A Topologically Consistent Visualization of High Dimensional Pareto-front for Multi-Criteria Decision Making

AKM Khaled Ahsan Talukder
Department of Computer Science and Engineering
Michigan State University
East Lansing, USA
talukde1@msu.edu

Kalyanmoy Deb
Department of Electrical and Computer Engineering
Michigan State University
East Lansing, USA
kdeb@egr.msu.edu

COIN Report Number 2018006
www.coin-laboratory.com

Abstract—There are a good number of different algorithms to solve multi- and many-objective optimization problems and the final outcome of these algorithms is a set of trade-off solutions that are expected to span the entire Pareto-front. Visualization of a Pareto-front is vital for an initial decision-making task, as it provides a number of useful information, such as closeness of one solution to another, trade-off among conflicting objectives, localized shape of the Pareto-front vis-a-vis the entire front, and others. Two and three-dimensional Pareto-fronts are trivial to visualize and allow all the above analysis to be done comprehensively. However, for four or more objectives, visualization for extracting above decision-making information gets challenging and new and innovative methods are long overdue. Not only does a trivial visualization becomes difficult, the number of points needed to represent a higher-dimensional front increase exponentially. The existing high-dimensional visualization techniques, such as parallel coordinate plots, scatter plots, RadVis, etc., do not offer a clear and natural view of the Pareto-front in terms of trade-off and other vital localized information needed for a convenient decision-making task. In this paper, we propose a novel way to map a high-dimensional Pareto-front in two and three dimensions. The proposed method tries to capture some of the basic topological properties of the Pareto points and retain them in the mapped lower dimensional space. Therefore, the proposed method can produce faithful representation of the topological primitives of the high-dimensional data points in terms of the basic shape (and structure) of the Pareto-Front, its boundary, and visual classification of the relative trade-offs of the solutions. As a proof-of-principle demonstration, we apply our proposed palette visualization method to a few problems.

Index Terms—Many-objective Optimization, High-dimensional Pareto-Front, Visualization, RadVis, Parallel coordinate plot, Decision Making.

I. INTRODUCTION

With the recent advancement in many-objective optimization (i.e. problems with four or more conflicting objective functions), there have been a number of post-optimization issues that need to be addressed on an immediate basis, if these methods have to be routinely used in practice. One of the most challenging issues is to comprehend the obtained trade-off solutions produced by a many-objective optimization (MOOP) solver, so that one or more preferred solutions can be chosen in a systematic manner by the decision makers. In an MOOP problem, if the target problem is consisted of two or three objectives, a 2D or 3D scatter plot is the most intuitive way to visualize the results. Decision-makers (DM) 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 migrating to a neighboring solution. DMs can also consider other criteria, such as, robustness or reliability of a solution (i.e. risk assessment) or some other utility functions representing the practical usefulness of a solution.

Naturally, it is not possible for DMs to comprehend four or more spatial dimensions visually in a straightforward manner, but a number of different visualization methods exists and are popularly used: (i) parallel coordinate plot (PCP) [1], RadVis [2], multiple scatter plots [3], and others. While most of these methods allow high-dimensional data to be plotted in a two-dimensional plot, they often do not implicate the inherent trade-off information against their neighbors. In fact, an idea of who is one’s neighbor is also not clearly demonstrated by these plots. While such visualization methods may be useful in other contexts, for multi-criterion decision making purposes, they certainly do not provide a clear picture. The goal of this paper is to highlight the need for specialized visualization methods for trade-off data solely for the purpose of a convenient and easier decision making task. We propose one such technique and provide our initial results to demonstrate its working principle. This is an important topic to the field of evolutionary many-objective optimization (EMO), which is currently handicapped without such tools in making the otherwise efficient optimization algorithms [4], [5], [6] less motivating to be applied in practice.

