A Taxonomy of Methods for Visualizing Pareto Front Approximations

Bogdan Filipič
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
bogdan.filipic@ijs.si

Tea Tušar
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
tea.tusar@ijs.si

ABSTRACT

In multiobjective optimization, many techniques are used to visualize the results, ranging from traditional general-purpose data visualization techniques to approaches tailored to the specificities of multiobjective optimization. The number of specialized approaches rapidly grows in the recent years. To assist both the users and developers in this field, we propose a taxonomy of methods for visualizing Pareto front approximations. It builds on the nature of the visualized data and the properties of visualization methods rather than on the employed visual representations. It covers the methods for visualizing individual approximation sets resulting from a single algorithm run as well as multiple approximation sets produced in repeated runs. The proposed taxonomy categories are characterized and illustrated with selected examples of visualization methods. We expect that proposed taxonomy will be insightful to the multiobjective optimization community, make the communication among the participants easier and help focus further development of visualization methods.

CCS CONCEPTS

• **Human-centered computing** → **Visualization techniques**; **Scientific visualization**; **Visualization theory, concepts and paradigms**; • **Theory of computation** → *Stochastic control and optimization*;

KEYWORDS

multiobjective optimization, visualization methods, taxonomy

ACM Reference Format:
Bogdan Filipič and Tea Tušar. 2018. A Taxonomy of Methods for Visualizing Pareto Front Approximations. In *GECCO '18: Genetic and Evolutionary Computation Conference, July 15–19, 2018, Kyoto, Japan.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3205455.3205607

1 INTRODUCTION

Multiobjective optimization deals with finding solutions to optimization problems with multiple conflicting objectives. An ideal multiobjective optimization algorithm returns a set of nondominated solutions approximating the Pareto front, referred to as an approximation set. While assessing the quality and comparing approximation sets is also a multiobjective problem, visualization can be utilized as a complementary technique to scrutinize the results. Visualization may be helpful in a number of tasks related to multiobjective optimization, such as estimating the properties of the Pareto front, investigating trade-offs between the objectives, identifying preferred solutions, tracking the convergence of algorithm runs, comparing the performance of algorithms, and supporting interactive optimization.

In the broad field of data visualization, the methods have traditionally been classified according to the characteristics of data being visualized, and the distinction between scientific visualization [22] and information visualization [37] has been adopted. This separation is practical, but may discourage research bridging the two subfields. To better understand how they relate and overlap, [39] introduces an alternative taxonomy of visualization methods considering the properties of the methods themselves and involving the users and their conceptual models.

On the other hand, not many attempts of classifying the visualization methods in view of multiobjective optimization have been reported. In a review of methods to visualize Pareto front approximations [42], we distinguish between the general multidimensional data visualization methods designed outside the field of evolutionary multiobjective optimization and the methods specifically designed for visualization in multiobjective optimization. While the former make no effort to preserve the Pareto dominance relation between solutions or any other feature specific to multiobjective optimization, the latter were designed as attempts to overcome these limitations.

In [4], visualization techniques are surveyed in the context of knowledge discovery in multiobjective optimization. The authors list numerous data mining methods that can be used to extract knowledge about the problems from the solutions generated in the optimization process and classify them by methodology and type of the extracted knowledge. While their top-level classification consists of descriptive statistics, visual data mining, and machine learning methods, the visual data mining methods are further divided into graphical visualization, clustering-based visualization and manifold learning methods. To our best knowledge, this is the most detailed classification of visualization methods for multiobjective optimization by now.

With the progress of multiobjective optimization, the number of visualization approaches deployed in the field rapidly grows too. This especially holds for newly designed methods tailored to

GECCO '18, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5618-3/18/07…$15.00
https://doi.org/10.1145/3205455.3205607

