundary points around the infeasible region.

It is needless to mention here that a combination of interactive decision-making, boundary points to a hole identification, knee point identification, near constraint point identification and other visual aid methods is possible to display with the proposed PaletteViz method.

6.6 Chapter Summary

In this chapter, we have demonstrated how a multi-criterion decision-making procedure like the NIMBUS method can be combined with the PaletteViz visualization method. On three many-objective optimization problems involving up to 10 objectives and starting with thousands of non-dominated solutions obtained by an EMO procedure, we have demonstrated how the visualization approach helps to execute the NIMBUS and Pareto-race procedures in an informative manner to arrive at a single or a handful of preferred solutions. Due to the lack of a real DM in this study, we have used the proximity to a target non-dominated point as the basis for choosing preferred solutions in the NIMBUS and Pareto-race approach. The decision-making tasks are specific to individual problems, but a suitable visualization aid and step-by-step procedures can help allow DMs to exercise their preferences better in choosing a single preferred solution and in the process also learning more about multiple alternative solutions.

Whenever the problem (6.1) identifies fewer non-dominated points, a reference-point based EMO method [48] can be invoked to get more focused points. While other ideas can be implemented to make the combined approach more practical, the use of PaletteViz on Pareto-race or NIMBUS methods can as well be useful in different engineering domains, from aerospace to financial industry.

Lastly, in Section 6.5, we have demonstrated an example use case of PaletteViz applied in the topological data analysis (TDA) domain. Since PaletteViz is based on visualization of the level sets, this makes PaletteViz a natural tool to visualize the homology group boundaries. Given the fact that this particular implementation is still in the development phase, there is lots of room for improvement and it can be extended to meet other application scenarios.

CHAPTER 7

PARETO EXPLORER: A WEB-BASED SOFTWARE APPLICATION TO DEMONSTRATE THE CAPABILITIES OF PALETTEVIZ

Given all the discussions and analyses in the preceding chapters, it would be more interesting if we could provide a concrete demonstration of PaletteViz. In this chapter, we present a web application that implements many of the concepts and ideas discussed so far. We call this application `ParetoExplorer`. `ParetoExplorer` is a `node.js` [55] and `react.js`-based [56] `javascript` web application built on top of Apache `echart` [57]. In this chapter, we discuss its functionality in details.

7.1 The Landing Page, Menu, Scatter Plots and Data Summary

The landing page of `ParetoExplorer` is presented in Figure 7.1a. Figure 7.1b illustrates the drop-down menu on the left. The application is composed of a number of *panels* where the plots are rendered. The menu contains different Pareto-optimal fronts. A user can choose a particular data set from the menu to visualize and conduct analysis on it. In this example we show a scenario with the constrained knee CDEBMD1K problem (refer to Appendix A.1.1). The Figure 7.2a presents the scatter plot panel. The data that has been selected from the menu will show up in this window in scatter plot and by default, data will be plotted on the first three axes. The summary panel (Figure 7.2b) shows a summary of data that has been selected. It shows different basic information such as – total number of data points, number of objective functions, constraints; along with their distributions. In the scatter plot panel, the color-scheme and the axis choices can be customized through the control panel located at the bottom. An axis selection interaction is presented in Figure 7.3a. A color selection UI operation is shown in Figure 7.3b.

We have the similar color schemes as discussed in Chapter 4. Figure 7.4a shows how knee points (please refer to Section 4.1.4) can be made hidden by clicking the blue circular "Knees"

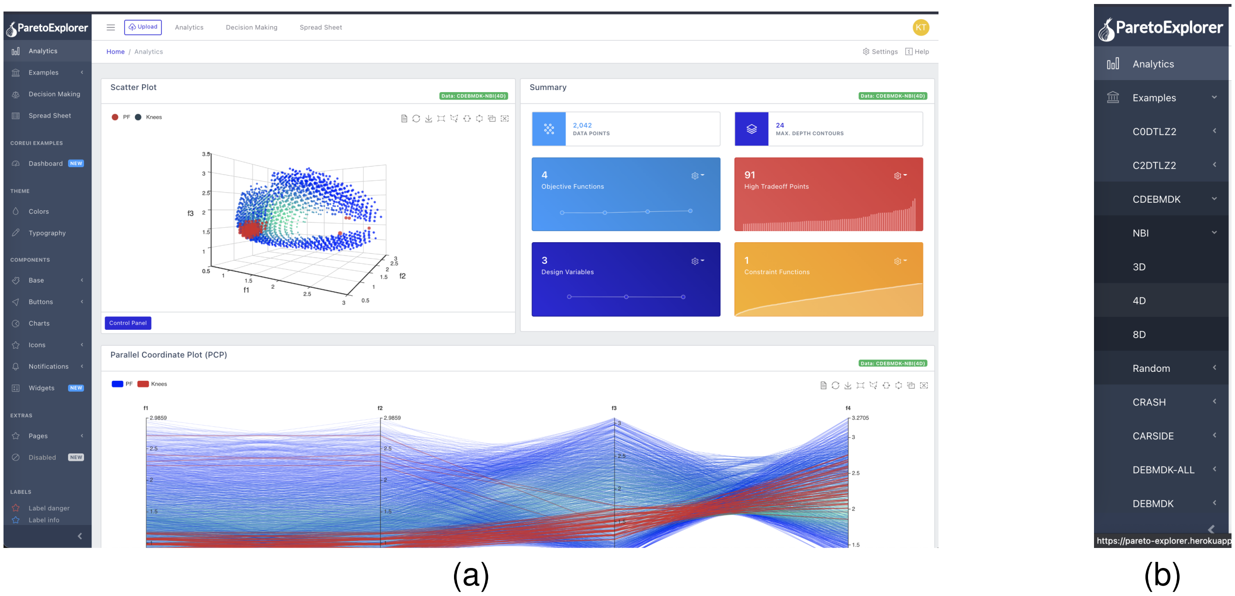


Figure 7.1: (a) ParetoExplorer main window interface. (b) ParetoExplorer exploratory menu on the left.

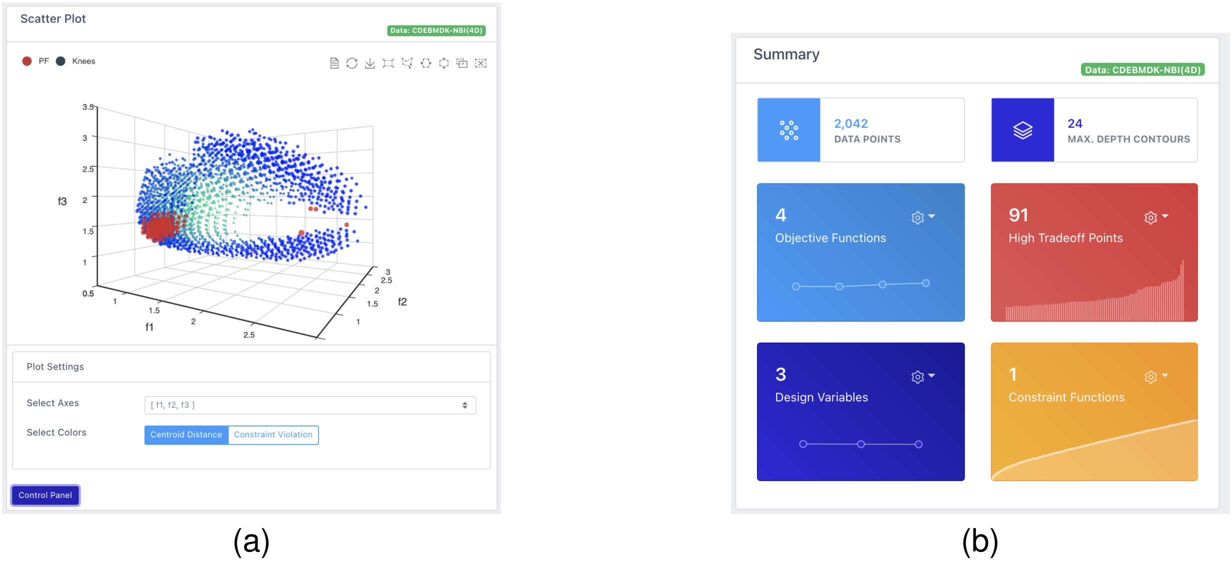
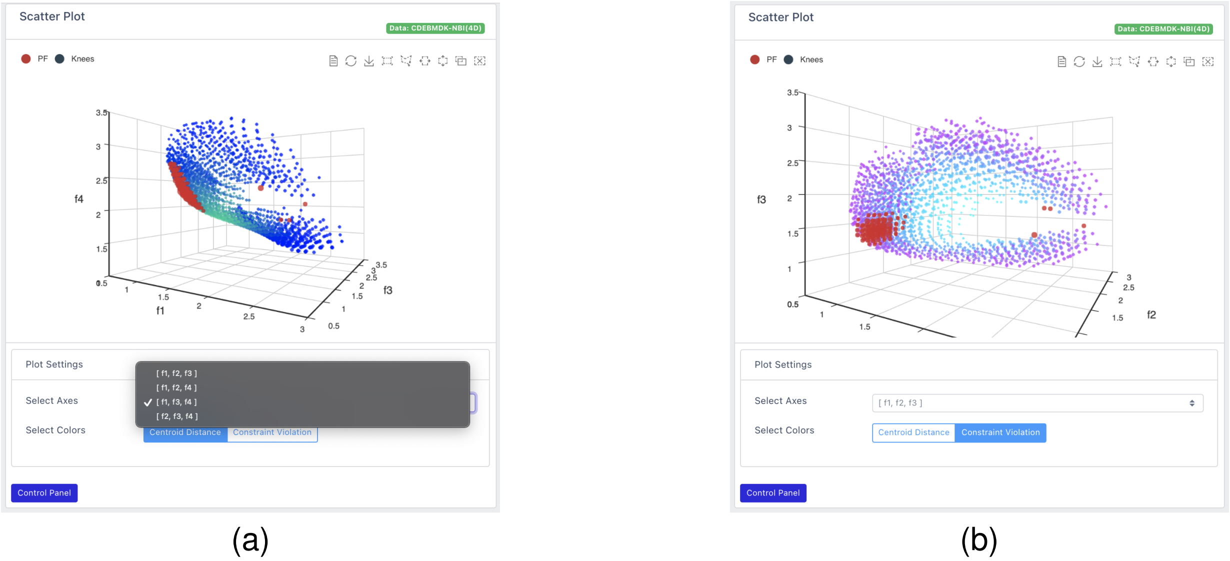