The paper is organized as follows: in Section II, we briefly describe a few popularly used existing visualization techniques and discuss their limitations thereafter. In the section III, we discuss a number of key decision-making considerations which DMs may be interested in applying to a set of trade-off solutions. Then, in Section IV, we describe our proposed technique in details. In Section V, we present some results and analysis of the proposed method and conclude the paper.
II. EXISTING VISUALIZATION METHODS

In terms of high-dimensional and multivariate data analysis, there have been a number of different visualization methods suggested. Some of which have been applied to the visualization of MOOP solution sets. As we have already mentioned, one of the most widely used methods is the PCP [1], in which every solution is represented with a line marking the individual objective values on multiple parallel vertical axes. Thus, if the lines for two solutions cross each other between two objective axes, these two solutions constitute a trade-off between these two specific objectives. However a PCP plot does not provide a clear indication of the trade-off between one solution with its neighbor in a comprehensive manner. Another challenging task is to arrange the parallel vertical axes optimally. Without the optimal arrangement of objective axes, a meaningful comprehension of the trade-off points becomes difficult.

A ‘Heatmap’ is another approach [7], in which all objective values (and variables and constraint values) of every trade-off solution are stacked in a row but colored based on their relative values. If an objective value is closer to its individual minimal value (assuming minimization of that objective), a darker red marker is placed. Thus, two rows (representing two trade-off solutions) having dissimilar colors at its objective places will indicate widely different solutions, but a keen eye enabling detection of changes of colors at two objective locations between two solutions can only understand the trade-off that the two solutions possess.

Despite the loss of important information relevant to decision making, dimension-reduction methods often offer a useful visualization tool and it is especially appealing since they are able to represent solutions as a point-clouds in the two-dimensional plane, since humans are more adept at interpreting planar diagrams. Along with PCA, there are sophisticated dimension-reduction methods such as self-organizing maps (SOM) [8], decision tree and Sammon Mapping [9], which can map high-dimensional data points onto a two-dimensional plane. More recently proposed Stochastic Neighborhood Embedding (SNE) [10] techniques are gaining popularity in the high-dimensional data visualization community. The interesting aspect about SNE based techniques that given a similarity metric, they can infer clusters in the data points. Hence such technique is capable of showing similar data points in a separate point cloud, which is not easily identifiable just looking at the data values.

A. Limitations of Existing MOOP Visualization Techniques

Despite the fact that there are a number of multivariate data visualization techniques, we do not foresee their direct application in the evolutionary multi-objective optimization (EMO) and multi-criterion decision-making (MCDM) context. However, we discuss a few recent methods which are promising.

In RadVis [2], each dimension is represented by a dimensional anchor, and each dimensional anchor is distributed evenly on a unit circle, as shown in Figure 1. Each line in the data set corresponds to a point in the projection, that is linked to every dimensional anchor by an imaginary spring. Each spring’s stiffness corresponds to the value for that particular data point in that particular dimension. The position of the point is defined as the point in the 2D space where the all springs are in equilibrium. Although the procedure allows a higher-dimensional data to be mapped in a two dimensional space, RadVis fails to represent even the most basic underlying structure of the data points. For example, if the data points are laid out in an n-dimensional hypersphere, and if we want to see which points lie on the surface and which lie inside the hypersphere, RadVis can not present them is a distinguishable way. This situation is presented in Figure 1. We find another visualization method called 3D-RadVis [11] which maps n-dimensional data points onto a three-dimensional space, for which the third axis represents the extent of non-linearity of the Pareto front. However, the proposed 3D-RadVis has a similar problem to that of the original 2D-RadVis in that it is not able to maintain the geometric relationships among solutions from higher dimension to 3D.

Nevertheless, there have been some recent research reports that address this issue from EMO perspective. For example, in [12], the authors present an idea called “Prosection Method”. The technique is to project the whole set of solutions to the orthogonal plane where only the solutions from the chosen section are projected. Because multiple planes can be selected for the projection (as in the scatter plot matrix), a prosection matrix is used to visualize all the orthogonal projections simultaneously. In addition, color coding can be used for distinguishing between feasible and infeasible solutions. However, as this approach depends on the construction of orthogonal
plane from multiple objectives to be used as an axis in the final plot, it still suffers from similar limitations as other methods, specially when the dimension is more than four.

