PaletteStarViz: A Visualization Method for Multi-criteria Decision Making from High-dimensional Pareto-optimal Front

AKM KHALED AHSAN TALUKDER, Computational Optimization and Innovation Laboratory
KALYANMOY DEB, Computational Optimization and Innovation Laboratory

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 causes a large cognitive burden on the part of 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 analytic domains and they are too generic to be effective for many-criteria decision making. In this paper, we propose a $\mathbb{R}^2$-dimensional visualization method, following an earlier concept, using star-coordinate plots for effectively visualizing many-objective trade-off solutions. The proposed PaletteStarViz respects some basic topological, geometric, and functional decision-making properties of high-dimensional trade-off points mapped to $\mathbb{R}^2$-dimensional space. We demonstrate the use of PaletteStarViz to a number large-dimensional test problems and a 10-objective real-world problem. The use of ‘Pareto Race’ concept from MCDM literature is introduced within PaletteStarViz to demonstrate the ease and advantage of using this and MCDM concepts for choosing key preferred points from a large-dimensional trade-off set.

Additional Key Words and Phrases: Many-objective Optimization, High-dimensional Pareto-optimal front, Visualization, Multi-criteria Decision Making.

ACM Reference Format:

1 INTRODUCTION

With the current development in many-objective optimization (MOP) algorithms to solve problems with three or more conflicting objective functions, an array of posteriori issues must now be addressed on an immediate basis, especially, if such algorithms are to be regularly applied in practice. One of the most pressing issues is to visually comprehend the obtained non-dominated points in the objective space produced by an MOP solver, so that the a few critical data points can be filtered out by the decision-makers (DMs) in a faster way. 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-optimal front. In such case, DMs can easily locate and isolate critical data points\(^1\) that correspond to the most beneficial trade-offs among objective function values. DMs can also visualize other preference criteria in lower dimensions, such as, robustness or reliability of a Pareto-optimal solution in the objective space.

\(^{1}\)In this paper, all mentions of ‘point’, ‘data point’, ‘data’ and ‘data-set’ refer to the data points in the objective function space, not in the design variable space.

Authors’ addresses: AKM Khaled Ahsan Talukder, Computational Optimization and Innovation Laboratory, Michigan State University, East Lansing, Michigan, 48824, talukde1@msu.edu; Kalyanmoy Deb, Computational Optimization and Innovation Laboratory, Michigan State University, East Lansing, Michigan, 48824, kdeb@egr.msu.edu.
There exist a number of different visualization techniques in the literature for generic data analytics. Evolutionary Multi-objective Optimization (EMO) and Multi-criteria Decision Analysis (MCDA) researchers have simply borrowed them in visualizing multi-dimensional trade-off data [1], [2]. While most of these methods allow high-dimensional data points to be rendered in a two-dimensional space, they are not able to convey any of the geometric and structural, functional, and decision-making properties in a clear way that a DM would be interested in. Due to the lack of any suitable visualization technique, these existing methods are still predominantly used, but it has been always questionable how DMs are able to utilize such methods in an effective way to choose a preferred point from the Pareto-optimal front.

In this paper, we propose a visualization framework – a palette visualization (PaletteStarViz) technique – which allows a DM to look at a non-dominated data points in a functionally decomposable manner so that a few critical and preferential points can be isolated for decision making and analysis. The technique resembles an artist’s palette – where similar colors emerge with respect to the local arrangement of points in the high-dimensional space, as if the neighboring points in high-dimensional space are mapped closer together on a functionally decomposed layers of two-dimensional palettes. The PaletteStarViz technique is demonstrated on a number of different Pareto-optimal fronts to show its usefulness. Wherever they are relevant, the plots are also compared with a few other widely used high-dimensional Pareto-optimal front visualization techniques.

