PaletteViz: Understanding a High-Dimensional Pareto-Optimal Data-Set for Multi-Criteria Decision Making

AKM Khaled Ahsan Talukder and Kalyanmoy Deb, Fellow, IEEE

COIN Report Number 2018013

Abstract—In order to solve multi/many-objective optimization problems, a large number of efficient algorithms have been proposed. The final outcome of these algorithms is a set of trade-off solutions that span the entire Pareto-optimal front. A subsequent decision-making task must analyze these Pareto-optimal solutions to choose a single preferred solution. This is where an efficient visualization technique makes a significant role in assisting the decision-makers (DM) to understand the high-dimensional Pareto-optimal solutions in terms of relevant information, such as, local and global shapes of the Pareto-optimal surface, spatial distance of one solution to another, a solution’s trade-off among conflicting objectives in its neighborhood, closeness of a solution to constraint boundary, and others. Two and three-dimensional Pareto-optimal fronts are trivial to visualize and allow all the above analysis to be done in a comprehensive manner. However, for four or more objectives, a suitable visualization method that can extract such information is still not available, and innovative methods are long overdue. In this paper, we propose a novel way to map a high-dimensional Pareto-optimal data-set into two-and-half dimensions with features that may be of great interest to DMs. As a proof-of-principle demonstration, we apply our proposed palette visualization (PaletteViz) technique to a number of different structures of Pareto-optimal data-sets and discuss how the proposed technique is different from a few popularly used visualization techniques and how it may be useful in a multi-criteria decision making tasks.

Index Terms—Many-objective Optimization, High-dimensional data-set, Visualization, Radviz, Parallel coordinate plot, Decision-making.

I. INTRODUCTION

With the current advancement in many-objective optimization algorithms for solving problems with four or more conflicting objective functions, an array of post-optimization issues must now be addressed on an immediate basis, especially if these algorithms are to be regularly used in practice. One of the most challenging issues is to visually comprehend the obtained non-dominated solutions produced by a many-objective optimization solver, so that the most preferred solutions can be filtered out in a natural way by the decision-makers (DMs). If the optimization problem is consisted of two or three objectives, a two or three-dimensional scatter plot is the most intuitive way to illustrate the obtained results. In such a case, 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 solutions). DMs can also visualize other preference criteria in lower dimensions, such as, robustness or reliability of a solution, or some other utility functions representing the practical usefulness of a solution.

A good visualization technique should enable a graphical exploration of the objective space and should help the DM to obtain better insights into the problem and to compare different alternative Pareto-optimal solutions functionally. Such visual comparisons must go beyond their dominance relationships to each other, but include various functional relationships, such as, the distance to constraint boundaries, relative trade-offs among their neighboring solutions, the distance from the boundary of the Pareto-optimal front etc. Moreover, the graphics must be clear as well as trivial to comprehend, interpret, and manipulate by the DM. In order to understand the entire high-dimensional Pareto-optimal front in such manner, the visualization mechanism should also preserve the structural properties of the original high-dimensional data-set as accurately as possible. While DMs may be interested in such a visualization mainly to filter a few critical solutions for further considerations, a visualization technique may also take into account DM’s preferences more directly, such as specification of a reference or aspiration point [1], [2], relative importance of one objective over the other [3], [4], or other utility-based information [5], [6].

While new procedures and advancements have still being made in multi-criterion decision analysis (MCDA) for various ways of analyzing large-dimensional and non-dominated data-set since early seventies, the evolutionary multi-criterion optimization (EMO) community has been mainly concentrating on developing efficient algorithms for searching high-dimensional and non-dominated solutions through computing algorithms. Besides the practical significance of developing an efficient and convenient visualization technique for large-dimensional data-set, EMO researchers (and for this matter all computational intelligence (CI) researchers interested in multi-objective problem solving tasks) wonder if their algorithmic acumen can be utilized somehow in coming up with effective visualization techniques. Our paper makes an attempt in this direction and suggests a visualization technique which may perform a post-optimal phase of computations by unearthing geometric and probable decision-making properties of a large, non-dominated data-set, before approaching the DMs. The hope is to reduce the large cardinality of solutions by filtering...
through a certain 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.