We can find other examples in [13] where the authors present a multiple ways to represent a high-dimensional PF. The paper addresses a common problem with the well-known Heatmap visualization. Since an arbitrary ordering of rows and columns renders the Heatmap unclear, the method uses spectral seriation to rearrange the solutions (along with the objective values) and thus enhance the clarity of the Heatmap. A multi-objective evolutionary optimizer is used to further enhance the simultaneous visualization of solutions in objective and parameter space.

In another study [14], the authors proposed a variant of RadVis method that maps data points from a high-dimensional objective space into a 2D polar coordinate plot while preserving Pareto dominance relationship, retaining shape and location of the PF, and maintaining distribution of individuals. The convergence of the approximate front is measured by radial values of all population members on that front. Meanwhile, the diversity performance is mainly determined by niche count of each subregion in a high-dimensional objective space.

In terms of lower dimensional embedding techniques, Stochastic Neighborhood Embedding (SNE) [15] [10] has gaining popularity in the recent years. SNE is a probabilistic approach that can place data points, described by high-dimensional vectors or by pairwise dissimilarities, in a low-dimensional space in a way that preserves neighborhood relations. A Gaussian is centered on each solution in the high-dimensional space and the densities under this Gaussian (or the given dissimilarities) are used to define a probability distribution over all the potential neighbors of the solution. The aim of the embedding is to approximate this distribution as well as possible when the same operation is performed on the low-dimensional “images” of the data points. A natural cost function is a sum of Kullback-Leibler divergences, one per solution, which leads to a simple gradient for adjusting the positions of the low-dimensional images. Unlike other dimensionality reduction methods, this probabilistic framework makes it easy to represent each point by a mixture of widely separated low-dimensional images. Although SNE can construct reasonably good visualizations, it is limited by a cost function that is difficult to optimize and also it suffers from the so called crowding problem. Moreover, TSNE/SNE like method is completely oblivious to the topological primitives of the high-dimensional data points. This limitation is presented in Figure 2.

All the existing methods for MOOP solution visualization mainly focus on the visualization of convergence of the PF. Although visualization of the movement of solutions in the search space might be useful for an algorithm designer, DMs are generally not interested in it. Reducing the number of solution choices and the position of infeasible/robust solutions (with respect to others) are more important to a DM. Also human eye is more used to spatial representation of data or function in two and three dimensions. Spatial representation enables us to navigate the space and make decisions accordingly. Therefore, reduction of the cardinality of Pareto solution choices and their simplest comprehension through a smart lower-dimensional visualization method is the main goal of this study.

III. VISUALIZATION FROM A DM’S PERSPECTIVE: THE PALETTE VISUALIZATION

Our goal of the visualization method differs from the existing techniques in a number of interesting ways:

- **Topological Consistency:** We are specifically interested in how close our visualization preserves the topology of the solutions from the original objective space to 2D or 3D.

- **Spatial Navigability:** Human mind is trained to navigate in a spatial dimension. Hence, two and three dimensional scatter plots are the most intuitive and intelligible representation of data. We want to keep this property of ‘spatial navigability’ in the visualization.

- **Solution Properties:** We are also not interested in how the solution converges to the true PF in this study, or visualization of the search trajectory. We are interested in a representative scheme of trade-off Pareto data so that a multi/many-criteria based decision making can be performed on them either quantitatively or qualitatively.

From our previous experiences with industry collaborations, we have seen that a decision making procedure requires a completely different set of considerations than an EMO developer considers within an optimization algorithm. The visualization procedure which we propose here is mainly followed from the RadVis method [2], but extended in a three-dimensional format, which we call here as the Palette Visualization. The entire plot is consisted of multiple layers two dimensional RadVis plots where each layer comes from the different layer-wise division of higher-dimensional Pareto solutions from boundary to the core. In the following subsections, we describe the procedure stepwise.