The paper is organized as follows. In Section 2, we discuss fundamental motivation behind this study. In Section 3, we describe a number of key decision-making considerations in which DMs might be interested while visualizing a set of Pareto-optimal points. Then, in Section 4, we describe our proposed PaletteStarViz technique in detail. In Section 5, we present PaletteStarViz plots on a number of Pareto-optimal front scenarios and compare with a few existing visualization techniques. In the next section, we also demonstrate how the proposed method can also be used Pareto-race based solution exploration. We also show that our methods can be used to comparative analysis EMO algorithms. We conclude the paper in Section 7 with some future directions of our current work.

2 MOTIVATION

Let us start our discussion by posing a simple scenario having a set of regularly spaced data points in a three-dimensional cube, as illustrated in Figure 1a. Since the data points are laid on a regular grid, we can assume that the cube is composed of three smaller nested cubes. A point of interest, marked in red, is located in the intermediate cube, as shown in Figure 1b. A user may be interested in understanding the location of this point using a visualization method shown in a two-dimensional space. We can flatten each nested cube to see an arrangement presented in Figure 1c. Here, the bottom layer corresponds to all the points on the boundary of the cube, the middle layer corresponds to all the points on the intermediate cube and so on. The flattened figure shows that the point of interest lies in an intermediate layer close to the core of the object.

Using a standard method of visualizing high-dimensional data points, the points can be shown in a Parallel Coordinate Plot (PCP) [3]. In PCP, 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. All points are shown in the plot including the red point. However, PCP does not provide any information the red point’s location relative to the entirety of the points. However, if the geometry of point set is known before hand, a PCP can capture some aspects of the structural property of the high-dimensional manifold [4]. A major limitation of PCP is that it generates different visualization 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 could be to arrange the data points in a matrix and color each cell according to their values (i.e. coordinate positions). This representation is known as Heat Map [5].
(a) Points arranged in a nested cube. (b) Unnested cubes. (c) Each cube can be flattened. (d) PaletteStarViz representation of the cube.

Fig. 1. (a) An example three-dimensional data points organized in three nested cubes on a regular grid. (b) The “point of interest” (i.e. point marked in red) 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. (d) PaletteStarViz representation of the cube. The plot indicates that the red point is an interior point (middle layer), but not at the core (bottom-most layer). Also has a relatively smaller $f_3$ value.

Fig. 2. (a) PCP visualization of the cube in Figure 1a. (b) Heatmap visualization of the same cube. The red line represents the "point of interest". (a) Star-coordinate visualization of the same cube in Figure 1a. (b) t-SNE plot of the cube in Figure 1a.

An example Heat Map visualization of the cube data-set is presented in Figure 2b. However, it is not possible to infer anything significant about the structure of point in its original data set.

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 a Star-coordinate Plot or SC-plot [6]. In this method, the mapping 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 $\mathbf{f} = (f_1, f_2, \ldots, f_m)$ is placed inside the circle at point $\mathbf{u} = (u_1, u_2)$. The position of $(u_1, u_2)$ is calculated as follows:

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

(1)

In this paper, 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 nice property of SC-plot is that it can capture the shape of the original point-set as one of its valid projections. An example SC-plot of the cube data set is presented in Figure 2c. In terms of neighborhood relationship among the data points, SC-plot provides less number of non-bijective mapping of the points than that of other projection methods, such as Radviz. A detailed analysis of this aspect of Radviz and SC-plot can be found in [7].

There are another class of visualization methods that employ manifold learning and embedding of the latent variables in the data points and use that information to map the points on a lower dimensional space. For example, recently proposed t-Stochastic Neighborhood Embedding (t-SNE) [8] is one such technique. If we apply this method, the resultant visualization looks like the plot in Figure 2d. t-SNE can maintain neighborhood relationships in a local region, however it does not preserve the global structure of the data points. In Figure 2d, the core points are distributed all over the plane. Also, the relative position of the “point of interest” with respect to the entire data-set does not conform to its position in the original high-dimensional space.