Figure 7.2: (a) Scatter plot panel. (b) Data set summary panel.

button on the top left corner. The plots can also be zoomed using scroll and Figure 7.4b shows how information about each data point appears when a user hovers over individual data points. In this case we see two floating menus appear.

The bigger menu on the top right corner shows the objective function values and the trade-off value (along with the point's index in the data set). The small window near the



Scatter Plot

Data: CDEBMDK-NBI(4D)

● PF ● Knees

Plot Settings

Select Axes

[f1, f2, f3]
[f1, f2, f4]
✓ [f1, f3, f4]
[f2, f3, f4]

Select Colors: Centroid Distance | Constraint Violation

Control Panel

(a)

Scatter Plot

Data: CDEBMDK-NBI(4D)

● PF ● Knees

Plot Settings

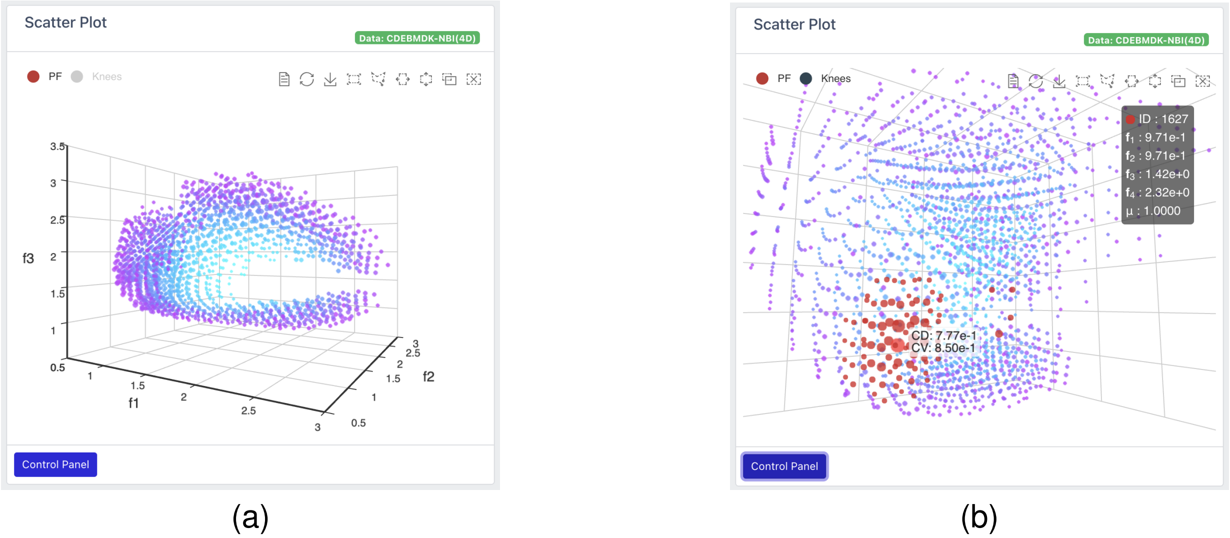
Select Axes: [f1, f2, f3]

Select Colors: Centroid Distance | Constraint Violation

Control Panel

(b)

Figure 7.3: (a) Axis selection for the scatter plot. (b) Color selection for the scatter plot.



Scatter Plot

Data: CDEBMDK-NBI(4D)

● PF ● Knees

Control Panel

(a)

Scatter Plot

Data: CDEBMDK-NBI(4D)

● PF ● Knees

ID : 1627
f_1 : 9.71e-1
f_2 : 9.71e-1
f_3 : 1.42e+0
f_4 : 2.32e+0
μ : 1.0000

CD: 7.77e-1
CV: 8.50e-1

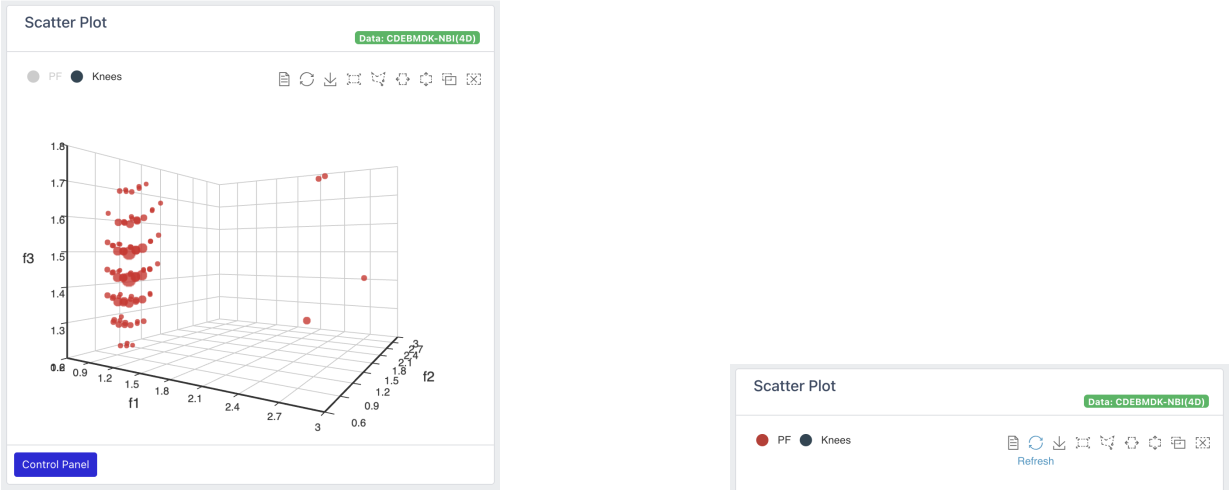
Control Panel

(b)

Figure 7.4: (a) Occluded knee points, i.e. Knee points are not clear in this picture. (b) Zooming in and examining individual data point.