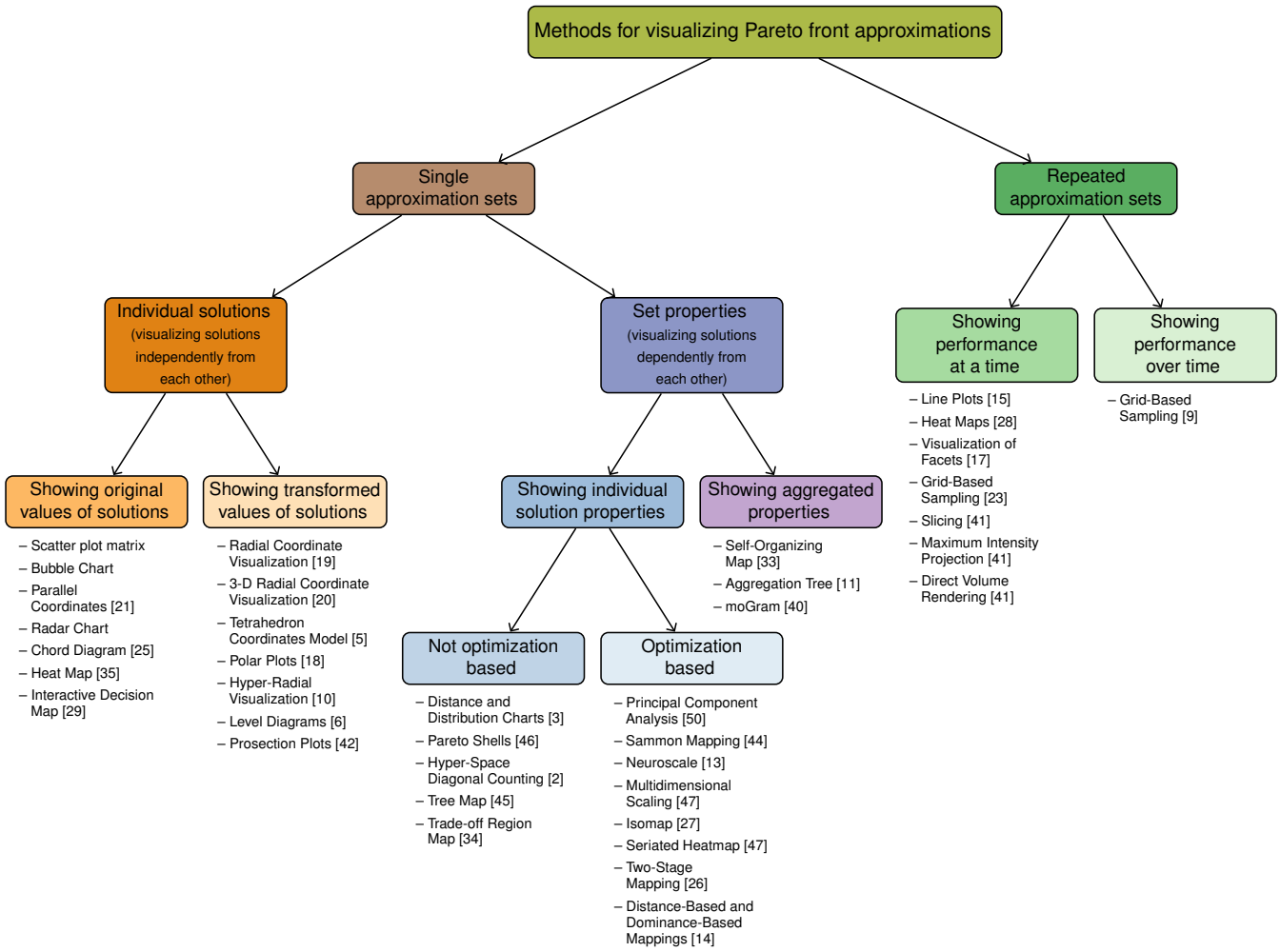


Figure 1: The proposed taxonomy.

the specificities of multiobjective optimization. To assist both the users and developers with a better insight into the field, we propose a detailed taxonomy of methods for visualizing Pareto front approximations. It covers the methods for visualizing individual approximation sets resulting from a single algorithm run as well as multiple approximation sets produced in repeated runs. Note, however, that besides approximation sets, visualization in multiobjective optimization covers additional aspects, such as visualizing the problem landscape in order to better understand the problem properties, or visualizing a (small) solution set to support decision making. While these are also important, they are outside the scope of this work and the related methods are not part of the proposed taxonomy.

The paper is further organized as follows. Section 2 presents the proposed taxonomy. Section 3 focuses on the methods visualizing single approximation sets. It introduces two approximation sets with particular characteristics, created to demonstrate the performance of the visualization methods. It then reviews the related categories of methods by presenting a selected method for each category, showing the resulting visualization of the test sets, and listing other methods fitting in the category. Section 4 deals with the methods for visualizing repeated approximation sets. It presents four groups of approximation sets mimicking results of repeated algorithm runs and discusses the available methods for visualizing algorithm performance at a time and over time. Section 5 concludes the paper with a summary of our work and some observations following the classification of methods for visualizing Pareto front approximations according to the proposed taxonomy.

2 THE PROPOSED TAXONOMY

The idea for the taxonomy is to build on the nature of the visualized data and the properties of visualization methods rather than on the employed visual representations, such as points, lines, graphs and maps. Shown in Figure 1, the taxonomy at the highest level distinguishes between the two fundamentally different categories of methods: those that visualize single approximation sets and those visualizing repeated approximation sets. The former category stands for methods that primarily represent one or few approximation sets

at a time. Even if multiple sets are shown, they are handled individually regardless of whether they result from one or more algorithms. These methods can also be used to monitor the algorithm progress by generating a sequence of such individual plots (animation). On the other hand, the latter category is aimed at representing multiple algorithm runs or the entire evolution of the optimization process in a single plot. This is achieved by introducing a function that collects the information on the optimizer performance in each point of the objective space and then visualizing this function.

Visualizing single approximation sets is further divided into visualizing individual solutions from an approximation set and visualizing the properties of an approximation set. Visualizing individual solutions means the solutions are visualized independently from each other, i.e., adding or removing a point to/from the original approximation set implies adding or removing the corresponding point in the visualized set without affecting the other points. This can be done in two ways, either by showing the original values of the solutions or first performing some transformation and then showing the transformed values of solutions.

Visualizing the properties of an approximation set means the solutions are visualized dependently of each other, i.e., any change of a point in the original approximation set reflects in a change in the considered set property and, consequently, in the visualization. This large category consists of the methods showing individual solution properties and the methods showing aggregated properties. Examples of individual solution properties are the relations between the points and the distances among them, while the aggregated properties are, for example, the clustering of solutions and the relations between objectives. The methods showing individual solution properties are further classified into those that involve no optimization procedure in extracting the set property to be visualized and those that make use of optimization for this purpose.

The second top-level category, i.e., the methods visualizing repeated approximation sets, comprises two lower-level categories, the methods showing the optimizer performance at a time and the methods showing the optimizer performance over time. They differ in the function used to represent the optimizer performance in each point of the objective space. In the former case it is the Empirical Attainment Function (EAF) [16] that collects the information on the proportion of runs in which the points were attained. In the latter case a generalization of the EAF is used, called the Average Runtime Attainment (aRTA) function [9] which additionally contains the information on when a point was first attained. In both cases the visualized function can be represented the aggregated information in a single plot.