3 REQUIREMENTS FOR PARETO-OPTIMAL FRONT VISUAL ANALYTICS

So far, EMO literature has shown lukewarm interest in devising efficient methods for choosing a single preferred point from the Pareto-optimal front. This is probably due to the subjective, often non-analytic, considerations associated with the task, whereas most EMO researchers are more interested in computational aspect of the multi-objective problem solving.

3.1 Points with Large Trade-off

Perhaps, the most practical and desirable aspect of a choosing a single Pareto-optimal solution is the trade-off information associated with each solution in the objective space [9] [10]. A point having a large trade-off means the gain achieved in a certain objective by choosing a neighboring point is small compared to the loss in other objective values. Thus, there is not much motivation to select Pareto-optimal points other than the ones with large trade-off values. As the loss outweighs the gain, the point with a large trade-off value is most desired to the decision-makers. The points/regions of the Pareto-optimal front with comparatively bigger (or better) trade-off are known as knee points. In this paper, we follow the analytical definition of knee points discussed in [9]. An example of a Pareto-front with one knee region is presented in Figure 3. This Pareto-front is found from solving a three-objective DEB3DK problem [10]. The knee points are presented with a red circle and data points with comparatively better trade-off are presented with circles of increasing radii. The color-coding (the gradient of light-green to dark-blue) corresponds to the distance of each point from the center of the entire data-set.

3.2 Boundary, Active and Isolated Points

In most practical problems, the Pareto-optimal front is bounded in the objective space. In some occasions, there are disjoint Pareto-optimal fronts where each of them is bounded. A boundary point [11] of a set (or manifold) asserts that there are no non-dominated point exist on the other side of the manifold, making the solution special in many cases. Therefore, a DM would be naturally interested in knowing the design variable vector of such boundary points.

In addition to boundary points, a DM may be interested in knowing Pareto-optimal points that are almost active (lying close to one or more constraint surfaces) and in contrast to the points that are far away from any constraint surface. A point lying on or very close to a constraint surface is usually less robust and reliable, particularly if uncertainties in problem parameters and design variables are expected.
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-optimal data analysis, isolated data points are special and important to DMs, as they signify unique and uncommon combinations of design variables that place them in an atypical part of the objective space compared to the bulk of the other Pareto-optimal data points.

3.3 Spatial Navigability

Formally, a visualization can be defined as a mapping of a set of data points $C \subset \mathbb{R}^m$ to a lower dimensional space $\mathcal{D} \subset \mathbb{R}^2$, such that spatial arrangements of points in $C$ are maintained in $\mathcal{D}$ as close as possible. In other way, we can say if a function $V(\cdot)$ can achieve this goal through a transformation $\mathcal{D} \subset \mathbb{R}^2 \leftarrow V(C \subset \mathbb{R}^m)$ is a good visualization process.

As human mind is trained to navigate in a spatial dimension involving two (or at most, three) dimensions, a good visualization technique should be able to provide the capability of spatial navigability after the transformation is completed. By being spatially navigable, formally, we mean the points in the lower-dimensional space are arranged in such a way that if a DM navigates through a specific topological path in $C$, the path should closely imitate a homotopic path in $\mathcal{D}$. Let us assume two end points $\{p, q\} \in P \subset C$ are connected by a topological path $P : [0, 1] \to P$. If there is a homotopically equivalent path $Q : [0, 1] \to Q \subset \mathcal{D}$ with two end points $\{r, s\} \in Q$, then we can say that the transformation $V(\cdot)$ produces a spatially navigable mapping.

4 THE VISUALIZATION METHOD

PaletteStarViz generates visualization by decomposing the data points in a layer wise manner using StarCoordinate plots, similar to a previous study with RadViz plots [12]. The idea is similar to the analysis performed in Figure 1 – i) Peeling: find each layer of data-set starting from the boundary and gradually proceed to the core. ii) Mapping/Flattening: represent each layer using a neighborhood preserving two-dimensional mapping. As a result, we get a representation of the high-dimensional data-set similar to that of Figure 1c. The goal of peeling step is to capture the center-outward ordering [13] of the point data-set. The goal of flattening (i.e. mapping) step is to represent each layer to make it suitable

---

Fig. 3. A three-dimensional Pareto-optimal front of DEB-3DK problem with a single knee at its center (1,005 data points).
for human perception and visual/spatial navigation. In terms of a formal definition of ‘layer’, we borrow the idea of point depth or data depth from the topological data analysis (TDA) domain [11].