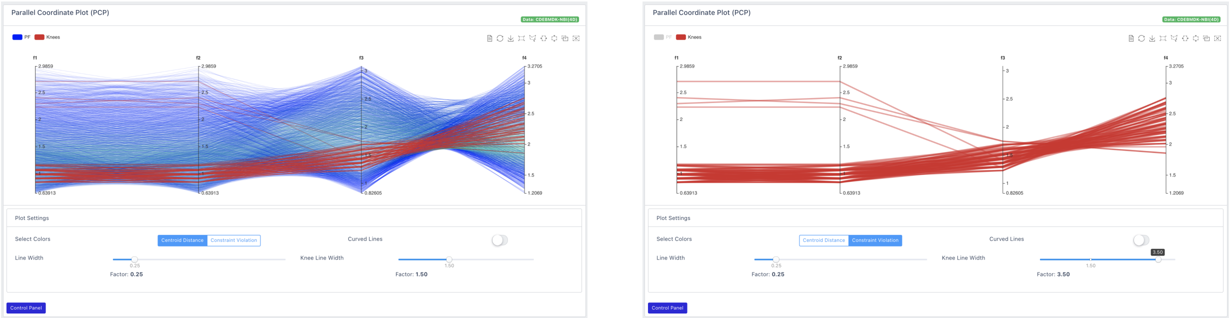
knee point shows normalized cumulative constraint violation value and the point's distance from the centroid.

In Figure 7.5a shows that the rest of the data points can be made hidden by clicking the red circular "PF" (Pareto-front) button. Now a DM can individually examine the knee points in the plot. The top right corner houses a number of graph control buttons, for example,



(a) (b)

Figure 7.5: (a) Keeping the knee points on the plot (b) A “Refresh” button – shown on the cropped part of the top edge of the panel.



Plot Settings
Select Colors: Centroid Distance | Constraint Violation
Curved Lines
Line Width 0.25 Factor: 0.25
Knee Line Width 1.50 Factor: 1.50

Plot Settings
Select Colors: Centroid Distance | Constraint Violation
Curved Lines
Line Width 0.25 Factor: 0.25
Knee Line Width 3.50 1.50 Factor: 3.50

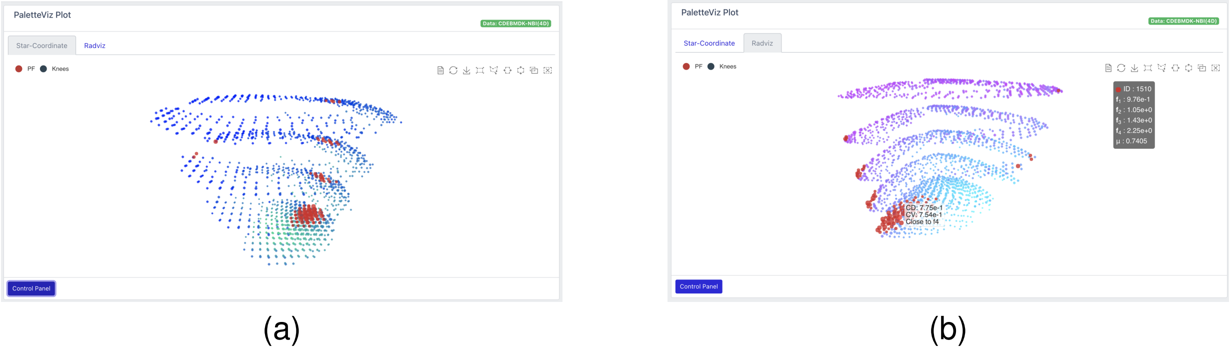
(a) (b)

Figure 7.6: (a) PCP plot with control panel opened. (b) PCP plot showing only high trade-off lines.

a circular sign with arrow-head is a “Refresh” button; this can be clicked to go back to the initial state of the plot, which is shown in Figure 7.5b

7.2 Parallel Coordinate Plot (PCP)

If a user scrolls down a bit, a panel for parallel coordinate plot (PCP) (Section 2.1) can be found. It uses the same color scheme as the scatter plot. The control panel houses a number of plot customization options. For example, line color, line width, knee line with etc. A view of the PCP and the control panel is presented in Figure 7.6a. In a similar



● ID : 1510
f_1 : 9.76e-1
f_2 : 1.05e+0
f_3 : 1.43e+0
f_4 : 2.25e+0
μ : 0.7405

CD: 7.75e-1
CV: 7.54e-1
Close to f4

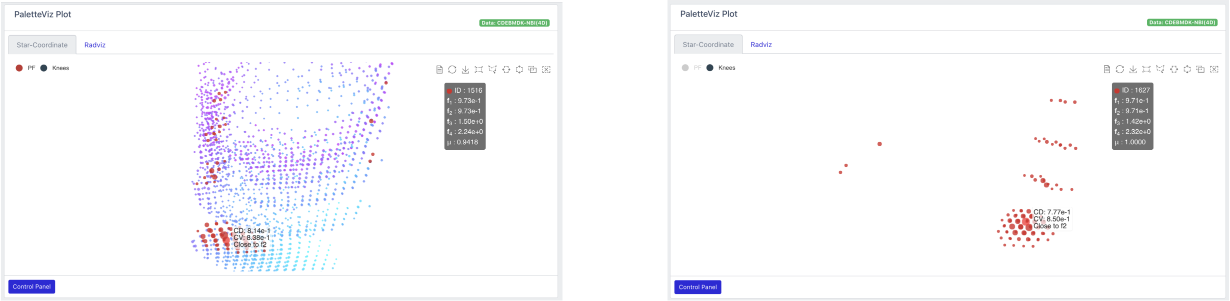
Figure 7.7: (a) PaletteViz plot showing depth-contour-based coloring of the data points. (b) PaletteViz plot showing a data point's information when a user hovers on the point.

manner, by selecting the "PF" button on the top left corner, a user can hide all the other lines except those of the high trade-off points. This interaction is shown in Figure 7.6b.

7.3 PaletteViz Plot

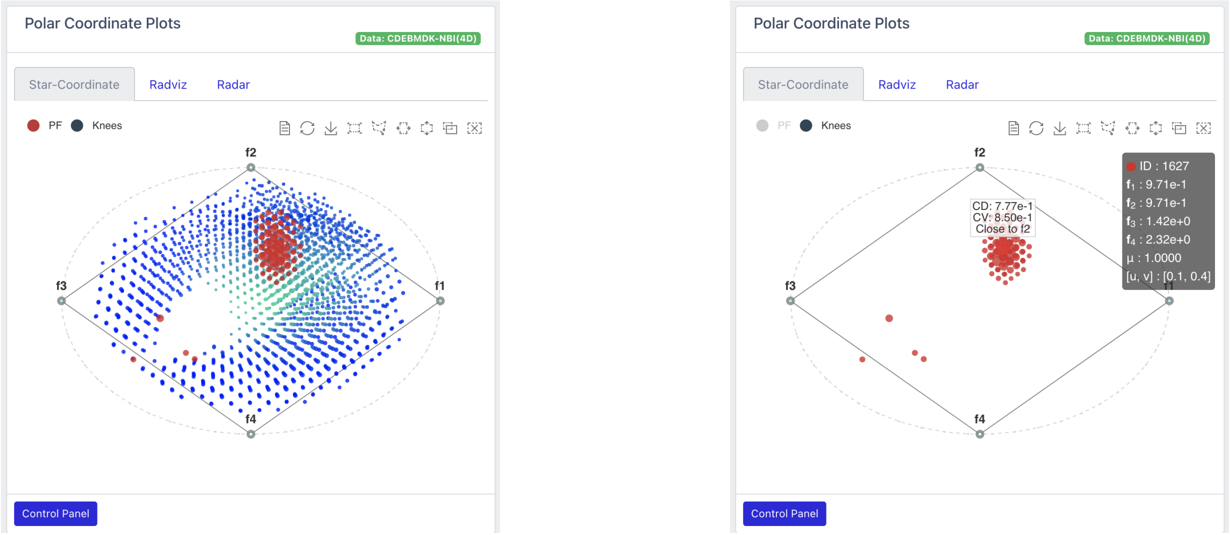
The next panel is dedicated to PaletteViz plots. We provide both star-coordinate and RadViz (Section 2.1.1) mapping. Two types of mapping are rendered on two different tabs at the top of the panel. All the other functionalities are similar to those of the scatter plot. A user can zoom in/out, hover, change color and inspect each point separately. Figure 7.7a presents PaletteViz plot with star-coordinate (Section 2.1.2) mapping. By clicking on the "RadViz" tab, a user can explore the PaletteViz plot with RadViz mapping. In Figure 7.7b, we demonstrate an interaction where the user has chosen a constraint-based color-scheme and is exploring a point near the high trade-off region. A small floating menu appears and it displays the cumulative normalized constraint violation value and the point's distance from the centroid.