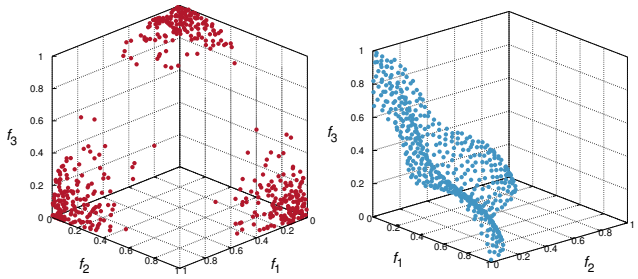
To be as comprehensive as possible, the taxonomy scheme in Figure 1 also lists the individual visualization methods belonging to each category together with their references.

3 VISUALIZING SINGLE APPROXIMATION SETS

3.1 Approximation Sets

We illustrate the methods for visualizing single approximation sets with two sets of equal dimensionality and size, but different characteristics. They can be regarded as solutions to two distinct multiobjective optimization problems where all objectives need to be minimized. While only the instantiation with four objectives and 500 points is used for visualization in this section, the procedures for constructing these sets are general and can be used for any dimension m and number of points n.

3.1.1 The Spherical Set. The spherical set contains points in the first orthant (the m-dimensional analogue to the first quadrant) of the m-dimensional hypersphere of radius 1. The points are concentrated in m clusters around the m standard basis vectors. Each cluster is constructed by generating points according to the von Mises-Fisher distribution [31] centered around the corresponding standard basis vector and with the concentration parameter κ set to 25. Any points located outside of the first orthant are discarded, therefore the generation of points (performed with the Ulrich-Wood algorithm [49]) is continued until the desired number of points in each cluster is generated. A 3-D instantiation of the spherical set with 500 points is shown in Figure 2 (a).



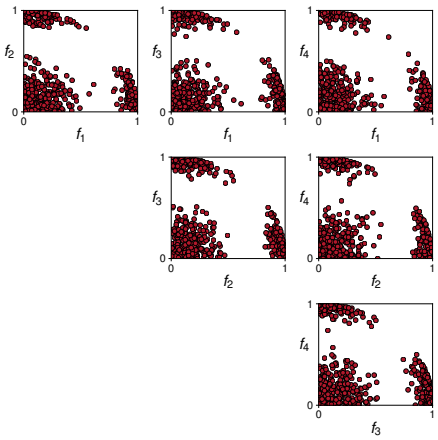
(a) The spherical set (b) The knee-shaped set

Figure 2: 3-D counterparts to the 4-D approximation sets used for comparing visualization methods in Section 3.

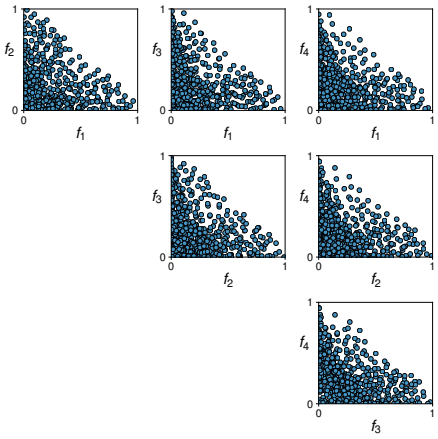
3.1.2 The Knee-Shaped Set. The second set consists of evenly distributed points that lie on the knee-shaped Pareto front of the scaled DEBmDK problem for $K = 1$. The scaled problem is obtained by normalizing the original DEBmDK problem [8] so that its Pareto front is in $[0, 1]^m$. The set is constructed by sampling the optimal solutions to the scaled problem using the Mitchell's best-candidate algorithm [32] that works as follows. To create each point, 100 candidate samples are first generated. Then, the candidate that has the largest minimal distance to the already existing points in the set is added to the set. A 3-D instantiation of the knee-shaped set with 500 points is shown in Figure 2 (b).

3.2 Visualizing Individual Solutions

3.2.1 Showing Original Values of Solutions. An example of a method showing the original values of solutions is the scatter plot matrix that projects the points from the objective space to a selected lower-dimensional space by disregarding all other dimensions. All possible combinations of these lower-dimensional spaces form the scatter plot matrix. Most frequently, scatter plots in a 2-D space are used for projection, where for m objectives $\frac{m(m-1)}{2}$ different combinations are possible (see Figure 3 for four objectives). The scatter plot matrix is a fast, simple and robust visualization method.



(a) The spherical set



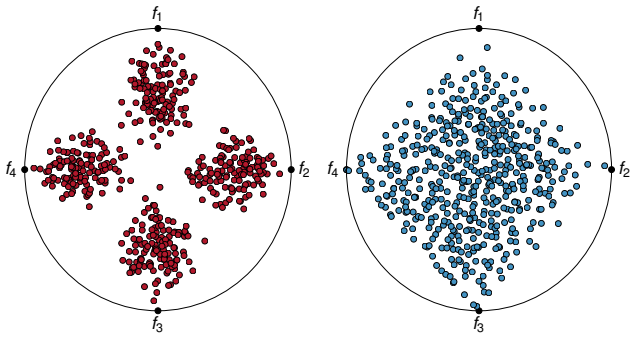
(b) The knee-shaped set

Figure 3: Scatter plot matrix.

Visualizing our test approximation sets with the scatter plot matrix, we can observe the loss of information because of the dimensionality reduction. For the clustered spherical set only three out of four clusters are seen, while for the knee-shaped set the information on even distribution of the points is lost. Both effects are due to the overlapping points in the visualized space. A way to alleviate this shortcoming to a certain degree would be the use of colors or interactivity.