Point depth or data depth provides center-outward orderings of points in the Euclidean space of any dimension and leads to a new non-parametric multivariate statistical analysis method where no assumption regarding the distribution of the data points is needed. A point depth measures how deep (or central) a given point $f \in C$ is relative to the entire data-set $C$. Enumerating the point depth is one way to infer the shape of the data-set in high-dimensional objective space, regardless of points being arranged on a regular grid in $\mathbb{R}^m$.

4.1 Shape of A Pareto-front

In our proposed visualization method, we use the concept of non-convex hull peeling depth, [14] [15] 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 the construction of the smallest non-convex hull which encloses all the 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 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 layer a point belongs to, the deeper the point is within the data set. Non-convex layer of a data set can be obtained through the construction of $\alpha$-shapes. [15]. 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 data points [16]. For example in Figure 1b, if we assume that the depth contours are nested as cubes, the boundary cube has a depth-contour of 1, the intermediate cube has a depth-contour of 2 and the smallest cube at the core has a depth-contour of 3.

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

4.2 Computing The Depth Contours

Since the $\alpha$-shape algorithm can be slow to run in a high-dimensional space, we have modified the algorithm to make it faster with a penalty for reduced accuracy. The approximate $\alpha$-shape algorithm is described in Algorithms 1 and 2. This is due to the fact that we do not need the most accurate enumeration of depth contours and some errors are tolerable, as we combine multiple boundary layers together to represent a few combined layers at the end.

Algorithm 1 starts with a set of points $C \subset \mathbb{R}^m$. At lines 1 to 3, we initialize the depth value of each point. Then we find the Delaunay triangulation using the Q-Hull algorithm [17]. We keep a flag for each edge of all the $m$-simplexes found after the triangulation. This flag is used for counting how many times an edge has been included or excluded during the filtration. The loop at line 13 finds the edges to be removed (or kept) to get the final $\alpha$-shape. Also During the filtration phase, we need to find the circum-radius of the hypersphere of each simplex, this can be achieved using the formula described in [18]. In the case of exact $\alpha$-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 computationally intensive step, instead, we assume that filtration by the removal of longest edge in the convex-hull should give an approximate
Algorithm 1 An approximate $\alpha$-shape algorithm.

Require: A set of $n$ data points $C \subset \mathbb{R}^m$
1: for each point $f_i \in C$ do
2: $D(f_i) \leftarrow 0$
3: end for
4: $d \leftarrow 1$
5: repeat
6: Delaunay triangulate $C$: $(T, H) \leftarrow C$. s.t. $T = \{\Delta_1, \Delta_2, \ldots, \Delta_k\}$ is a set of all $m$-simplexes and $H = \{e_1, e_2, \ldots, e_l\}$ is a set of convex hull edges
7: for each simplex $\Delta_j \in T$ do
8: for each edge $e \in \Delta_j$ do
9: $\text{flag}(e) \leftarrow 0$
10: end for
11: end for
12: $\alpha \leftarrow \min\{e_1, e_2, \ldots, e_l\}$ (Approximate $\alpha$)
13: for each simplex $\Delta_j \in T$ do
14: $r \leftarrow$ circum-radius of $\Delta_j$
15: if $r > \alpha$ then
16: for each edge $e \in \Delta_j$ do
17: if $\text{flag}(e) \geq 0$ then
18: $\text{flag}(e) \leftarrow \text{flag}(e) - 1$
19: end if
20: end for
21: else
22: for each edge $e \in \Delta_j$ do
23: $\text{flag}(e) \leftarrow \text{flag}(e) + 1$
24: end for
25: end if
26: end for
27: $R \leftarrow \{\emptyset\}$
28: for each simplex $\Delta_j \in T$ do
29: for each edge $e \in \Delta_j$ do
30: if $\text{flag}(e) \geq 0$ then
31: $R \leftarrow R \cup e$
32: end if
33: end for
34: end for
35: $B \leftarrow \text{find boundary points of } C$ using $R$ (Algorithm 2)
36: for each point $f_i \in B$ do
37: $D(f_i) \leftarrow d$
38: end for
39: $C \leftarrow C - B$
40: $d \leftarrow d + 1$
41: until $C = \{\emptyset\}$
42: return $\{D(f_1), D(f_2), \ldots, D(f_n)\}$