This plot can be zoomed-in or out, shown in Figure 7.8a. Similar information about a particular data point appears if it's selected, as we have already seen in the scatter plot. By selecting "Knees" button, we can hide rest of the data points and only keep the high trade-off values alive on the plot. This interaction is shown in Figure 7.8b. The same interactions can be done on the PaletteViz plot with star-coordinate mapping as well.



(a) (b)

Figure 7.8: (a) PaletteViz plot while zoomed in and displaying information about a high trade-off point when selected. (b) PaletteViz plot showing only the knee points.



(a) (b)

Figure 7.9: (a) Star-coordinate plot of CDEBMD1K. (b) Only displaying the knee points and other points are hidden from the view.

7.4 Polar Coordinate Plots: Star-coordinate, RadViz and Radar Plots

In order to keep most of the existing visualization methods, we have included the standard star-coordinate, RadViz and radar plot in the next panel. This panel can be found under the left of the PaletteViz plot panel. Figure 7.9a shows a star-coordinate plot of the data. The default color scheme is based on depth-contour. Figure 7.9b shows only the knee points. We can select a particular point and view its corresponding information.

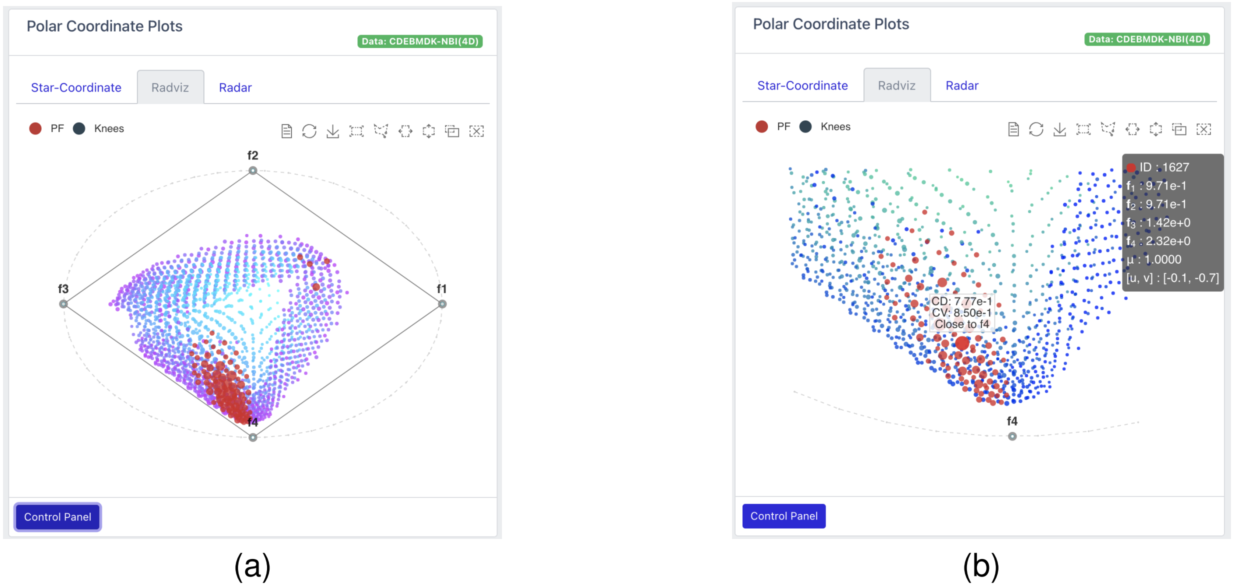


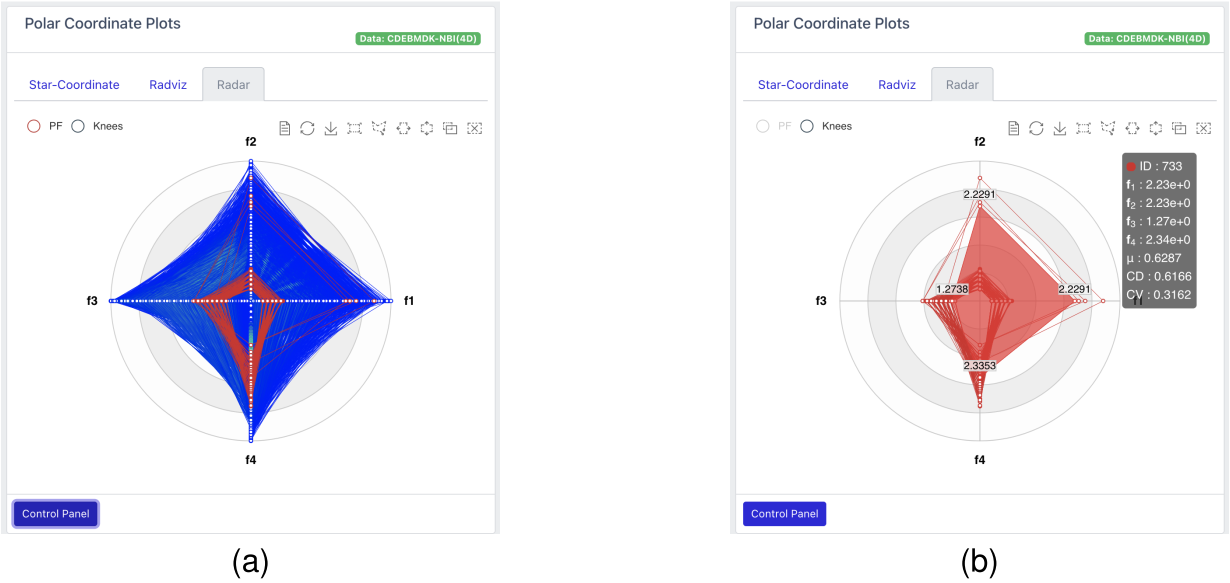
Figure 7.10: (a) A RadViz plot of CDEBKD1K with normalized cumulative constraint-violation-based color scheme (b) A user zooms into the knee region of the Pareto-front on RadViz.

Next, if we go the next tab, we will find a RadViz plot of the data, which is shown in Figure 7.10a. A user can zoom-in/out or scroll the points to explore different parts of the Pareto-optimal front. In this case, the RadViz is using the constraint-violation-based color scheme. Figure 7.10b shows an interaction where the user has zoomed and scrolled into the knee region of the Pareto-front. The relevant information appear in a similar way.

The next tab houses the radar plot. The UI interactions are similar to other plots. A radar plot is presented in Figure 7.11a. A user can also filter out the rest of the data points and keep the high trade-off points, which is shown in Figure 7.11b. In this case, when individual points are selected, the radar plot highlights the corresponding polygon.

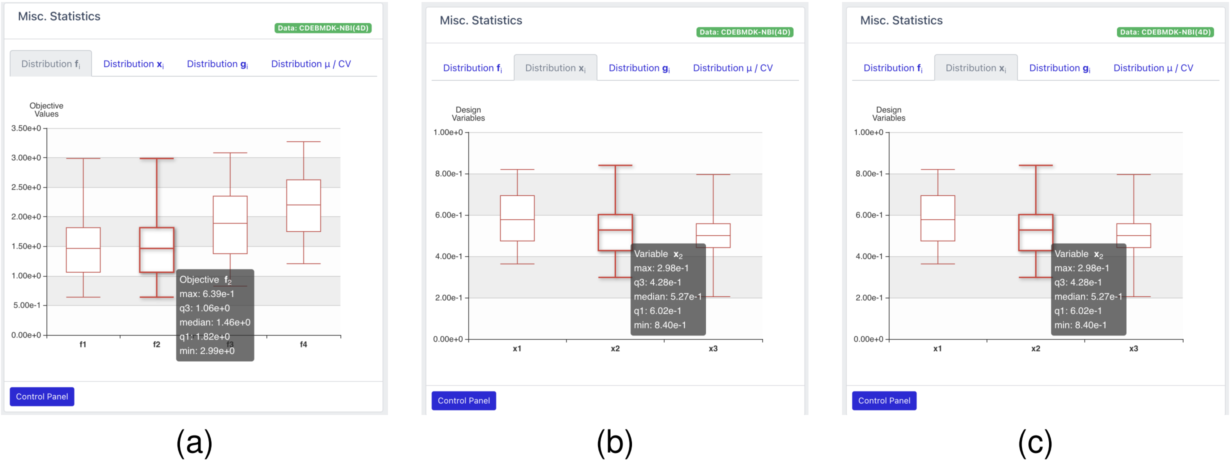
7.5 Other Statistics

The last panel provides statistical information on the design variables, objective and constraint function values, on respective tabs. The last tab shows distribution of trade-off and normalized cumulative constraint violation values. Figure 7.12a shows the distribution of objective functions values as box-plots. Each box can be individually selected and their



(a) (b)

Figure 7.11: (a) A radar plot with depth-contour-based coloring scheme. (b) The knee points are kept and the rest of the data points are hidden. Mouse selection on a point highlights the corresponding radar polygon.