There exist a number of different visualization techniques in the literature and are widely used in EMO and MCDA domains: (i) parallel coordinate plot (PCP) [7], radial visualization or Radviz [8], multiple scatter plots [5], heatmap [9], self-organizing maps [10], and others. A number of recent studies [11], [12] 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 any of the geometric and structural, functional and decision-making properties mentioned above clearly to a DM. Due to lack any suitable visualization technique, these existing methods are used, but it has been always questionable how DMs are able to utilize these methods in any effective and convenient way to choose a preferred solution.

In this paper, we propose a visualization technique – a palette visualization (PaletteViz) technique – which allows a DM to look at a large number of non-dominated data-set from the three different perspectives mentioned above. The idea of PaletteViz is to present the original data-set by maintaining their geometric structures as much as possible and by clearly identifying points that are functionally and preferentially appealing to a DM. The PaletteViz technique is demonstrated on a number of different Pareto-optimal data-sets to show its usefulness. Wherever necessary, the plots are also compared with other existing visualization methods.

The paper is organized as follows. In Section II, we discuss a foundational aspects of understanding the shape of a high-dimensional data-set. In Section III, we discuss a number of key decision-making considerations which DMs may be interested while visualizing a set of trade-off solutions. Then, in Section IV, we describe our proposed PaletteViz technique in details. In Section V, we present PaletteViz plots on a number of Pareto-optimal front scenarios and compare with a few existing visualization techniques. We conclude the paper in Section VII with some future directions of our current work. The appendix presents the description of the many-objective optimization problems used in this study.

II. SHAPE OF HIGH-DIMENSIONAL DATA-SETS

Data visualization for many-objective decision-making demands certain properties of the ensuing visualization technique, which may be different from other high-dimensional visualization tasks, including data analytics, multivariate statistics, and machine learning. Before going into the detail, let us define the concept of data visualization for our purpose:

**Definition II.1.** (Visualization) Given a set of \( n \), \( m \)-dimensional data-set \( C \subset \mathbb{R}^n \), a visualization is a transformation function \( D \leftarrow V(C) \) that maps \( C \) onto \( D \), where \( D \subset \mathbb{R}^2 \).

The reason for such a two-dimensional mapping of higher-dimensional data-set is for an easier analysis and understanding of the interactions of the points. However, a desirable mapping procedure should be able to make a visualization to accurately portray the high-dimensional data points. Therefore, we are looking for an ideal visualization procedure for which the transformation function maintains certain key properties of the high-dimensional data-set to the two-dimensional mapped space.

A. Point Depth Ranking

The notion of data depth or point depth is frequently encountered in non-parametric multivariate data analysis [13]. This concept provides center-outward orderings of points in the Euclidean space of any dimension and leads to a novel non-parametric multivariate statistical analysis method where no assumption regarding the point cloud distribution is needed. A point depth measures how deep (or central) a given point \(^1\) \( f \in C \) is relative to the entire data-set \( C \).

In our proposed visualization method, we use the concept of the ‘non-convex hull peeling depth’, [14], which, at a point \( f \) with respect to the entire data-set \( C \), is simply the level of the non-convex layer that \( f \) belongs to. A non-convex layer is defined as follows. Construct the smallest non-convex hull which encloses all points in \( C \). 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 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 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 \( \alpha \)-shapes [15]. We discuss the \( \alpha \)-shape construction process in Subsection IV-A in more details. Henceforth, \( D_C(f) \) is used to indicate non-convex layer depth of point \( f \) in \( C \), unless otherwise specified. A larger \( D_C(.) \) always implies a deeper (or more central) \( f \) in \( C \). All the points that belong to the same depth form a depth-contour and such depth-contours are the basic cue to understand the shape of a high-dimensional point cloud [16]. Our proposed visualization technique is based on the depth-contour measure of a data-set.