Algorithm 2 Find Boundary Points

Require: Set of all $m$-simplexes $T = \{\Delta_1, \Delta_2, \ldots, \Delta_k\}$ in $C$ and a subset of edges $R$ from $T$ after filtration.
1: for each edge $e \in R$ do
2: $\text{count}(e) \leftarrow [0, 0, \ldots, 0]$ s.t. $|\text{count}(e)| = k$
3: end for
4: for each edge $e \in R$ do
5: for each $m$-simplex $\Delta_j \in T$ do
6: if $e \in \Delta_j$ then
7: $\text{count}(e[j]) \leftarrow 1$
8: end if
9: end for
10: end for
11: $B \leftarrow \{\emptyset\}$
12: for each edge $e : \{p, q\} \in R$ do
13: if $\sum \text{count}(e[j]) \leq (m - 1)$ then
14: $B \leftarrow B \cup \{p, q\}$
15: end if
16: end for
17: return $B$
Fig. 4. Steps of a Palette visualization: (a) a simple three-dimensional spherical Pareto-optimal front with 376 data points. Each depth-contour is enumerated. Starting from the boundary, the outermost layer is colored dark blue and as we proceed to the central region of the surface, they turn green. (b) This figure shows how each layer is represented using a reversed SC-plot (see Section 4) and how they are stacked on top of each other. There are 17 such layers. (c) To reduce the clutter, we merge them to make the PaletteStarViz consisted of four layers.

The shape of corresponding non-convex hull. Therefore, we set $\alpha$ to the length of the shortest edge of convex-hull (i.e. in line 12 of Algorithm 1). 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 2.

The next step is to devise a way to flatten each layer. Since the example given in Figure 1 is organized in a cube, we can open up each nested cube in a regular way. In order to generalize, we resort to standard two-dimensional mapping methods like SC-plot. A resultant PaletteStarViz of the cube data-set is presented in Figure 1d. Here, we take reverse of each normalized objective function vector, i.e. replacing $f_i$ by $1 - f_i$. So that a point with best (i.e. minimum) value of $f_i$ stays very close to the anchor $f_i$. Now if we take a small neighborhood radius around the red point (i.e. "point of interest") on the intermediate layer, and take a normal projection on the top (i.e. boundary) layer, the corresponding points on the top layer are spatially close to the red point, although they have different point-depth rankings. This arrangement clearly conveys the relative placement of the data points in their original dimension.

The choice of number of layers in a PaletteStarViz plot along the z-axis is DM-dependent. However, in most of our illustrations 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 points. We attempt to distribute all points equally among the layers in this study. An overall construction process of PaletteStarViz is presented in Figure 4.

5 RESULTS AND COMPARATIVE ANALYSIS

To demonstrate the working of the proposed method, we consider a number of multi- and many-objective optimization problems. Readers should be aware that none of the existing EMO visualization methods [19] consider the topological properties of the Pareto-optimal front. Therefore, we have only compared with the most widely used methods such as PCP, Radviz, Heat Map, etc.

First, we visualize the relative position of a knee in a high-dimensional Pareto-optimal front, we use a benchmark test problem from [10] called DEBMDK. The Pareto-front presented in this paper has single knee at the central region. The results for three-, and eight-dimensional DEBMDK problems are presented in Figures 5a and 5b, respectively.