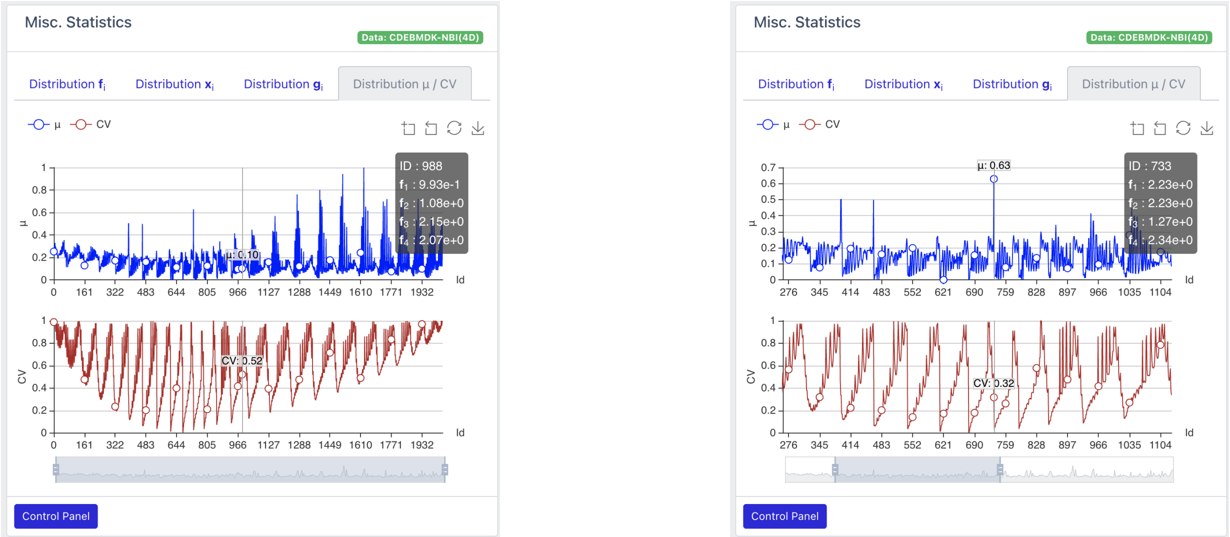


(a) (b) (c)

Figure 7.12: (a) Box plot for the distribution of individual objective function values. (b) Box plot for the distribution of individual design variable values. (c) Box plot for the distribution of individual normalized constraint function values.

corresponding statistics are displayed in a floating menu. Similar statistics can viewed inside the tabs for design variables and normalized constraint function values. The are presented in Figure 7.12b and Figure 7.12c, respectively.

The last tab contains bar plots for the trade-off and cumulative normalized constraint violation values. They are presented on two separate rows, the top row contains the bar

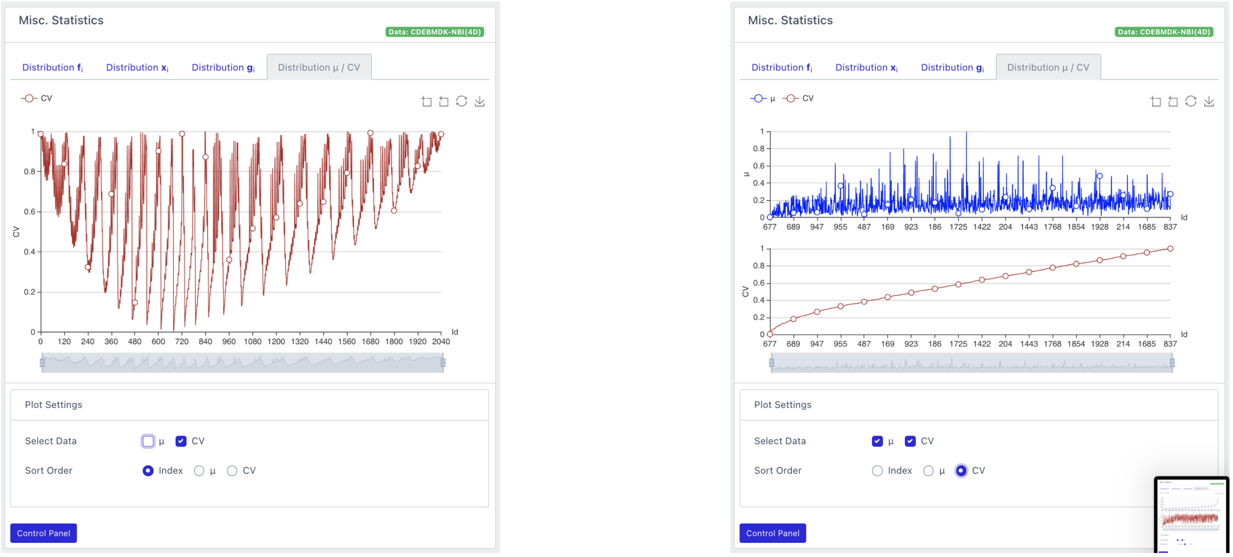


(a) (b)

Figure 7.13: (a) Bar plot for the trade-off and normalized cumulative constraint violation values. (b) A view when the bar plots have been zoomed using the horizontal scroll bar at the bottom.

plot for the trade-off values (in blue). When a mouse is being hovered, individual μ and constraint violation (CV) values will be displayed along an aligned vertical line, which has been shown in Figure 7.13a. Using the horizontal scroll bar, these plots can be farther zoomed in and examined, this is shown in Figure 7.13b. By collapsing the control panel, a user can select/deselect particular bar plot. In Figure 7.14a, we can see the user has chosen the bar plot for trade-off values. Using the “sort” option, we can sort the bar plot according to point indices, trade-off values or constraint violation values. If one bar plot is sorted, the other plot will be sorted accordingly.

In this way, ParetoExplorer can be helpful to conduct a detailed visual analysis on a Pareto-optimal front. Since ParetoExplorer provides different kinds of visualization methods for the same data set, a DM can easily do comparative exploration of the data points. A scatter plot gives an overview, a PCP plot provides a quantitative comparison among the objectives, PaletteViz provides topological and spatial information etc. With a set of orthogonal aspects in the visualization, ParetoExplorer can be a useful tool for practical MCDM.



Plot Settings

Select Data ☐ μ ☑ CV

Sort Order ● Index ○ μ ○ CV

Plot Settings

Select Data ☑ μ ☑ CV

Sort Order ○ Index ○ μ ● CV

(a)

(b)

Figure 7.14: (a) Deselect one of the bar plots using the control panel. (b) Sorting the bar plots using the control panel.

7.6 Chapter Summary

In this chapter, we have discussed how the `ParetoExplorer` application can be used in a basic visual MCDM. The software still needs further development for it to be widely used. We do not have any feature for custom data uploading. Also the application does not provide any interactive MCDM using PaletteViz as we have discussed in great detail in Chapter 6. An interactive visual MCDM framework needs to be incorporated with a different plotting scheme in `ParetoExplorer`. Given those limitations, this application is still useful in conducting initial/basic visual analytics on the Pareto-optimal front.

CHAPTER 8

CONCLUSIONS

In this dissertation, we have demonstrated a new visualization technique – PaletteViz – to address the issue of high-dimensional Pareto-optimal data-set visualization and its application to multi-criteria decision making (MCDM) analytics. High-dimensional and non-dominated objective vectors have been mapped into several two-dimensional RadViz plots and stacked according to their *inward-outward relationships* in the original high-dimensional space. Unlike the existing visualization methods, PaletteViz technique employes concepts from the computational topology domain and utilizes those ideas. This makes PaletteViz approach drastically different from all the other existing methods for high-dimensional data visualization, especially in MCDM domain. PaletteViz can utilize different color and marker sizing schemes to filter and highlight other functional and decision making properties by clearly marking a few critical points, which would be of further interest to decision-makers (DMs). We have also seen how PaletteViz capabilities can be extended to other domain, namely in visualization of $\mathbf{H}_{m-1}$ homology group boundaries. In that aspect, PaletteViz can be useful to topological data analysis (TDA) practitioners as well. Hopefully, in this way PaletteViz will be able to create a bridge between TDA and MCDM practioners.