Other methods in this category include various general multidimensional data visualization methods, such as the bubble chart, the radar chart, the parallel coordinates [21], the chord diagram [25], and the heat maps [35], as well as a specialized method called the interactive decision maps [29].

3.2.2 Showing Transformed Values of Solutions. Transformed values of solutions are, for example, shown by the Radial Coordinate Visualization, also known as RadViz [19]. It is inspired by spring mechanics. The objectives (called dimensional anchors) are distributed evenly on the circumference of the unit circle. One can then imagine that each point in the objective space is held with springs attached to the anchors and the spring force is proportional to the value of the corresponding objective. The point positions where the spring forces are in equilibrium. As illustrated by our test examples in Figure 4, the RadViz is capable of preserving the distribution of points in the approximation sets, but not the shape of the set.



(a) The spherical set (b) The knee-shaped set

Figure 4: Radial coordinate visualization.

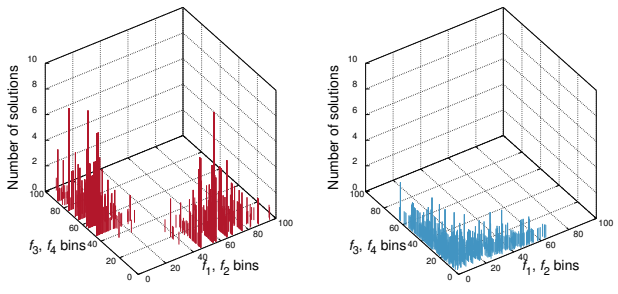
While the RadViz is a general data visualization method, additional methods in this category were proposed specifically for visualizing approximation sets. They comprise the 3-D radial coordinate visualization [20], the tetrahedron coordinates model [5], the polar plots [18], the hyper-radial visualization [10], the level diagrams [6, 7], and the prosections [42].

3.3 Visualizing Set Properties

3.3.1 Showing Individual Solution Properties. According to the proposed taxonomy, the methods showing the properties of individual solutions from an approximation set are further divided into a subcategory involving no optimization when generating a visualization, and another subcategory where visualization is based on an optimization procedure. We illustrate the former with the hyper-space diagonal counting [1] and the latter with the Sammon mapping [36].

Hyper-space diagonal counting is inspired by Cantor's proof that the set of natural numbers $\mathbb{N}$ has the same cardinality as the set $\mathbb{N}^m$, where $m \in \mathbb{N}$. As a consequence, the set $\mathbb{N}^m$ can be mapped into $\mathbb{N}$ using the hyper-space diagonal counting presented in [2] and visualized as further suggested in [1]. We demonstrate the approach on the case of 4-D approximation sets. Each objective is first divided into a predefined number of bins, 10 in our example. Next the bins of a pair of objectives are counted according to hyper-space diagonal counting. This produces indices for this pair of objectives. Finally, the indices are plotted on the first two axes, while the third axis is used to plot the number of points of the approximation set appearing in the same set of bins.

The hyper-space diagonal counting visualization of our test sets is shown in Figure 5. In the case of the clustered spherical set, the loss of information is evident. One cannot recognize the original clustering of points as the visualization results in two clusters only.



(a) The spherical set (b) The knee-shaped set

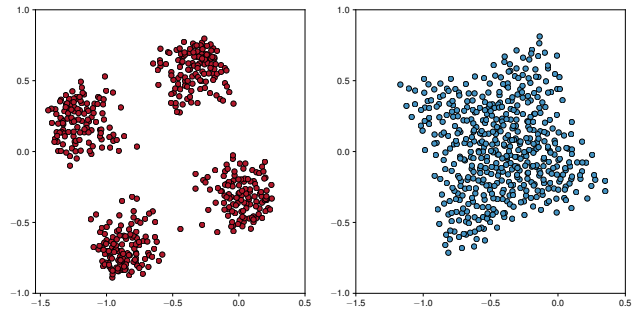
Figure 5: Hyper-space diagonal counting.

On the other hand, the uniform distribution of points in the knee-shaped set is better preserved and, to some degree, their distance to the ideal point is maintained too.

Additional visualization menthods involving no optimization procedure include the distance and distribution charts [3], the Pareto shells [46], the treemaps [45], and the trade-off region maps [34].

As an optimization-based method, Sammon mapping [36] aims to keep distances between the mapped points as similar as possible to distances between the original points. Specifically, it minimizes a stress function reflecting the preservation of local distances. The stress function is defined as $S = \sum_i \sum_{j>i} (d^*_{ij} - d_{ij})^2$, where d^*_{ij} is the distance between points $\mathbf{x}_i$ and $\mathbf{x}_j$ in the objective space, and d_{ij} the distance between the two points in the visualized space. The function can be minimized either by gradient descent or by other (iterative) optimization methods. While Sammon mapping was designed as a general data visualization technique, its use in multiobjective optimization was reported in [44].

For our test approximation sets the Sammon mapping preserves the distribution of points to a high degree (Figure 6). With respect to this feature the Sammon mapping is comparable with the RadViz. What is however lost in Sammon mapping, but partially preserved in RadViz is the information on the relative position of points in the objective space.