This might depend on the properties of the data set as well.
In the Figure 5a and 5b, the bottom layer corresponds to the deepest (near core) points in the original dimension. The knee region is marked with a few dark-red points in the bottom-most layer. The points in the bottom layer (i.e. light green) are located near the center of the data points, where the points close to the boundary (dark blue) are on the top most layer. 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 worst-objective points indicate that a large trade-off exists at these extreme points. For a comparison, we visualize the eight-dimensional problem using the original Radviz plot in Figure 5c. Here, we cannot see any boundary-core feature of the data set. As somewhat evident from the colors, the boundary points cover the core points hence we cannot visualize any of the light green (i.e. points in the core) points on the plot. Moreover, it is not also possible to understand where the exactly knee points are located in the high-dimensional space.

Figure 5d represents a PCP for DEB8DK problem. The knee points are colored in red and the three arbitrary boundary points are colored in blue. The rest of the points are colored in light gray. If there was no coloring applied, it is impossible to comprehend which points are on knee or on the boundary, or what makes them a knee point.

Next, we investigate if the proposed method can address the topological features of a high-dimensional Pareto-optimal data set. We modify DTLZ2 problem [20] 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 6a (a small cluster lying near the worst \( f_3 \) value.). When constraint functions are present, we follow a different color gradient scheme – the points that are close to the constraint boundary are painted as \textcolor{lightgray}{pink} and those that are far are painted with \textcolor{lightgray}{cyan}. One cluster has many more points than the other, thus the smaller cluster creates an \textit{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 isolated 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 values of \( f_1 \) and \( f_2 \). The corresponding PaletteStarViz for the three-objective problem is presented in Figure 6b. The isolated region opposite the \( f_3 \) anchor is clearly identifiable. An eight-objective case of the same problem is presented in Figure 6c. The isolated region near the \( f_4 \) anchor – opposite of \( f_8 \) – is clearly identifiable. There are 4,004 points in this data-set, but as the plot shows there exists a single knee point close to minimum \( f_3 \) point (biggest red circle in the middle layer). The isolated region (near minimum \( f_4 \) point) does not contain any high trade-off point.

Thus, the proposed visualization approach is capable of reducing a total number of 4,004 points to a few key points –
a high trade-off point (red circle) near the core of the trade-off frontiers and a representative isolated point but not having a high trade-off, for this problem. This will be extremely beneficial to the decision-makers (DMs).

In many real-world problems, a Pareto-front can be consisted of clusters of data points having different dimensions. For example, a Pareto-front may be consisted 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 PaletteStarViz method works on such a scenario, we consider the DTLZ8 problem [20], where the Pareto-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 7a. The PaletteStarViz of a four-objective version of the same problem is presented in Figure 7b. 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 case, the PaletteStarViz plot allows a DM to look at only about 20 critical points (marked in red color) from a set of 1,292 non-dominated points for further analysis. If the DM prefers boundary points (for which at least one of the objectives has an extreme value) a few red-circled 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 three red-circled points in Layers 3 and 4 (from top) can be further looked into.

6 AN EXAMPLE MCDA USING PARETO RACE

As a proof of concept of utilizing the proposed PaletteStarViz method in aiding Multi-criteria Decision-Making and Analysis (MCDA) by decision-makers (DMs), we demonstrate its use in the Pareto-race [21] method. Pareto-race is an interactive method, in which DMs provide an $m$-dimensional directed vector in the objective space (usually, starting from the nadir point to move towards the ideal/utopia point). Points are created on the directed vector to simulate the movement along the line. For every intermediate point, the best trade-off point from the obtained EMO set of points is chosen using the achievement scalarization function (ASF) [22] with a weight vector of ones. DMs are then expected to analyze the chosen trade-off points as they are identified for every reference point on the directed vector and increase...
Fig. 7. (a) A three-objective Pareto-optimal data set (1,044 points) for the DTLZ8 problem. (b) A PaletteStarViz plot of the four-objective DTLZ8 Pareto-optimal front (1,292 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.