Finally, with all the different visualization capabilities, an user-interactive software that allows DMs to make multiple plots along with PaletteViz can be very useful in real-world MCDM process. Such a method will enable a proper functional visualization of multi-criterion data-set before a single or a handful of points are chosen. Finally, the proposed PaletteViz technique should also be tried for high-dimensional data visualization purposes in other scientific domains, specially to the machine learning and data-mining communities.

8.1 Future Developments

This proof-of-principle study can be extended with modifications of each step. The Radviz mapping can be changed with other two-dimensional mappings, such as t-SNE. The decomposition of layers can be changed with other point depth metrics, instead of the non-convex hull peeling depth, e.g. Chebyshev centrality measure. Moreover, multivariate depth ranking that preserves the directions of points [58] can also be used. A recent study [59] used a three-dimensional Radviz plot (3D Radviz) where the z-axis is being used to represent the distance of a point from a hyper-plane to illustrate the convexity of the point set. Layers can also be made from such plots. To extend the performance-based preferencial information, the use of recently-proposed KKT proximity measure (KKTPM)[1] [60] would be a natural choice. A KKTPM based coloring approach could provide additional information to the DM about each point's convergence measure, if this must be a requirement for the choice of a preferred point. Although we have not tried here, but a robustness measure or a reliability measure can be computed for each point in problems with uncertainties in parameters and variables. The points can then be classified based on their robustness or reliability measure.

A local convexity measure can also be used as another classification scheme. In terms of preferetial information aspect, any existing MCDM-based utility function, instead of trade-off function can be also used. Different dominance relationships, such as properly Pareto-optimal points with certain pre-defined minimum trade-off value, or the ϵ-Pareto-optimal points with a pre-defined ϵ can be used to mark the critical points.

Furthermore, the `ParetoExplorer` web application can be improved. For example, the current version does not support file uploading and automated MCDM methods discussed in Chapter 6. In the future, we would like to complete its application for general use. In summary, there is a lot of room to improve and extend the ideas in this thesis, so that PaletteViz can be applied to a wide range of engineering and data analytics domains.

[1] KKTP can quantify the proximity of a non-dominated point to the true Pareto-front.

APPENDICES

APPENDIX A

LIST OF BENCHMARK PARETO-OPTIMAL FUNCTIONS

A.1 Pareto-optimal Front Generator Functions

In this appendix, we present the Pareto-optimal front generator functions for the problems used in this study. This paper assumes all the objective functions (i.e. $\mathbf{f} = [f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})]^T$) are to be minimized and in all cases there are n variables $\mathbf{x} = [x_1, x_2, \ldots, x_n]^T$.

A.1.1 Problem with a Knee (DEBMD1K)

This is a slightly modified version of DEBMDK problem described in [1], so that the resulting Pareto-optimal front will have only one knee irrespective of the number of objectives. The problem is defined below:

$$\begin{aligned}
\min\ f_m(\mathbf{x}) &= g(\mathbf{x})r(\mathbf{x}) \left[\prod_{i=1}^{m-2} \sin\left(\frac{\pi}{2}x_i\right)\right] \sin\left(\frac{\pi}{2}x_{M-1}\right), \\
\min\ f_{m-1}(\mathbf{x}) &= g(\mathbf{x})r(\mathbf{x}) \left[\prod_{i=1}^{m-2} \sin\left(\frac{\pi}{2}x_i\right)\right] \cos\left(\frac{\pi}{2}x_{m-1}\right), \\
\min\ f_{m-2}(\mathbf{x}) &= g(\mathbf{x})r(\mathbf{x}) \left[\prod_{i=1}^{m-3} \sin\left(\frac{\pi}{2}x_i\right)\right] \cos\left(\frac{\pi}{2}x_{m-2}\right), \\
&\vdots \\
\min\ f_2(\mathbf{x}) &= g(\mathbf{x})r(\mathbf{x}) \sin\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right), \\
\min\ f_1(\mathbf{x}) &= g(\mathbf{x})r(\mathbf{x}) \cos\left(\frac{\pi}{2}x_1\right), \\
\text{where, } g(\mathbf{x}) &= 1 + \frac{9}{n-m+1} \sum_{i=m}^{n} x_i,\ r(\mathbf{x}) = \frac{1}{m-1} \sum_{i=1}^{m-1} r_i(x_i), \\
r_i(x_i) &= 5 + 10\,(x_i - 0.5)^2 + \frac{2}{K}\cos(2\pi x_i K), \\
\mathbf{x} &\in [0,1]^n.
\end{aligned} \tag{A.1}$$

The knee point is located at $\mathbf{x} = [0.5]^n$. This fixed location of the knee is useful to see how the mapping/visualization distorts the topology in higher dimension. To have a single knee, we have used $K = 1$ and since it's fundamentally an equation of a hypershpere, we can set $n \geq m - 1$. An example Pareto-optimal front for this problem in $m = 3$ is illustrated in Figure 2.5.

A.1.2 Problem with an Isolated Region (C0DTLZ2)

To construct a Pareto-optimal front with one isolated part and another major part, we add the following constraint to the original DTLZ2 [43] problem:

$$(0.98 - f_m(\mathbf{x}))(f_m(\mathbf{x}) - 0.75) \leq 0. \tag{A.2}$$

This problem has one relatively smaller outlier region near $f_i = 0$ where $i = 1, 2, \ldots, (m-1)$ and $f_m = 1$. However, the majority of Pareto-optimal front lies for $f_m \leq 0.75$. The problem is defined in such a way that irrespective of the number of dimensions, there will always two disjointed patches – one small and one large. An example Pareto-optimal front for this problem for $m = 3$ was presented in Figure 5.5a.

A.1.3 Constrained Problem with Multiple Clusters (C2DTLZ2)

In this case, we have used the C2-DTLZ2 problem described in [51]. The objective functions are calculated in the same way as in the original DTLZ2 [43] problem, except that a constraint is now introduced:

$$\begin{aligned} c(\mathbf{x}) = -\min \left\{ \min_{i=1}^{m} \left[(f_i(\mathbf{x}) - 1)^2 + \sum_{j=1, j\neq i}^{m} f_j^2 - r_1^2 \right], \right. \qquad & \qquad \text{(A.3)} \\ \left. \left[\sum_{i=1}^{m} (f_i(\mathbf{x}) - 1/\sqrt{m})^2 - r_2^2 \right] \right\} \geq 0, & \end{aligned}$$

where $r_1 = 0.4$, for $m = 3$ and 0.5, otherwise, and $r_2 = r_1/2$.

A.1.4 Constrained Problem with Differing Dimensions (DTLZ8)

The next problem is an adaptation of the constrained DTLZ8 problem [43]:

$$\begin{aligned} \min\ f_i(\mathbf{x}) &= x_i,\ \text{ for } i = 1, 2, \ldots, m, \qquad \text{(A.4)} \\ \text{s.t. } g_j(\mathbf{x}) &= x_m + m x_j - 1 \geq 0,\ \text{ for } j = 1, 2, \ldots, (m-1), \\ g_m(\mathbf{x}) &= 2x_m + \sum_{j=1}^{m-1} x_j - 1 \geq 0, \\ \mathbf{x} &\in [0, 1]^m. \end{aligned}$$

The Pareto-optimal front is a combination of a straight line from $(0, \ldots, 0, 1)^T$ to $\frac{1}{m+1}(1, \ldots, 1, 1)^T$ and a linear hyper-plane. An example of the Pareto-optimal front for this problem for $m = 3$ was presented in Figure 6.5a.

A.1.5 Constrained Problem with a Knee (CDEBMD1K)

The next problem is the same knee problem defined in Subsection A.1.1, but a constraint is now added:

$$\sum_{i=1}^{m-1} (f_i(\mathbf{x}) - 1.75)^2 + (f_m(\mathbf{x}) - 2.1)^2 \leq r^2, \tag{A.5}$$

where $r = \{1.5, 1.5, 1.75\}$ for $m = 3$, 4, and 8, respectively. This problem keeps the central knee region feasible, but makes the extreme parts of the original Pareto-optimal front infeasible. Thus, this problem will allow the PaletteViz technique to demonstrate its working with a constraint problem and also having a knee point. An example three-dimensional Pareto-optimal front was presented in Figure 5.3a.