B. Neighborhood Relationships

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

\(^1\) In this paper, we denote each data point with \( f \) instead of a more standard notation \( x \). Since in our case, each data point \( f = (f_1,f_2,\ldots,f_m) \) is a vector of \( m \) multi-objective function values in \( \mathbb{R}^m \).
same neighborhood relationship when they have been mapped onto the two-dimensional space. More formally, we can define this condition with the notion of hyper-sphere neighborhood. Assuming a function $D \subset \mathbb{R}^2 \leftarrow V(C \subset \mathbb{R}^m)$, if two points $\{p, q\} \in C$ are within a hyper-sphere of radius $r_C$, then $\{r, s\} \in \mathcal{D}$ should be within a circle of radius $r_C$, where $\{r, s\} \leftarrow V(\{p, q\})$, assuming all points in both $C$ and $\mathcal{D}$ are normalized within $[0, 1]$.

This condition of neighborhood is another basic requirement for an ideal visualization technique. We need 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 [17] 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 inside in the two-dimensional shape at the cost of minimizing the error in the neighborhood relationship [18]. For example, if use t-SNE (t-Stochastic Neighborhood Embedding) [19] to visualize a three-dimensional Pareto-optimal data-set for a vehicle crashworthiness Problem [20] we observe that two different runs generate two different visualization and the center-outward relationship among the points are 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 1. However, there are other deterministic methods that are able to map high-dimensional data onto two-dimensional space with maintaining neighborhood relationship with minimal surjective mapping. Radial visualization (Radviz) technique [21] is one such method, which we describe next.

**C. Radviz Method**

In Radviz method, the mapping procedure of points from $m$-dimensional space into 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 $f = (f_1, f_2, \ldots, f_m)$ is placed inside the circle at point $u = (u_1, u_2)$ by assuming that each anchor, representing an objective $f_i$, connects point $u$ with an imaginary spring having a stiffness $f_i$. The location of $u$ is determined by the equilibrium of $m$ spring force vectors. A little calculation will reveal the position of $(u_1, u_2)$, as follows:

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

**III. Visualization of a Pareto-Optimal Data-Set from a Decision Maker’s Perspective**

Most existing visualization methods used in EMO studies today are borrowed from the high-dimensional data visualization literature. From our previous experience with industry collaborations, we have observed that a decision making procedure requires a few unique considerations which the usual high-dimensional data visualization methods may not have found to be important. Thus, it is time that EMO researchers develop new and more meaningful visualization techniques rather than simply following existing methods from the literature. Therefore, the next important question is what are these unique properties that a multi-objective data visualization technique must have so that decision-makers can be presented with only a few meaningful and relevant Pareto-optimal solutions for a faster and worthy decision analysis.

**A. Desired Properties of Pareto-optimal Solutions**

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

1) **Points with Large Trade-off and Knee Points:** Perhaps, the most desirable aspect of Pareto-optimal solutions is the points that offer most trade-off among all other points in a solution set [22]. 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. If the trade-off is defined as the ratio of the maximum loss among objectives in moving to a neighboring solution to the minimum gain among objectives in making the move, then a point having a large trade-off means that gain in moving to a neighbor is small compared to the loss in 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. It is clear from the above discussion that the trade-off of every Pareto-optimal 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 be set by DMs and the Pareto-optimal points which exceed the threshold can be presented clearly in a Pareto-set visualization technique so that DMs can easily recognize these points from hundreds of other Pareto-optimal points for further considerations.

In this paper, we follow the definition of knee points discussed in [23]. Although other definitions may also be used. Let us consider the simple Pareto-optimal front depicted in Figure 2, with three objectives to be minimized. This 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 likely, the solutions at the knee are most likely to be a good choice to the DM. Note that in Figure 2, due to the concavity at the edges, similar reasoning holds for the extreme solutions, which is why these extreme solutions should also be considered as knee-like points.

2) **Boundary Points:** In most practical problems, the Pareto-optimal set is usually bounded in the objective space. In some
occasions, there are disjointed Pareto-optimal sets, each is 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 of a set means that immediate to one side of the point there does not exist any other non-dominated point, making the point special in some sense. A DM would be interested in knowing the variable vector and the solution of such boundary points which lie on the edge of separating non-dominated points from dominated points. This aspect can be of importance in choosing a final preferred solution.