![Figure 2](image.png)

The left figure is the actual input data point, where the center blue points are surrounded by yellow points. The right figure is the TSNE mapping of the input point into another space after 5,000 iterations with perplexity value [15] of 2. The topological relation of the data points are lost.
A. Topological Consistency

As we have seen in Section II-A, in a high-dimensional space, it’s difficult to preserve the very basic topological primitives when we map the points onto lower dimensional space. Even the inside-outside relations of the point cloud do not conform to their original relation after the mapping. We solve this problem by decomposing the point cloud into layers of convex hulls and each convex hull layer is plotted on a three-dimensional RadVis plot where the position of each layer is defined on the z-axis. Generally outer layer of the point cloud will have the highest z-axis value and inner layers have smaller z-axis value as inner the layers come to the vicinity of the centroid of the point cloud. In this way, we can ensure that outer layer of the original point cloud stays outward on the RadVis plot so that the data points do not fall over each other. Moreover, all the points on the layers maintain a similar distance from the centroid as in the original data points in the high-dimensional space. This arrangement can at least maintain two basic topological properties of the high-dimensional data points: a) distances from the centroid of the data points and b)inner-outer relations of the original data points. An example of such plot is presented in Figure 3.

B. Spatial Navigability

Basically, the three-dimensional layer-wise RadVis plot gives the spatial navigability on the high-dimensional space. Since Pareto optimal objective function values only stay in one of the high-dimensional quadrant (i.e. empty 3D-sphere in the first quadrant if the Pareto front is minimizing all non-negative objectives.), the shape of the point cloud is not complete. As a result, the top portion of the shape of the point cloud is not going to be superimposed on the points on the bottom portion of the point cloud (in the case of three-dimension). Therefore, each layer of the three-dimensional RadVis comes from unique boundary solutions of the original data points. This actually provides the spatial navigability property to the visualization. A DM can now traverse the high-dimensional data points without confusing the relative neighborhood relationships of the original data points. A DM can also understand which points lie on the boundary of the original high-dimensional space and navigate and visualize the entire Pareto-front as if he/she is looking at a two/three dimensional scatter plot.

C. Pareto Solution Properties

Once we have the skeleton found from the three-dimensional RadVis plot, we can ask that for a given set of Pareto-optimal solutions (over 4 dimensions), what aspects of the solution set is more relevant to look at? Perhaps, the most desirable aspect of the Pareto-front solutions are those that offer most trade-off among all the points in a solution set. In literature they are known as Knee points [16]. If we assume that we do not have any knowledge about the user’s preferences, it can be justified that the region around that knee is most likely to be interesting for the DM. Since the knee solutions are characterized by the fact that a small improvement in either objective will cause a large loss in the other objective, which makes such movement in either direction not very useful.

Formally, the knee solutions are those, from which, moving towards any of the objective axes will cause comparatively higher amount of deterioration in one or more other objectives. In this paper, we will follow the definition of knee points discussed in [17]. Let us consider the simple Pareto-optimal front depicted in Figure 4, with two objectives to be minimized. This front has a clearly visible bulge in the middle. If we assume
linear preference functions, and (due to the lack of any other information) furthermore assume that each preference function is equally likely, the solutions at the knee are most likely to be the most desirable choice of the DM. Note that in Figure 4, due to the concavity at the edges, similar reasoning holds for the extreme solutions (edges), which is why these should be considered knees as well. The goal of this paper is to how for the extreme solutions (edges), which is why these should be the most desirable choice of the DM. Note that in Figure 4, due to the concavity at the edges, similar reasoning holds for the extreme solutions (edges), which is why these should be considered knees as well. The goal of this paper is to how for the extreme solutions (edges), which is why these should be the most desirable choice of the DM.

Fig. 4. A simple 3-D Pareto-optimal front of DEBMDK problem with a single knee [18]. The axes are \(f_1, f_2\) and \(f_3\). The knee solutions (i.e. solutions with better trade-off) are presented by circles with increasing radius.

IV. PROPOSED PALETTE VISUALIZATION PROCEDURE

The Palette visualization is consisted of mainly of three chronological steps – a) Finding multiple clusters of points in the original high-dimensional objective space, b) Layer-wise decomposition (boundary to core) of points of each cluster, and c) Graphical representation of different relevant decision making information for trade-off points in each cluster. Each of these steps is described in the following subsections.