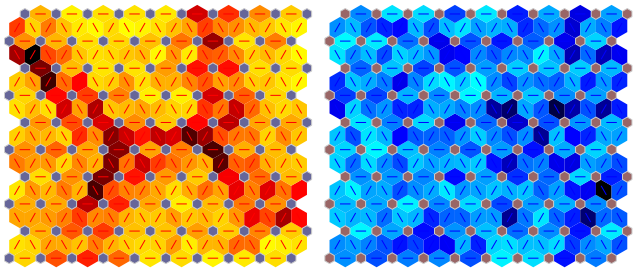
(a) The spherical set (b) The knee-shaped set

Figure 6: Sammon mapping.

A number of methods fall in the optimization-based category: the principal component analysis [50], the neuroscale [13, 30], the multidimensional scaling [47], the isomap [27, 38], the seriated heatmaps [47], the two-stage mapping [26], and the distance-based and dominance-based mappings [14].

3.3.2 Showing Aggregated Properties. Aggregated properties of approximation sets can, for instance, be visualized using the Self-Organizing Maps (SOMs) [24]. These are artificial neural networks aimed at providing a topology-preserving mapping from an m-D space to a lower-dimensional (typically 2-D) space, such that nearby points in the input space are mapped to nearby units (neurons). Among various methods suitable to visualize the SOMs, the most frequently used is the unified distance matrix (U-matrix) that presents the distance between adjacent neurons using a color scale. Light areas denote clusters of similar neurons, while dark areas correspond to cluster boundaries. Illustrating their performance, we use the SOMs with neurons arranged on the hexagonal grid [33].

Figure 7 shows the visualization of the two test sets. The SOMs well present the distribution of points in both cases. One can see the four clusters of the spherical set and the uniform distribution of points in the knee-shaped set.



(a) The spherical set (b) The knee-shaped set

Figure 7: Self-organizing maps.

Besides the SOMs, the aggregation trees [11] and the moGrams [40] fit into the category of visualization methods showing aggregated properties of approximation sets.

4 VISUALIZING REPEATED APPROXIMATION SETS

4.1 Repeated Approximation Sets

Repeated approximation sets are represented by two groups of spherical sets that have different radii and distributions of points—clustered and uniform. Moreover, algorithm convergence is mimicked by adding information on the time when each point of a set was created (expressed as the consecutive number) and translating the points so that the initial points are away from the front and approach the front as time increases. This results in two additional groups of sets—clustered sets with a logarithmic convergence and uniform sets with a linear convergence. For the purpose of this study, each group contains five sets and each set contains 100 points. The sets of the first two groups are instantiated in 3-D, while the

sets of the last two groups are instantiated in 2-D, matching the capabilities of the corresponding visualization methods.

4.1.1 Clustered Sets. This group consists of spherical sets with a non-uniform distribution of solutions, i.e., the same kind of sets as the ones already presented in Section 3.1.1. The radii of these five sets were determined randomly following the normal distribution with mean 0.9 and standard deviation 0.05.

4.1.2 Uniform Sets. The five sets in this group have spherical shape and a uniform distribution of vectors constructed by following the procedure detailed in Section 3.1.2. The radii of these sets were set randomly following the normal distribution with mean 1.0 and standard deviation 0.05.

4.1.3 Clustered Sets with a Logarithmic Convergence. This group is constructed based on the clustered spherical sets. The logarithmic convergence is mimicked by randomly sorting all points of a set and then assigning to each point in turn a new position along the same direction from the ideal point, but with a different distance to the ideal point. This distance starts at the value of 2.0 and is diminished to 1.0 following a logarithmic-like convergence behavior—faster at the beginning and slower at the end. Each point in a set has an associated 'time' that equals its consecutive number in this procedure. Note that the resulting sets contain many dominated points.

4.1.4 Uniform Sets with a Linear Convergence. This group is formed analogously to the procedure described in Section 4.1.3 with the difference that the underlying sets have a uniform distribution of points and the convergence is linear. For the first point, the distance to the ideal point equals 1.8, which is then linearly diminished until reaching 0.9 for the last point.

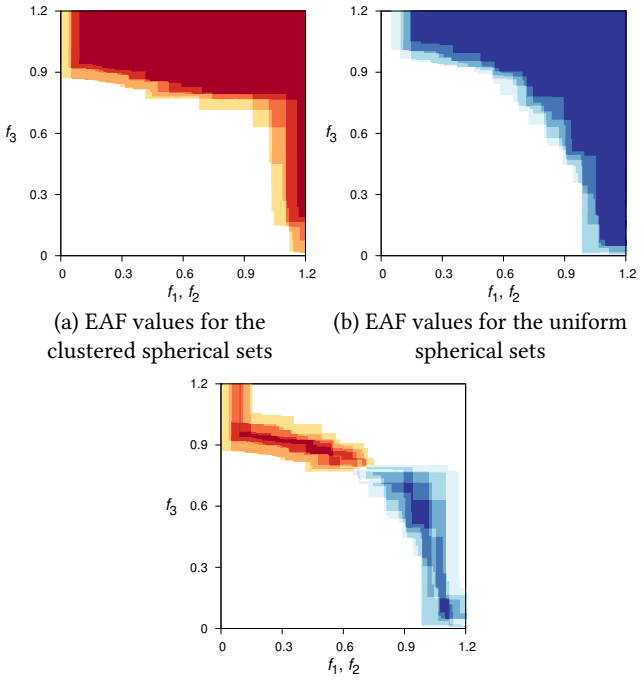
4.2 Showing Performance at a Time

Assume an algorithm was run r times, producing r approximation sets. For each point in the objective space, the EAF value equals the frequency of attaining that point by these approximation sets [16]. That is, the EAF value is a number in $\{0, 1/r, 2/r, \ldots, (r-1)/r, 1\}$ assigned to each point in the objective space. This means that the objective space is partitioned into areas of different EAF values. The boundaries between those areas are called *attainment surfaces*. The EAF can be valuable also when comparing two algorithms. The differences between the EAF values of two algorithms show areas in which one algorithm performs differently (or equally) to the other.