3) Active Points: Boundary points may also lie on the intersection of several constraint surfaces in constrained multi-objective optimization problems. Thus, in addition to boundary points, a DM may be interested in knowing Pareto-optimal points that are active (lying on one or more constraint surfaces) and in contrast the points that are 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 Pareto-optimal front and being un-reliable due to uncertainties of the problem. In any case, such active points must be identified in a Pareto-optimal set and represented through a visualization technique in a clear manner – being close to a constraint surface or far away from it. This will allow DMs to focus on such points for further analysis.

4) Isolated Points: In statistical data analysis studies, isolated or outlier points in a data-set are usually neglected and more focus is usually put on the remaining well-behaved data points. In multi-objective Pareto-optimal data-set analysis, isolated points are special and important, as they signify a unique and non-common combination of objectives that make a different trade-off compared to that in the bulk of the other Pareto-optimal solutions. A DM should be interested in knowing these special outlier points, as they may dictate 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.

IV. PROPOSED PALETTE VISUALIZATION METHOD

The proposed method, we call a Palette Visualization technique, mimics an artist’s palette which is a two-dimensional plate and has regions earmarked for 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 proximity of points to each other. Of course, when high-dimensional data is compressed into a lower dimension,
there is always loss of information. However, our technique, as will be revealed soon, focuses on not losing information for the preferred Pareto-optimal points and does not care much if the information is lost on non-preferred points.

First, we need to map high-dimensional data-set into a two-dimensional space. Various methods exist in the literature for this purpose, but here we use the Radviz method [21] due to its simplicity and popular use in EMO studies. Second, we need a method to maintain the structure and neighborhood information of 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-optimal data-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 central points according to their preference. Third, the points are marked in different sizes and colors by providing special emphasis to the DM-preferred points so that they become visually and clearly apparent to DMs.

1) Construction of depth contours and assigning the depth-equivalence class to each of the data points in $C \subset \mathbb{R}^m$.

2) Represent each depth contours in a two-dimensional palette mapping so that the neighborhood relationships and center-outward orderings are maintained as accurately as possible.

A. Finding Depth Contours Using $\alpha$-Hull Method

In order to find the depth contours of the data points in $C \subset \mathbb{R}^m$, we have utilized the notion of non-convex hull pealing depth using an algorithm called ‘$\alpha$-shape’ [15] [25]. Since the $\alpha$-shape algorithm can be slow in higher dimension, we have modified the algorithm to make it faster with a penalty for reduced accuracy. This is due to the fact that we do not need a most accurate enumeration of depth contours and some errors are tolerable, as will be clear later.

Conceptually, $\alpha$-shapes are a generalization of the convex hull of a point set. Given $C \subset \mathbb{R}^m$ and $\alpha$, a real number with $0 \leq \alpha \leq \infty$, the $\alpha$-shape of $C$ is a polytope that is neither necessarily convex nor necessarily connected. For $\alpha = \infty$, the $\alpha$-shape is identical to the convex hull of $C$. However, as $\alpha$ decreases, the $\alpha$-shape 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 $\alpha$ becomes small enough so that a sphere with radius $\alpha$, or several such spheres, can occupy its space without enclosing any of the points of $C$. We can think of $\mathbb{R}^m$ filled with styrofoam and the points of $C$ are made of more solid material, such as rock. Now imagine a spherical eraser with radius $\alpha$. It is omnipresent in the sense that it carves out styrofoam at all positions where it does not enclose any of the sprinkled rocks, i.e., points of $C$. The resulting object is called the $\alpha$-hull.

As we have mentioned before, in our case, very accurate depth contour enumeration is not necessary since if the number of points are large, then there will be a large number of depth contours and a DM might not be interested in examining each and every contour separately. Therefore, we simplify the $\alpha$-shape algorithm to approximate the non-convex hull pealing depth that is good enough for a human to visualize the high-dimensional Pareto-optimal data-set. However, an exact $\alpha$-shape algorithm can also be used without any loss in visualization.