A. Clustering

The complete Pareto-optimal front may be constituted with a number of disconnected regions. In order to not confuse properties of each disconnected region independently, our proposed Palette visualization works for a single cluster at a time. Therefore, we first apply a suitable clustering approach in the original high-dimensional objective space to separate trade-off points into separate clusters. In this study, we use an unsupervised algorithm, such as DBSCAN [19], for this purpose, but any other clustering method can also be used.

B. Layer-wise Decomposition

This is the main crux of the proposed palette approach. For each cluster of data point, we further sub-divide points into multiple layers, starting from boundary layer which is composed of all boundary points in the high-dimension. Thereafter, intermediate layers are extracted one by one until all points are classified into different layers. This procedure requires a boundary identification method which can be applied repeatedly starting with the complete set of points and then deleting points which have already been identified with a layer. The number of total layers will depend on the placement of the points with respect to each other and also on the definition of boundary identification algorithm. In this study, we implement a simple Jarvis-March [20] based algorithm which is described in Algorithm 1.

The algorithm first finds the centroid \(\mu\) of the clustered data points. All points are then sorted according largest distance from the centroid to shortest. The point \(q\) having the largest distance is declared as one of the boundary points in the boundary set \(Q\). Then, for every point \(p\) from the sorted order, the procedure finds the angle \(\theta\) between two vectors: \((p - \mu)\) and \((p - q)\), for every member \(q \in Q\). If \(0 < \theta \leq \theta_t\) for any of the already identified boundary point \(q\), then \(p\) is not a boundary point and the next point in the sorted list is checked. In our study, we set the angle threshold \(\theta_t\) to \(\pi/60\). In case a boundary point is identified, it is included in the boundary set \(Q\). Since in higher dimension, wrapping around is not possible, our algorithms runs in \(O(n^2)\) time.

Although the above method is not the most-efficient procedure, but it is one of the ways to identify boundary points. Other methods, like Quick-hull [21], can also be used and we are investigating other more efficient methods.

The above layer-wise division of points may end up producing a large number of layers for our next analysis. To make the procedure amenable for the final decision-making purpose, multiple layers can be combined together and only a handful (say, 3-5) of layers can be formed. For this purpose, we combine layers so that almost an equal number of points are included in each finally decomposed layer.

C. Graphical Representation of Layers

Once the layer-wise decomposition is performed, points from each layer are then represented in a 2-D RadVis plot. Although we have used RadVis plot here, other many-dimension to 2-D mapping can also be used. *Stochastic Neighborhood Embedding (SNE)* [10] is one other representation scheme which can be used. It is important to note that the outer-most (boundary) layer in higher-dimension need not map as boundary points on a 2-D plot (such as RadVis or SNE); the points can cover the entire 2-D mapping region inside the RadVis polygon. For this purpose, we plot the subsequent layer of points staggered in a third dimension (vertical), in which the top-most 2-D RadVis plot corresponds to the boundary layer, the next one below corresponds to the next to boundary layer and so on. The bottom-most 2-D RadVis plot correspond to the inner-most layer in the original high-dimensional space.

Once the layer-wise points are mapped in 2-D RadVis plot, for decision-making purposes, we can represent the points according to various preferential features. Here, we compute...
Algorithm 1 Decompose Pareto-front into boundary layers.