A.1.6 m-dimensional Spherical Pareto-optimal Front with $\mathbf{H}_{m-1}$ Homology Groups

This problem is same as DTLZ2 [43] but we introduce a constraint function that creates a hole on the surface:

$$\sum_{i=1}^{m-1} (f_i(\mathbf{x}) - \mu)^2 \geq r^2 \tag{A.6}$$

where μ is the centroid of the data point:

$$\mu = \frac{1}{n} \left[\sum_{i=1}^{n} f_1(\mathbf{x}_i), \sum_{i=1}^{n} f_2(\mathbf{x}_i), \ldots, \sum_{i=1}^{n} f_m(\mathbf{x}_i) \right]^T \tag{A.7}$$

and $r = \{0.15, 0.2, 0.25\}$ for $m = 3, 4$, and 5, respectively. Moreover, $m = n - 1$ and $m \geq 3$. The Parto-optimal front for a three dimensional example is given in Figure 6.7a. Using the similar principle, we can introduce more than one hole on the Pareto-optimal surface. In that case, the problem designer needs to choose different suitable μ value (a point in the space of $\mathcal{C}$) for the experiment.

APPENDIX B

COMPUTER SOFTWARE IMPLEMENTATION

B.1 Software Implementations

A set of software libraries and tools have been implemented during the development of this thesis and they are listed in this section.

B.1.1 pviz: A PaletteViz Implementation in Python

A full-fledge software library that implements PaletteViz and other related visualization methods. This library also provides different benchmark test functions used in this thesis. pviz is a free software (Apache License 2.0) and can be downloaded through this link:

https://gitlab.msu.edu/talukde1/pviz

pviz is also available via pypi:

https://pypi.org/project/pviz/

B.1.2 pviz-bench: Benchmarking pviz

This is a collection of programs and python notebooks to showcase different capabilitis of PaletteViz . Almost all results in this thesis have been generated using pviz-bench. This is also free (Apache License 2.0) and can be downloaded from this link:

https://gitlab.msu.edu/talukde1/pviz-bench

B.1.3 ParetoExplorer: A Web Based Application

ParetoExplореris a functional web application built on top of node.js, react.js and apache-echart, which showcases different capabilities of pviz. This is also an open source software (Apache License 2.0) and the source code can be downloaded through this link:

https://gitlab.msu.edu/talukde1/pareto-explorer

ParetoExplorer is hosted on *Heroku* and can be accessed through this link:

https://pareto-explorer.herokuapp.com/

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Jürgen Branke, Kalyanmoy Deb, Henning Dierolf, and Matthias Osswald. Finding knees in multi-objective optimization. In Xin Yao, Edmund K. Burke, José A. Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiňo, Ata Kabán, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature - PPSN VIII*, Lecture Notes in Computer Science, pages 722–731. Springer Berlin Heidelberg, 2004. ISBN 978-3-540-30217-9.

[2] Xingtao Liao, Qing Li, Xujing Yang, Weigang Zhang, and Wei Li. Multiobjective optimization for crash safety design of vehicles using stepwise regression model. *Structural and Multidisciplinary Optimization*, 35(6):561–569, June 2008. ISSN 1615-1488. doi: 10.1007/s00158-007-0163-x.

[3] A. K. A. Talukder and K. Deb. A topologically consistent visualization of high dimensional Pareto-front for multi-criteria decision making. In *Proceedings of 2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1579–1586, Nov. 2018. doi: 10.1109/SSCI.2018.8628892.

[4] Andrzej P. Wierzbicki. The use of reference objectives in multiobjective optimization. In Günter Fandel and Tomas Gal, editors, *Multiple Criteria Decision Making Theory and Application*, Lecture Notes in Economics and Mathematical Systems, pages 468–486. Springer Berlin Heidelberg, 1980. ISBN 978-3-642-48782-8.

[5] Yash Vesikar, Kalyanmoy Deb, and Julian Blank. Reference point based NSGA-III for preferred solutions. In *Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI-2018)*, Bengaluru, India, November 2018. IEEE.

[6] Ralph E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application.* Wiley, 1986. ISBN 978-0-471-85970-3.

[7] Vira Chankong and Yacov Y. Haimes. *Multiobjective Decision Making: Theory and Methodology.* Dover Publications, Mineola, N.Y, 2008 edition, February 2008. ISBN 978-0-486-46289-9.

[8] Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* International Series in Operations Research & Management Science. Springer US, 1998. ISBN 978-0-7923-8278-2.

[9] Alfred Inselberg and Tova Avidan. Classification and visualization for high-dimensional data. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '00, pages 370–374, New York, NY, USA, 2000. ACM. ISBN 978-1-58113-233-5. doi: 10.1145/347090.347170.

[10] Patrick Hoffman, Georges Grinstein, and David Pinkney. Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations. In *Proceedings of the 1999 Workshop on New Paradigms in Information Visualization and Manipulation in Conjunction with the Eighth ACM Internation Conference on Information and Knowledge Management*, NPIVM '99, pages 9–16, New York, NY, USA, 1999. ACM. ISBN 978-1-58113-254-0. doi: 10.1145/331770.331775.

[11] Andy Pryke, Sanaz Mostaghim, and Alireza Nazemi. Heatmap visualization of population based multi objective algorithms. In Shigeru Obayashi, Kalyanmoy Deb, Carlo Poloni, Tomoyuki Hiroyasu, and Tadahiko Murata, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 361–375. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-70928-2.

[12] Shigeru Obayashi and Daisuke Sasaki. Visualization and data mining of pareto solutions using self-organizing map. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Lothar Thiele, and Kalyanmoy Deb, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 796–809. Springer Berlin Heidelberg, 2003. ISBN 978-3-540-36970-7.

[13] T. Tušar and B. Filipič. Visualization of pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 19(2):225–245, April 2015. ISSN 1089-778X. doi: 10.1109/TEVC.2014.2313407.

[14] Z. He and G. G. Yen. Visualization and performance metric in many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 20(3):386–402, June 2016. ISSN 1089-778X. doi: 10.1109/TEVC.2015.2472283.

[15] K. Miettinen and M.M. Mäkelä. Interactive bundle-based method for nondifferentiable multiobjeective optimization: nimbus. *Optimization*, 34(3):231–246, 1995. doi: 10.1080/02331939508844109. URL `https://doi.org/10.1080/02331939508844109`.

[16] Pekka Korhonen and Jyrki Wallenius. A Pareto race. *Naval Research Logistics (NRL)*, 35(6):615–623, 1988. doi: 10.1002/1520-6750(198812)35:6<615::AID-NAV3220350608>3.0.CO;2-K.

[17] Maurice d'Ocagne. *Coordonnées parallèles & axiales: méthode de transformation géométrique et procédé nouveau de calcul graphique déduits de la considération des coordonnées parallèles*. Gauthier-Villars, 1885.

[18] X. Yan, C. F. Lai, and C. W. Fu. Visual Signature of High-Dimensional Geometry in Parallel Coordinates. In *2014 IEEE Pacific Visualization Symposium*, pages 65–72, March 2014. doi: 10.1109/PacificVis.2014.41.

[19] Leland Wilkinson and Michael Friendly. The History of the Cluster Heat Map. *The American Statistician*, 63(2):179–184, May 2009. ISSN 0003-1305. doi: 10.1198/tas.2009.0033.

[20] P. Hoffman, G. Grinstein, K. Marx, I. Grosse, and E. Stanley. DNA visual and analytic data mining. In *Proceedings. Visualization '97 (Cat. No. 97CB36155)*, pages 437–441, October 1997. doi: 10.1109/VISUAL.1997.663916.

[21] Eser Kandogan. Visualizing Multi-dimensional Clusters, Trends, and Outliers Using Star Coordinates. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 107–116, New York, NY, 2001. ACM. ISBN 978-1-58113-391-2. doi: 10.1145/502512.502530.

[22] M. Rubio-Sánchez, L. Raya, F. Díaz, and A. Sanchez. A comparative study between RadViz and Star Coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):619–628, January 2016. ISSN 1077-2626. doi: 10.1109/TVCG.2015.2467324.

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008. ISSN ISSN 1533-7928.