The $\alpha$-shape algorithm starts with a Delaunay triangulation of the high-dimensional data points. Then the algorithm decides which edges need to be removed to get the final non-convex hull. In the case of exact $\alpha$-shape, the filtration (i.e. removal of edges in a Delaunay triangulation) is done according to all the simplices ranging from 0 to $m - 1$ dimensions. However we avoid such loop, instead, we assume that filtration by the removal of longest edge in the convex-hull should give an approximate non-convex hull shape. Therefore, we set $\alpha$ to the length of the shortest convex-hull edge. Also During the filtration phase of the $\alpha$-shape algorithm, we need to find the circum-radius of the hypersphere of each simplices, this can be achieved using the formula described in [26]. Now if the circum-radius is greater than $\alpha$, then we exclude the corresponding edges of the simplex. Once the edges are removed from the simplex, we need to identify the vertices that reside on the shape boundary. This has been done in Algorithm 1.

Algorithm 1 counts the number of simplices that include each edge and if the count is less than or equal to $(m - 1)$, then the edge belongs to the boundary of $C$ and returns these data points. The next step is to represent the data points according to their depth contours and utilize a specific kind of mapping that respects the neighborhood and spatial navigability condition.

B. Representing Points in Two Dimensions

We know the depth contour and depth equivalence class, we can consider the points within a same depth equivalence class as a ‘layer’ of the high-dimensional data-set. The $\alpha$-shape algorithm basically decomposes the point cloud in a layer-wise manner. Once this decomposition is performed, points from each layer are represented in a two-dimensional Radviz
plot [21]. Although we have used Radviz, other mapping techniques like SNE [17], [19], [27] 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 Radviz plot of each layer. The top-most two-dimensional Radviz plot corresponds to the boundary layer, the next one below corresponds to the next to boundary layer and so on. The bottom-most Radviz plot correspond to the inner-most points (close to the central region with 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 the Figure 3.

C. Visualization of Pareto-optimal Data-set

With a layer-wise plot of the points through two-dimensional Radviz plots, DMs have the centrality information of points. In addition, DMs would now be interested in getting more information about three different properties of the data-set: (i) geometric structure, (ii) functional structure, and (iii) preferential structure. For this purpose, we use different color schemes and marker sizes to bring our different structures in the Figure 3.

- Boundary to Centroid: In addition to place the points in different layers based on the $\alpha$-shape method, points that are far away from the centroid of the data-set are painted as dark-blue and as we go to 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, Euclidean distance metric is used to measure the distance from centroid. In some scenarios, some boundary points may be close to the centroid due to shape and density of data-set and the DMs have another visual mean to identify the closeness of a point to the centroid in addition it being close to the boundary. See Figure 2.

- Constraint Boundary: In the presence of constraints, the closeness of a point to any constraint boundary may be of importance to the DM. Hence, points that are close to a constraint boundary is painted in pink color and points well inside the feasible region are colored in cyan. Of course, it is expected that the Pareto-optimal set will not have any infeasible solution. See Figure 8.

- Large Trade-off: Points with large trade-off values are marked in dark-red circle. A point having a trade-off $\theta$ value larger than $\mu_\theta + 3\sigma_\theta$ of the entire data-set is shown in dark-red color. The points with a larger $\theta$ than the above threshold are marked with a bigger marker in dark-red color. See Figure 2. For the calculation of trade-off value $\theta$ for a point, we follow the method presented in [23]. Thus, a palette visualization plot or, a PaletteViz plot, will have multiple layers of two-dimensional Radviz plots with non-dominated points painted with either item 1 or item 2 above, and then a large trade-off point (item 3), if any, is 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.

D. Number of Layers

The choice of 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 a point. It is interesting to note that the division of layers can also be made based on distance to the centroid or using any other metrics that is important for the DM to visualize the entire data-set in a hierarchical manner. Our methodology does not exclude any of such possibilities.

V. 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. The problem descriptions are given in the Appendix. For the ease of visualization, we group the original layers from the high-dimensional space in PaletteViz plots into three to four layers.

A. Pareto-optimal Data-set with Knees

To demonstrate the working of the proposed PaletteViz concept, first, we use a benchmark test problem from [24], called DEBMDK. The problem has a parameter that can be changed to introduce multiple knees, however we choose the problem with a single knee region for an easier demonstration of our proposed palette visualization technique. Results for three, and eight-dimensional DEBMDK problems are presented in Figures 4 and 6, respectively.