Require: a cluster of data points $C$ found from applying DBSCAN on the Pareto-front in original dimension
1: $B \leftarrow \emptyset$
2: $\mu \leftarrow$ centroid of $C$
3: sort all points $p \in C$ in descending order of $\|p - \mu\|$ 
4: repeat
5: $i \leftarrow 0$
6: $L_i \leftarrow \emptyset$
7: for each point $p \in C$ do
8: if $\text{flag}(p)$ is not set then
9: $\hat{u} \leftarrow (p - \mu)/\|p - \mu\|$ 
10: for each $q \in C$ do
11: $\hat{v} \leftarrow (p - q)/\|p - q\|$
12: $\theta \leftarrow \cos^{-1}((\hat{u} \cdot \hat{v})/(\|\hat{u}\|\|\hat{v}\|))$
13: if $0 < \theta \leq \theta_i$ then
14: $\text{flag}(p) \leftarrow \text{non-boundary}$
15: end if
16: end for
17: if $\text{flag}(p) \neq \text{non-boundary}$ then
18: $\text{flag}(p) \leftarrow \text{boundary}$
19: $L_i \leftarrow L_i \cup p$
20: end if
21: end if
22: $B \leftarrow B \cup L_i$
23: $C \leftarrow C - L_i$
24: $i \leftarrow i + 1$
25: until $C \neq \emptyset$
26: return $B$

The trade-off information of each point in the original high-dimensional space and mark a point with a larger circle, if the point corresponds to a higher trade-off.

1) Computation of Trade-off Value: In our case we are interested in the relative trade-off of solutions known as knees. For the identification of the knees solutions, we will follow the method presented in [17]. The definition of Pareto-optimality is a non-reflexive, transitive, and antisymmetric binary relation, i.e., Pareto dominance. Pareto dominance is defined in the following manner:

Given a set of objective functions $\mathbf{F} = [f_1, f_2, \ldots, f_M]$ to be minimized, the vector $\mathbf{F}(x_i)$ is said to dominate another vector $\mathbf{F}(x_j)$, denoted $\mathbf{F}(x_i) \prec \mathbf{F}(x_j)$, if and only if $f_k(x_i) \leq f_k(x_j)$ for all $k \in \{1, 2, \ldots, M\}$ and $f_m(x_i) < f_m(x_j)$ for some $m \in \{1, 2, \ldots, M\}$. A point $x^* \in S$ is said to be globally Pareto optimal or a globally efficient for a multi-objective optimization problem (MOP) if and only if there does not exist $x \in S$ satisfying $\mathbf{F}(x) \prec \mathbf{F}(x^*)$. $\mathbf{F}(x^*)$ is then called globally non-dominated solution.

Trade-off value can be computed over a pair of non-dominated objective vectors and may be defined as the net gain of improvement in some objectives offset by the accompanying deterioration in other objectives as a result of substituting one objective vector with another non-dominated objective vector.

Algorithm 2 Palette Visualization
1: compute trade-off value of each Pareto point
2: take all the Pareto-points, use DBSCAN algorithm and find $n$ clusters $\{C_1, C_2, \ldots, C_n\}$
3: take a cluster $C_i$
4: invoke Algorithm 1 to find the layer decomposition $B = \{L_1, L_2, \ldots, L_k\}$ of $C_i$
5: use RadVis to display each layer $L_i \in B$ on a two-and-half dimensional space in which the sequence of layers is staggered along the z-axis.
6: Pareto points with higher trade-off value are highlighted.

Mathematical definition of trade-off is commonly given for every pair of objective functions. The equation to compute this quantity has been borrowed from [17]:

$$T(x_i, x_j) = \sum_{1 \leq m \leq M} \max[0, f_m(x_i) - f_m(x_j)] / \sum_{1 \leq m \leq M} \max[0, f_m(x_i) - f_m(x_j)]$$

In the definition of $T(x_i, x_j)$, the numerator evaluates the total improvement gained by exchanging $x_j$ with $x_i$ while the denominator evaluates the deterioration caused by the exchange. The actual metric to evaluate the worth of a solution $x_i \in R \subset S$ (where, $R$ is an $\epsilon$ neighborhood around $x_j$) in terms of performance trade-off, is given in the following equation:

$$\mu(x_i, R) = \min_{\forall j: x_j \in R, x_i \neq x_j, x_i \neq x_j} T(x_i, x_j)$$

A solution with a larger value of the quantity $\mu(x_i, R)$, within a neighborhood of $R$, signifies if the solution belong to the knee region. If solution far from the knee region the $\mu$ value will be smaller.