Visualization of the EAF values (or differences in EAF values) can take two forms—visualization of the attainment surfaces and visualization of the attained areas. A crucial consideration that needs to be made when visualizing EAF values is the dimensionality of the objective space. If the objective space has two dimensions, attainment surfaces are lines and can be plotted with simple line plots [15]. Similarly, the areas with equal EAF values are rectangles and can be plotted using heat maps [28]. In 3-D, however, the surfaces are facets of rectangular cuboids and the areas are rectangular cuboids, which is challenging for computation as well as visualization. This is why a sensible approach to visualization of EAF values in the 3-D objective space is to employ approximation of the elements to be visualized.

The first approach to visualization of 3-D attainment surfaces was a grid-based sampling of the surface [23], while an exact way of visualizing the facets was used only very recently [17]. On the other hand, the first (and so far only) methods to visualize 3-D attained areas were presented in [41]. Some (slicing and maximum intensity projection [48]) can be applied to visualize exact rectangular cuboids, while others (direct volume rendering [12]) can only be used if the objective space is discretized into voxels (spatial pixels).

Slicing [41] cuts through the objective space under an angle and visualizes the resulting intersection between the cutting plane and the 3-D attained areas with a 2-D heat map. Figure 8 shows the slices of EAF values and differences in those values for the clustered and uniform sets when cutting through the plane f_1, f_2 under the angle of 45°. Slicing through the objective space under multiple angles is required to gain an understanding of the algorithm performance in the entire objective space.



(a) EAF values for the clustered spherical sets

(b) EAF values for the uniform spherical sets

(c) Differences in EAF values between the two groups (red/blue hues show better attainment by the clustered/uniform sets, while white areas were attained equally well by both groups)

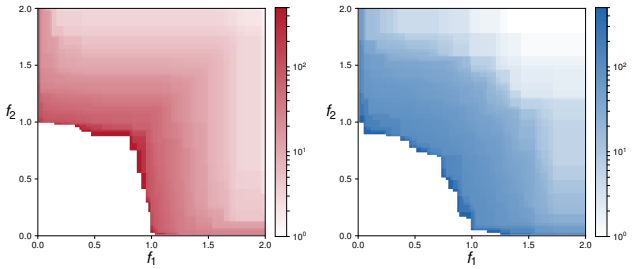
Figure 8: EAF values (top) and their differences (bottom) for the 3-D groups of clustered and uniform spherical sets visualized using slicing of the plane f_1, f_2 under the angle of 45°. Darker colors correspond to higher absolute values.

4.3 Showing Performance over Time

The aRTA function [9] can be seen as a generalization of the EAF. Instead of counting only the frequencies of attaining objective space points, the aRTA value equals the average time (measured

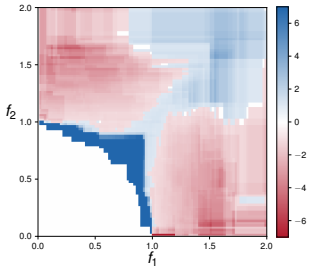
in number of evaluations) needed to attain the objective space points. This means that it is able to capture the performance of an algorithm over multiple runs *and* over time. Analogous to the differences in EAF values, the ratio of aRTA values can be used to compare performances of two algorithms.

As the EAF, the aRTA function assigns values to objective space points and the elements to be visualized are essentially the same—objective space areas. However, because aRTA records the entire optimization process, not just its final best solutions, the resulting partition of the objective space is much finer than the one created by the EAF. This can be very demanding already for small sets, therefore [9] proposes to visualize aRTA values using grid-based sampling. So far, the aRTA values have only been visualized for 2-D objective spaces, but the same methods capable of visualizing approximated 3-D EAF values could also be used for visualizing approximated 3-D aRTA values. Figure 9 presents the grid-based sampling visualization of aRTA values and ratios between those values for the 2-D clustered sets with a logarithmic convergence and the uniform sets with a linear convergence.



(a) aRTA values for the clustered sets with a logarithmic convergence

(b) aRTA values for the uniform sets with a linear convergence



(c) Ratios of aRTA values between the two groups (red/blue hues show better attainment by the clustered/uniform sets, while white areas were attained equally well by both groups)

Figure 9: aRTA values (top) and their ratios (bottom) for the 2-D groups of clustered sets with a logarithmic convergence and uniform sets with a linear convergence using grid-based sampling with a grid of 100×100 points. Darker colors correspond to higher absolute values.

5 CONCLUSIONS

In multiobjective optimization many approaches are used to visualize the results, ranging from general-purpose data visualization methods to specialized methods designed to serve the specific needs of visualizing the datasests in multiobjective optimization. The number of specialized approaches rapidly grows in the last years. To upgrade the previous straightforward classifications known from the literature, we have devised a more detailed taxonomy of methods for visualizing Pareto front approximations. The taxonomy builds on the nature of the visualized data and the properties of visualization methods rather than on the employed visual representations. It covers the methods for visualizing individual approximation sets resulting from a single algorithm run as well as multiple approximation sets produced in repeated runs. The proposed taxonomy categories are characterized, equipped with the lists of corresponding methods and illustrated with selected examples of visualization methods. In these examples, the test approximation sets created specifically in this work are used to show some aspects of the methods.