In the Figure 4, the bottom layer corresponds to the deepest points in the original dimension. A few knee solutions (in dark-red color) in this layer are clearly visible. Three other
Next we compare our palette visualization plot with an existing visualization method – the parallel coordinate plot (PCP) [30]. Figure 5 shows all points and marks three extreme points with a large trade-off in blue color and the knee points in red color. The PCP plot is unable to indicate any small or large trade-off information. They do not appear to be any different from other points. The comparison clearly demonstrates that palette visualization conveys a great amount of information about the key points of the Pareto-optimal data-set than that by PCP.

Next we apply the palette visualization to eight objective DEBM7DK problem in Figure 6. The knee points in the bottom-most layer are clearly identifiable from the PaletteViz plots. Existence of a large trade-off at a few extreme points are also evident from the plots. Again for further comparison, we visualize the eight-dimensional problem using the original Radviz plot in Figure 7. The Radviz plot does not show any boundary-core relationship among the points. Also, all points seem to cluster around the central part of the Radviz’s circle, thereby not making much use of the plot to DMs.

B. Pareto-optimal Data-set with an Isolated Region

Here we are interested to see if the proposed palette visualization method can address the topological features of
the high-dimensional point cloud. In order to simulate this condition, we modify the DTLZ2 problem [31] by slicing the Pareto-optimal front using two constraint functions in such a way that two clusters are created. A three-dimensional scatter plot is shown in Figure 8. One cluster has many 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 clearly identifying at least one solution from this region, as such a point comes with best choices on $f_1$ and $f_2$. The corresponding palette visualization for three, and eight-objective cases are presented in Figures 9 and 10, respectively. Figures show two clearly distinct regions, one having a few points and the other having majority of points. 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 no large trade-off point in bottom layers indicate 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 adequate number of neighboring points making a large trade-off value for them.

Besides all the above information, filtering the total number of non-dominated points from 4,004 to a handful of large trade-off points and a few isolated points is a tremendous advantage derived from the PaletteViz plots.

C. Pareto-optimal Front having Differing Dimensions

In many real-world problems, a Pareto-optimal front can be consisted of clusters of data points having different dimen-
Fig. 10. PaletteViz plot for the isolated Pareto-optimal data-set in eight dimensions. The isolated region near the $f_8$ anchor (i.e. maximum of $f_8$ in the original dimension) is clearly identifiable. There are 4,004 points in this data-set.

Fig. 11. Three-objective Pareto-optimal data-set (1,044 points) for the DTLZ8 problem. The Pareto-optimal data-set is consisted of two geometrically different point clouds – a straight line is connected with an $m-1$-dimensional hyper-plane.

D. Disjointed Pareto-optimal Front

As we have discussed before, in the presence of a Pareto-optimal front having multiple disjointed 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 consider DTLZ8 problem [31], where the Pareto-optimal front is consisted of a $(m-1)$-dimensional hyper-plane and a one-dimensional straight line. The corresponding three-dimensional Pareto-optimal front is presented in Figure 11. The palette visualization a four-objective version of the problem is presented in Figure 12.

It can be clearly seen that the PaletteViz plots maintain a similar geometrical structure of the data points as they are in the original high-dimensional objective space. Besides the geometrical structure, some boundary points having a large trade-off in the hyper-plane are also observed.

Fig. 12. PaletteViz plot of the four-objective DTLZ8 Pareto-optimal data-set containing 1,292 points. The plot clearly shows that there is a straight line and a point cloud in the original space.

Fig. 13. Pareto-optimal data-set for the three-objective C2DTLZ2 problem. There are four isolated regions on the data-set. The data-set contains 1,717 points.

Fig. 14. PaletteViz plot for 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 colors, providing a 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
A. Vehicle Crash-worthiness Problem

We apply our method to the vehicle crash-worthiness problem from [20]. This is a three-objective optimization problem, however the Pareto-optimal front in this problem is composed of two separate clusters, as shown in Figure 1a. Our proposed palette visualization (Figure 16) can clearly show two clusters independently. The palette visualization of the top cluster (cluster A) is presented in Figure 16a and that of the bottom cluster (cluster B) is presented in Figure 16b. Points are

one of the bottom layer 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 [33], [9] visualization in Figure 15. Heatmap is one of the most widely used techniques for high-dimensional data visualization, however for this particular application, it does not provide much information about six clusters that the data possess and nearness of any point to a constraint boundary.