Fig. 8. (a) A 10-objective Pareto-optimal data set (3,112 points) for the GAA problem presented in three-dimension. (b) A PaletteStarViz plot of the same Pareto-optimal front. (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.

or decrease their speed of movement or steer in a different direction. The process is expected to lead DMs to land up in a compromise solution which DMs like. Since PaletteStarViz can present a clear and easily interpretative spacial arrangement of the data points in the high-dimensional space, the Pareto-race method can guide a DM through the high-dimensional space in an intuitive manner and help in identifying the location of the chosen trade-off point: lying on boundary or core, lying close to a constraint boundary or not, lying close to a high trade-off point or not, lying on an isolated part or not, lying on a specific desired part or not, etc.

An example of this illustration is presented in Figure 7c for four-objective DTLZ8 problem. In this MCDA process, the DM starts from the nadir point and moves towards the ideal point in the four-dimensional objective space with uniform speed. Points a and b represent the first two points on the adjoining vector and their respective non-dominated points are shown on the PaletteStarViz 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. The DM now gets interested in point b, as the DM may have a preference for a point which lies at the core of the frontier. So, the DM
slows down at the directed vector and reduces speed to create points \( c \) and \( d \). The respective trade-off points stay at the core layer. To focus the investigation further, DM reduces the speed further and create points \( e \) and \( f \). However, the respective trade-off points seem to be left behind in the inner layers and move away from the core-region while moving towards the boundary layer. Any further movement along the directed vector seems to be associated with trade-off points which lie on the two outer-most layers, before converging to point \( n \) which is the respective ASF point for the ideal point. The whole journey from nadir to ideal point provide a lot of useful information to the DMs in choosing a preferred points. The trade-off points can be marked in constraint-based coloring approach (pink vs. cyan) so that every ASF point can be visualized for their closeness to constraint boundary, knee point, isolated region, etc. No existing visualization method has such a capability to provide so rich and useful information to the DM.

A natural application of this approach can demonstrated by illustrating promising solutions (e.g. points with higher trade-off) around each point that was found during the exploration phase. One such example is presented in Figure 8. Here we show obtained non-dominated points for a 10-objective, 18-constraint General Aviation Aircraft Design (GAA) problem [23]. The scatter plot of the Pareto-optimal front in the first three-objective space is presented in Figure 8a to show that there are two clear clusters marked based on their closeness to constraint boundaries – points that are close to one of the constraint surfaces are colored as pink and points away from the constraint are marked in cyan. The corresponding \( 2 \frac{1}{2} \)-dimensional PaletteStarViz plot is presented in Figure 8b. In this case, as every time the DM discovers a new solution in Figure 8c, we present all the points within its small neighborhood with the specified color coding; as if we light up each region where a solution is found during the explorations. Each exploration is performed according to a pre-specified direction vector from the Nadir to Utopia point (given in Figure 8d). This process allow the DM to navigate the surrounding regions for high trade-off points. For example, from 3,112 non-dominated points, DM can focus on the three high trade-off points (close to points, \( e \), \( f \) and \( d \)) for further analysis. It is clear from these illustrations the uniqueness and advantages of our proposed method for multi-objective decision making tasks.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated a new visualization approach – PaletteStarViz – to address the issue of high-dimensional Pareto-optimal data visualization to aid a multi-criteria decision making (MCDM) task. Non-dominated objective vectors in more than three-dimensional space have been mapped into three or four two-dimensional Star-Coordinate plots (SC-plots) and stacked according to their point-depth ranking, which preserves the center-outward placement [13] of data points in their original dimension. PaletteStarViz has also allowed us to apply different color and marker sizing schemes to filter and highlight vital multi-criterion functionalities by clearly marking a few critical points which can be helpful to reduce the cognitive burden of decision makers (DMs). The use of PaletteStarViz for an MCDM task of choosing a single preferred solution has been another hallmark aspect of this paper.

REFERENCES