Some observations that immediately follow from the new taxonomy are as follows. It is evident that the visualization of single approximation sets has been investigated much more intensively than the visualization of repeated approximation sets. As a result, many more methods exist for the first task than for the second one. On the other hand, the numerous methods of visualizing single approximation sets can be used to visualize the repeated sets too if appropriately utilized, e.g., in animations. Further development in this direction would certainly increase the potentials of the methodology.

Moreover, a more detailed analysis of strenghts and weaknesses of the approaches, which was not possible here due to space limitation, would help better understand the methodology and allow for deeper insight in when and how to use individual methods or their combinations. However, in pursuing this goal it is important that methods are first analyzed on solution sets with known properties, such as the ones used in this work. An additional example is a study of visualizing knee-shaped 4-D Pareto front approximations with various methods reported in [43]. Working with benchmarks of this kind will make comparison of visualization methods easier, as it is the case with the test suites used to compare the optimization algorithms.

In conclusion, we expect the proposed taxonomy will be insightful to the multiobjective optimization community and will make communication on the visualization methods among the participants easier. Specifically, the practitioners may be able to better navigate among the many categories of the visualization methods and their properties, while the researchers developing new visualization methods may find it easier to better focus their further efforts.

ACKNOWLEDGMENTS

We acknowledge financial support from the Slovenian Research Agency (research core funding no. P2-0209 and project no. Z2-8177). This work is also part of a project that has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement no. 692286. The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions.

REFERENCES

[1] G. Agrawal, C. L. Bloebaum, and K. Lewis. 2005. Intuitive design selection using visualized n-dimensional Pareto frontier. In *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*. American Institute of Aeronautics and Astronautics, 1813:1–1813:14.

[2] G. Agrawal, K. Lewis, K. Chugh, C.-H. Huang, S. Parashar, and C. L. Bloebaum. 2004. Intuitive visualization of Pareto frontier for multi-objective optimization in n-dimensional performance space. In *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. American Institute of Aeronautics and Astronautics, 4434:1–4434:11.

[3] K. H. Ang, G. Chong, and Y. Li. 2002. Visualization technique for analyzing non-nominated set comparison. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, SEAL'02, Singapore*. Nanyang Technological University, 36–40.

[4] S. Bandaru, A. H. C. Ng, and K. Deb. 2017. Data mining methods for knowledge discovery in multi-objective optimization: Part A – Survey. *Expert Systems with Applications* 70 (2017), 139–159.

[5] X. Bi and B. Li. 2012. The visualization decision-making model of four objectives based on the balance of space vector. In *Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, Vol. 2. 365–368.

[6] X. Blasco, J. M. Herrero, J. Sanchis, and M. Martínez. 2008. A new graphical visualization of n-dimensional Pareto front for decision-making in multiobjective optimization. *Information Sciences* 178, 20 (2008), 3908–3924.

[7] X. Blasco, G. Reynoso-Meza, E. A. Sánchez Pérez, and J. V. Sánchez Pérez. 2016. Asymmetric distances to improve n-dimensional Pareto fronts graphical analysis. *Information Sciences* 340-341 (2016), 228–249.

[8] J. Branke, K. Deb, H. Dierolf, and M. Oswald. 2004. Finding knees in multi-objective optimization. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature, PPSN VIII*. 722–731.

[9] D. Brockhoff, A. Auger, N. Hansen, and T. Tušar. 2017. Quantitative performance assessment of multiobjective optimizers: The average runtime attainment function. In *Proceedings of the 9th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2017*. 103–119.

[10] P.-W. Chiu and C. Bloebaum. 2010. Hyper-radial visualization (HRV) method with range-based preferences for multi-objective decision making. *Structural and Multidisciplinary Optimization* 40, 1–6 (2010), 97–115.

[11] A. R. R. de Freitas, P. J. Fleming, and F. G. Guimarães. 2015. Aggregation trees for visualization and dimension reduction in many-objective optimization. *Information Sciences* 298 (2015), 288–314.

[12] K. Engel, M. Hadwiger, J. M. Kniss, C. Rezk-Salama, and D. Weiskopf. 2006. *Real-Time Volume Graphics*. A. K. Peters, Natick, MA, USA.

[13] R. M. Everson and J. E. Fieldsend. 2006. Multi-class ROC analysis from a multi-objective optimisation perspective. *Pattern Recognition Letters* 27, 8 (2006), 918–927.

[14] J. E. Fieldsend and R. M. Everson. 2013. Visualising high-dimensional Pareto relationships in two-dimensional scatterplots. In *Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2013*. 558–572.

[15] C. M. Fonseca and P. J. Fleming. 1996. On the performance assessment and comparison of stochastic multiobjective optimizers. In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature, PPSN IV*. 584–593.

[16] V. D. Grunert da Fonseca, C. M. Fonseca, and A. O. Hall. 2001. Inferential performance assessment of stochastic optimisers and the attainment function. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, EMO 2001*. 213–225.

[17] A. P. Guerreiro, C. M. Fonseca, and L. Paquete. 2016. Greedy hypervolume subset selection in low dimensions. *Evolutionary Computation* 24, 3 (2016), 521–544.

[18] Z. He and G. G. Yen. 2016. Visualization and performance metric in many-objective optimization. *IEEE Transactions on Evolutionary Computation* 20, 3 (2016), 386–402.

[19] P. E. Hoffman, G. G. Grinstein, K. Marx, I. Grosse, and E. Stanley. 1997. DNA visual and analytic data mining. In *Proceedings of the IEEE Conference on Visualization*. IEEE, 437–441.