VI. PALETTE VISUALIZATION 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. However, the features were so far demonstrated on different test problems. In this section, we apply PaletteViz technique on two engineering design problems.

B. 10-dimensional General Aviation Aircraft (GAA) Design Problem

Finally, we apply our proposed palette visualization technique on an engineering design problem having 10 objectives and 16 constraints [34]. Figure 17 clearly shows that there are two disjointed clusters in the original ten-dimensional Pareto-optimal front, 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 18. 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 nine-dimensional front representation. Finding more points near each of these large trade-off points using reference-point based EMO methods [2], [35] may be a way to estimate the real trade-off of these points. But, even with a small density of points, the DM can start analyzing the large trade-off points in one of the two clusters, depending on whether the DM prefers solutions close to constraint surfaces or well inside the feasible region. The PaletteViz plot can also be redrawn with green-blue color coding for evaluating the centrality status of each point as an additional information. All these functionalities will be helpful for the DM to get a good idea of the properties of the points before choosing a final preferred solution.

There is another information of this problem worth pointing here. Despite the points being normalized before making the PaletteViz plot, the density of the data-set seems to concentrated towards relatively large values of $f_5$ to $f_7$ and small values of $f_1$ to $f_3$ and $f_9$ and $f_{10}$.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated a new visualization approach – PaletteViz – to address the issue of high-dimensional Pareto-optimal data-set visualization. High-dimensional and non-dominated objective vectors have been mapped into several two-dimensional Radviz plots and stacked according their boundary to central location in their original high-dimensional space. Thus, the proposed PaletteViz technique has used a two-and-half dimensional mapping of the entire data-set. Unlike the existing visualization methods, PaletteViz technique uses different color and maker sizing schemes to filter and highlights a few critical points which would be of further interest to decision-makers (DMs). In this paper, the criticality of a point has been defined from three different considerations: (i) geometric criticality, (ii) functional criticality, and (iii) preferential criticality of points. Geometric criticality has been defined for points being in clusters, close to boundary, close to centrality, and dimensional degeneracy. The functional criticality has been defined for points being close to constraint boundary. The preferential criticality has been defined for points having a large trade-off (large sacrifice to gain ratio for moving to neighboring points). Each of these three types of criticality can be extended to include other metrics.

The systematic filtering of solutions into clusters in the original high-dimensional space, then into layers according to their centrality within each cluster and finally into different levels of trade-off values allow a decision-maker to look at only a handful of critical solutions for an assisted decision-making task. On many shapes and features of Pareto-optimal and non-dominated data-sets from test problems and two engineering design problems and containing hundreds to thousands of points, PaletteViz is able to identify a handful of critical points, which would be of further interest to DMs.

This proof-of-principle study and its associated results can be extended with modifications of each step. The Radviz mapping can be changed with other two-dimensional mappings, such as t-SNE or other structure-preserving methods. The division of points into several layers can be changed with other point depth metrics instead of the non-convex hull peeling depth used in this study. It would be an interesting study to investigate if other types of depth metrics can provide a faster and more useful visualization. Moreover, there are some other multivariate depth ranking mechanisms which preserve the directions of data points [36], which can also be tried. A recent study [37] used a three-dimensional Radviz plot with the third-axis representing distance of a point from a hyper-plane, clearly demarcating convexity property of points. Layers can also be made from such 3D Radviz plots. To extend functional criticality based identification, the use of KKT proximity measure (KKTPM) [38] which is capable of quantifying the proximity of a non-dominated point to the true Pareto-optimal front would be a natural choice.

With all these different visualization possibilities, it is now worth developing a user-interactive software which will...
allow DMs to make multiple PaletteViz plots with different combinations of subdivision and classification metrics applied to the same data-set before the nature and properties of critical points are well understood. Such a method will enable a proper visualization of multi-criterion data-set before a single or a handful of solutions are selected. Finally, the proposed PaletteViz technique should also be tried for high-dimensional data visualization purposes in other scientific domains, specially to machine learning and data-mining communities.

REFERENCES