[24] S. Liu, D. Maljovec, B. Wang, P. Bremer, and V. Pascucci. Visualizing High-Dimensional Data: Advances in the Past Decade. *IEEE Transactions on Visualization and Computer Graphics*, 23(3):1249–1268, March 2017. ISSN 1077-2626. doi: 10.1109/TVCG.2016.2640960.

[25] L. Rachmawati and D. Srinivasan. Multiobjective evolutionary algorithm with controllable focus on the knees of the pareto front. *IEEE Transactions on Evolutionary Computation*, 13(4):810–824, August 2009. ISSN 1089-778X. doi: 10.1109/TEVC.2009.2017515.

[26] A. K. M. Khaled Ahsan Talukder, Kalyanmoy Deb, and Shahryar Rahnamayan. Injection of extreme points in evolutionary multiobjective optimization algorithms. In Heike Trautmann, Günter Rudolph, Kathrin Klamroth, Oliver Schütze, Margaret Wiecek, Yaochu Jin, and Christian Grimme, editors, *Evolutionary Multi-Criterion Optimization*, Lecture Notes in Computer Science, pages 590–605. Springer International Publishing, 2017. ISBN 978-3-319-54157-0.

[27] John Lee. *Introduction to Topological Manifolds*. Graduate Texts in Mathematics. Springer-Verlag, New York, 2 edition, 2011. ISBN 978-1-4419-7939-1.

[28] J.R. Munkres. *Elements Of Algebraic Topology*. Avalon Publishing, 1996. ISBN 9780201627282.

[29] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2(1):18–21, March 1973. ISSN 0020-0190. doi: 10.1016/0020-0190(73)90020-3.

[30] Edelsbrunner, Letscher, and Zomorodian. Topological persistence and simplification. *Discrete & Computational Geometry*, 28(4):511–533, Nov 2002. ISSN 1432-0444. doi: 10.1007/s00454-002-2885-2. URL `https://doi.org/10.1007/s00454-002-2885-2`.

[31] Regina Y. Liu, Jesse M. Parelius, and Kesar Singh. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics*, 27(3):783–840, 1999. ISSN 0090-5364.

[32] Dmitry Krasnoshchekov and Valentin Polishchuk. Order-k α-hulls and α-shapes. *Information Processing Letters*, 114(1):76–83, January 2014. ISSN 0020-0190. doi: 10.1016/j.ipl.2013.07.023.

[33] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, July 1983. ISSN 0018-9448. doi: 10.1109/TIT.1983.1056714.

[34] M. H. Gross and R. Koch. Visualization of multidimensional shape and texture features in laser range data using complex-valued Gabor wavelets. *IEEE Transactions on Visualization and Computer Graphics*, 1(1):44–59, March 1995. ISSN 1077-2626. doi: 10.1109/2945.468389.

[35] Shishir Pandey and Rahul Vaze. Trustworthiness of t-distributed stochastic neighbour embedding. In *Proceedings of the 3rd IKDD Conference on Data Science, 2016*, CODS '16, pages 17:1–17:2, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4217-9. doi: 10.1145/2888451.2888465.

[36] Avi Levy. The circumradius of a simplex via edge lengths. Technical report, Ind. Res., February 2017.

[37] Kenta Kobayashi. A recursive formula for the circumradius of the n-simplex. *Forum Geometricorum*, 16:179–174, April 2016. ISSN 1534-1178.

[38] Laurens van der Maaten and Geoffrey Hinton. Visualizing non-metric similarities in multiple maps. *Machine Learning*, 87(1):33–55, April 2012. ISSN 1573-0565. doi: 10.1007/s10994-011-5273-4.

[39] Himangshu Jain and Kalyanmoy Deb. An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach. *IEEE Transactions on Evolutionary Computation*, 18(4):602–622, August 2014. ISSN 1089-778X. doi: 10.1109/TEVC.2013.2281534.

[40] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[41] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, May 2002. ISSN 0162-8828. doi: 10.1109/34.1000236.

[42] Bogdan Filipič and Tea Tušar. A taxonomy of methods for visualizing pareto front approximations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '18, pages 649–656, New York, NY, USA, 2018. ACM. ISBN 978-1-4503-5618-3. doi: 10.1145/3205455.3205607.

[43] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, volume 1, pages 825–830 vol.1, May 2002. doi: 10.1109/CEC.2002.1007032.

[44] T.W. Simpson, J.K. Allen, W. Chen, and F. Mistree. Conceptual design of a family of products through the use of the robust concept exploration method. In *6th Symposium on Multidisciplinary Analysis and Optimization*, pages 1535–1545, 1996.

[45] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387332545.

[46] Kalyanmoy Deb, Ankur Sinha, Pekka J. Korhonen, and Jyrki Wallenius. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739, 2010. doi: 10.1109/TEVC.2010.2064323.

[47] Jürgen Branke, Salvatore Greco, Roman Słowiński, and Piotr Zielniewicz. Interactive evolutionary multiobjective optimization using robust ordinal regression. In Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux, editors, *Evolutionary Multi-Criterion Optimization*, pages 554–568, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-01020-0.

[48] Kalyanmoy Deb and J. Sundar. Reference Point Based Multi-objective Optimization Using Evolutionary Algorithms. In *Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation*, GECCO '06, pages 635–642, New York, NY, USA, 2006. ACM. ISBN 978-1-59593-186-3. doi: 10.1145/1143997.1144112.

[49] Kalyanmoy Deb and Abhishek Kumar. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, GECCO '07, page 781–788, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936974. doi: 10.1145/1276958.1277116. URL `https://doi.org/10.1145/1276958.1277116`.

[50] Kalyanmoy Deb and Abhay Kumar. Light beam search based multi-objective optimization using evolutionary algorithms. In *2007 IEEE Congress on Evolutionary Computation*, pages 2125–2132, 2007. doi: 10.1109/CEC.2007.4424735.

[51] Kalyanmoy Deb and Himangshu Jain. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, 18(4):577–601, August 2014. ISSN 1089-778X. doi: 10.1109/TEVC.2013.2281535.

[52] Kalyanmoy Deb and Deb Kalyanmoy. *Multi-Objective Optimization Using Evolutionary Algorithms.* John Wiley &amp; Sons, Inc., USA, 2001. ISBN 047187339X.

[53] S. C. Chiam, K. C. Tan, and A. Al Mamum. Evolutionary multi-objective portfolio optimization in practical context. *International Journal of Automation and Computing*, 5(1):67–80, Jan 2008. ISSN 1751-8520. doi: 10.1007/s11633-008-0067-2. URL `https://doi.org/10.1007/s11633-008-0067-2`.

[54] Sergey Sarykalin, Gaia Serraino, and Stan Uryasev. *Value-at-Risk vs. Conditional Value-at-Risk in Risk Management and Optimization*, chapter Chapter 13, pages 270–294. Informs, 2008. doi: 10.1287/educ.1080.0052. URL `https://pubsonline.informs.org/doi/abs/10.1287/educ.1080.0052`.

[55] Node.js Development Team. *`Node.js` is a JavaScript runtime built on Chrome's V8 JavaScript engine.* OpenJS Foundation, San Francisco, CA, USA, 2021. URL `https://nodejs.org`.

[56] React.js Development Team. *`React.js` A JavaScript library for building user interfaces.* Facebook Open Source, Menlo Park, CA, USA, 2021. URL `https://reactjs.org`.

[57] Deqing Li, Honghui Mei, Yi Shen, Shuang Su, Wenli Zhang, Junting Wang, Ming Zu, and Wei Chen. Echarts: A declarative framework for rapid construction of web-based visualization. *Visual Informatics*, 2(2):136–146, 2018. ISSN 2468-502X. doi: https://doi.org/10.1016/j.visinf.2018.04.011. URL `https://www.sciencedirect.com/science/article/pii/S2468502X18300068`.

[58] V. I. Koltchinskii. M-estimation, convexity and quantiles. *The Annals of Statistics*, 25 (2):435–477, 1997. ISSN 0090-5364.

[59] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb. 3d-radvis: Visualization of pareto front in many-objective optimization. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 736–745, July 2016. doi: 10.1109/CEC.2016.7743865.

[60] K. Deb and M. Abouhawwash. An optimality theory-based proximity measure for set-based multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 20(4):515–528, August 2016. ISSN 1089-778X. doi: 10.1109/TEVC.2015.2483590.