[20] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb. 2016. 3D-RadVis: Visualization of Pareto front in many-objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2016*. 736–745.

[21] A. Inselberg. 2009. *Parallel Coordinates: Visual Multidimensional Geometry and its Applications*. Springer, New York, NY, USA.

[22] J. Kehrer and H. Hauser. 2013. Visualization and visual analysis of multifaceted scientific data: A survey. *IEEE Transactions on Visualization and Computer Graphics* 19, 3 (2013), 495–513.

[23] J. Knowles. 2005. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In *Proceedings of the 5th International Conference on Intelligent Systems Design and Applications, ISDA '05*. IEEE, 552–557.

[24] T. Kohonen. 2001. *Self-Organizing Maps*. Springer-Verlag, Berlin, Germany.

[25] R. H. Koochaksaraei, I. R. Meneghini, V. N. Coelho, and F. G. Guimarães. 2017. A new visualization method in many-objective optimization with chord diagram and angular mapping. *Knowledge-Based Systems* 138 (2017), 134–154.

[26] M. Köppen and K. Yoshida. 2007. Visualization of Pareto-sets in evolutionary multi-objective optimization. In *Proceedings of the 7th International Conference on Hybrid Intelligent Systems, HIS 2007*. IEEE, 156–161.

[27] F. Kudo and T. Yoshikawa. 2012. Knowledge extraction in multi-objective optimization problem based on visualization of Pareto solutions. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2012*. IEEE, 1–6.

[28] M. López-Ibáñez, L. Paquete, and T. Stützle. 2010. Exploratory analysis of stochastic local search algorithms in biobjective optimization. In *Experimental Methods for the Analysis of Optimization Algorithms*. Springer, 209–222.

[29] A. V. Lotov, V. A. Bushenkov, and G. K. Kamenev. 2004. *Interactive Decision Maps: Approximation and Visualization of Pareto Frontier*. Kluwer Academic Publishers, Boston, MA, USA.

[30] D. Lowe and M. E. Tipping. 1996. Feed-forward neural networks and topographic mappings for exploratory data analysis. *Neural Computing & Applications* 4, 2 (1996), 83–95.

[31] K. V. Mardia and P. E. Jupp. 2008. *Directional Statistics*. John Wiley & Sons, Inc.

[32] D. P. Mitchell. 1987. Generating antialiased images at low sampling densities. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1987*. ACM, 65–72.

[33] S. Obayashi and D. Sasaki. 2003. Visualization and data Mining of Pareto solutions using self-organizing map. In *Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003*. 796–809.

[34] R. L. Pinheiro, D. Landa-Silva, and J. Atkin. 2015. Analysis of objectives relationships in multiobjective problems using trade-off region maps. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 2015*. 735–742.

[35] A. Pryke, S. Mostaghim, and A. Nazemi. 2007. Heatmap visualisation of population based multi objective algorithms. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007*. 361–375.

[36] J. W. Sammon. 1969. A nonlinear mapping for data structure analysis. *IEEE Trans. Comput.* C-18, 5 (1969), 401–409.

[37] B. Shneiderman. 1996. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the IEEE Symposium on Visual Languages*. IEEE, 336–343.

[38] J. B. Tenenbaum, V. de Silva, and J. C. Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 5500 (2000), 2319–2323.

[39] M. Tory and T. Möller. 2004. Rethinking visualization: A high-level taxonomy. In *Proceedings of the 10th IEEE Symposium on Information Visualization, InfoVis 2004*. 151–158.

[40] K. Trawinski, M. Chica, D. P. Pancho, S. Damas, and O. Cordón. 2018. moGrams: A network-based methodology for visualizing the set of non-dominated solutions in multiobjective optimization. *IEEE Transactions on Cybernetics* 48, 2 (2018), 474–485.

[41] T. Tušar and B. Filipič. 2014. Visualizing exact and approximated 3D empirical attainment functions. *Mathematical Problems in Engineering* 2014 (2014), 18. Article ID 569346.

[42] T. Tušar and B. Filipič. 2015. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation* 19, 2 (2015), 225–245.

[43] T. Tušar and B. Filipič. 2016. Showing the knee of a 4-D Pareto front approximation via different visualization methods. In *Proceedings of the 19th International Multiconference Information Society, IS 2016*, Vol. A. 52–55.

[44] J. Valdés and A. Barton. 2007. Visualizing high dimensional objective spaces for multi-objective optimization: A virtual reality approach. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007*. IEEE, 4199–4206.

[45] D. J. Walker. 2015. Visualising multi-objective populations with treemaps. In *Companion Material Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '15*. ACM, 963–970.

[46] D. J. Walker, R. M. Everson, and J. E. Fieldsend. 2010. Visualisation and ordering of many-objective populations. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010*. IEEE, 1–8.

[47] D. J. Walker, R. M. Everson, and J. E. Fieldsend. 2013. Visualizing mutually nondominating solution sets in many-objective optimization. *IEEE Transactions on Evolutionary Computation* 17, 2 (2013), 165–184.

[48] J. W. Wallis, T. R. Miller, C. A. Lerner, and E. C. Kleerup. 1989. Three-dimensional display in nuclear medicine. *IEEE Transactions on Medical Imaging* 8, 4 (1989), 297–230.

[49] A. T. A. Wood. 1994. Simulation of the von Mises Fisher distribution. *Communications in Statistics - Simulation and Computation* 23, 1 (1994), 157–164.

[50] M. Yamamoto, T. Yoshikawa, and T. Furuhashi. 2010. Study on effect of MOGA with interactive island model using visualization. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010*. IEEE, 1–6.