An overall algorithm for the entire procedure is summarized in Algorithm 2. For multiple clusters, we can generate multiple palette for an easy visualization of different clusters. If the total number of layers are too many, we can also merge multiple layers into one layer to reduce the total number of layers which decision makers need to analyze. An example plot is presented in Figure 3 for a 3-D Pareto-front, where each of five layers on the palette is actually composed of multiple individual layers.

The above procedure makes a systematic division of the original high-dimensional data into separate clusters based on their relative location in the entire Pareto-front and then into different layers based on the centrality of points in their own cluster. This $2^{1/2}$-D representation of the points allows a decision-maker 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. Then, the relative trade-off value information of each point in each layer and in each cluster will allow DMs to analyze the Pareto-front further by only looking at a few critical points to finally select a single preferred solution.
V. VISUALIZATION RESULTS

To demonstrate the working of the proposed palette visualization concept, first, we use a benchmark test problem from [18], called DEBMDK. The problem has a parameter that can be changed to introduce multiple knees, however we choose the problem with a single knee for an easier demonstration of our palette visualization approach. In all cases, we have marked the point size according to the trade-off values so that a bigger marker means a solution with higher trade-off value. Also, the solutions that are farthest from the centroid is colored in red and closer ones are colored in blue color. The results for the five-dimensional DEBMDK problem is presented in Figure 5. It is interesting to note that in addition to the actual knee point, the trade-off metric value given by equation 2 is relatively large for the boundary points, we can see some points on the top-most layer with bigger radius.

Next, for the spherical Pareto-front problem shown in Figure 3, all Pareto points are drawn with a similar marker size, indicating that in this problem, there is no knee-like point. In such cases, DM may be tempted to select a boundary point – from the top-most layer in our palette visualization plot, unless there are other more compelling practicalities to consider.

Finally, we apply our method to the vehicle crash-worthiness (CARCRASH) problem from [22]. This is a three-objective optimization problem, however the Pareto-front in this problem is composed of three separate clusters (marked in the figure as clusters A, B and C). Our proposed palette visualization can show three clusters independently. The entire Pareto-front of the CARCRASH problem is presented in Figure 6.

The top cluster (A) and its palette visualization are presented in Figure 7. Since all points in this cluster are drawn with a similar marker size, there is a smooth variation of points in this cluster. There is one knee-like point (E). Either point E or a boundary point may be of interest to a DM and our top-most RadVis plot, representing boundary points, allows a DM to easily focus on the filtered boundary points for a preferred solution.

The palette visualization for the second and third clusters together are presented in Figure 8. A point with a relatively large trade-off is observed in this cluster. This point corresponds to point K marked in Figure 6, which staying in the intersection of two parts of the Pareto-front makes an interesting choice for the DM.

The most interesting part of this visualization is that irrespective of the dimension of the Pareto-front, a DM can easily navigate through the critical points in the objective space and make a decision about which solutions to choose. This visualization method thus helps to reduce the overhead associated with solution assessment and provides a great aid to the human-computer interaction during the decision making process.
VI. CONCLUSIONS AND FUTURE WORK

In this paper, we have demonstrated an alternative visualization approach to address the issue of high dimensional Pareto-front visualization. Our approach filters out a few critical points from the entire non-dominated set of solutions discovered by a multi-objective optimization algorithm. The identification of critical points is limited in this study only to knee points having a large local trade-off and boundary points which are important in their own rights. 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. Another important contribution is that the proposed palette visualization can be generically applied to any large dimension of trade-off data.

Our approach has mainly based on the RadVis mapping, but any other approach that tries to retain the global topological relations among the data points from high to low dimension can also be used. In the future we would like to improve the current boundary identification procedure by adopting an ‘alpha-shape’ [23] based non-convex hull finding algorithm. Along with the trade-off values of the solutions, we can also color the points according to the constraint violation values for constrained MOOPs. This will help a DM to have a clear idea of the extent of feasibility of their selected points. Our approach can also be helpful for visualization in other scientific domains, specially our method might be useful to the data-mining community.

REFERENCES


