Visualizing 4D Approximation Sets of Multiobjective Optimizers with Prosections

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

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

ABSTRACT

In ideal multiobjective optimization, the result produced by an optimizer is a set of nondominated solutions approximating the Pareto optimal front. Visualization of this approximation set can help assess its quality as well as present various features of the problem. Most often, scatter plots are used to visualize 2D and 3D approximation sets, while no scatter plot equivalent exists for visualization in higher dimensions. This paper presents a method for visualizing 4D approximation sets which performs dimension reduction using prosections (projections of a section). The method yields a prosection matrix—a matrix of intuitive 3D scatter plots that well reproduce the shape, range and distribution of vectors in the observed approximation set. The performance of visualization with prosections is analyzed theoretically and demonstrated on two examples with approximation sets of state-of-the-art test optimization problems.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization

General Terms

Performance

Keywords

Multiobjective optimization, Visualization, Dimension Reduction

1. INTRODUCTION

In multiobjective optimization we wish to simultaneously optimize several (possibly conflicting) objectives. This can be achieved by means of a *multiobjective optimizer*—an algorithm that finds an approximation of the Pareto optimal front, called *approximation set*. An approximation set consists of objective vectors that are nondominated with regard to each other, each representing a different trade-off between the objectives. There exist many measures to assess the quality of an approximation set (i.e. how well it approximates the Pareto optimal front in terms of distance, spread and distribution of objective vectors). However, no measure is as effective as the visualization of the approximation set, especially if the Pareto optimal front is known and can be visualized as well.

Visualization in multiobjective optimization is essential in many aspects—it can be used to [8]: present various features of the problem, estimate the location, range and shape of the Pareto optimal front, assess conflicts and trade-offs between objectives, select preferred solutions, monitor the progress or convergence of an optimization run, or assess the relative performance of different optimizers. When tackling optimization problems with two or three objectives, the most straightforward (as well as the most often used) way to visualize an approximation set is by plotting its vectors on a scatter plot. However, when the number of objectives $m \geq 4$, a simple and intuitive visualization of approximation sets is much harder to achieve.

In this paper we present a method that can visualize 4D approximation sets in 3D in an intuitive way using *prosection* (*pro*jection of a *section* [7]). The method is simple and yet powerful—the resulting 3D visualization can be used for all purposes mentioned in the previous paragraph. In particular, it reproduces well the shape, range and distribution of vectors in the observed approximation set thus facilitating comparison between different optimizers. The method is analyzed in theory and presented in practice on approximation sets of state-of-the-art test optimization problems.

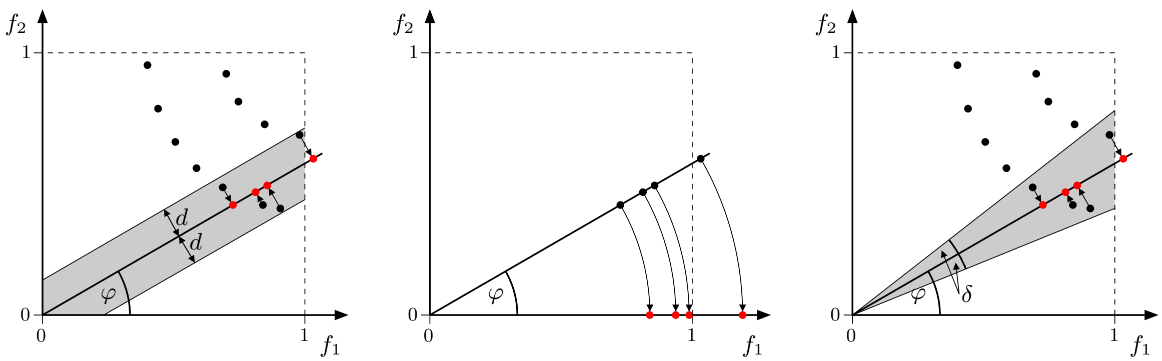
The remainder of this paper is organized as follows. Section 2 reviews the related work, while visualization using prosections is described in Section 3 and illustrated with two examples in Section 4. The paper concludes with a discussion in Section 5 and final remarks in Section 6.

2. RELATED WORK

2.1 Visualizing Approximation Sets

Existing techniques for visualization of high dimensional approximation sets can be divided in two groups. The first group consists of methods that use dimension reduction to map approximation sets to 2D or 3D, like for example, scatter plot matrix, distance and distribution charts [2], self-organizing maps [11], interactive decision maps [10], neuroscale [5], virtual reality [15], two-stage mapping by Köppen [9], level diagrams [3], and most recently, Pareto shells [17]. These methods are very different from one another,

GECCO'11, July 12–16, 2011, Dublin, Ireland.



(a) 2D($f_1f_2, 30°, 0.1$) (b) rotation (c) 2D($f_1f_2, 30°, 8°$)

Figure 1: Graphical presentation of: (a) prosection, (b) rotation and (c) a prosection alternative.

each trying to emphasize a different aspect of the visualized approximation set. For example, level diagrams try to retain the vectors' proximity to the ideal point, while the two-stage mapping by Köppen preserves as many Pareto dominance relations among the vectors as possible. Although successful in their intent and scalable to many objectives, these methods do not provide an intuitive visualization, which could be used, for example, to assess the relative performance of different optimizers.

The second group comprises methods such as parallel coordinates [6], scatter plots with addition of size or color (also called bubble charts) [12], heatmaps [13] and hyper-space diagonal counting [1]. These methods try to retain and visualize all the information at once. Unfortunately, in this way the achieved presentations usually loose clarity and comprehensibility.

2.2 Slices and Sections

More than the visualization methods used in multiobjective optimization, our method resembles two methods for visualization from other domains. These are the HyperSlice for visualizing scalar functions of many variables [16] and the prosection matrix, which is used for externalizing abstract mathematical models [14].

With HyperSlice, a multi-dimensional function is presented with a matrix of slices. When a point of interest in the mD space is selected, the function can be represented as a matrix of orthogonal 2D slices of fixed width. This representation facilitates interactive navigation, location of optima and marking of the explored path.

The prosection matrix is basically a scatter plot matrix with the exception that only a portion (section) of the space is used in the orthogonal projection to the corresponding 2D plane. The ranges of the section can be adjusted interactively and color coding is used for distinguishing between feasible and infeasible solutions.

3. VISUALIZATION WITH PROSECTIONS

Although there are many techniques for visualizing 4D approximation sets, none of them can be regarded as a scaled scatter plot with all of its benefits—a clear and informative presentation of the shape, range and distribution of vectors in the observed approximation set. Moreover, despite all these new visualization possibilities, most researchers in the field of multiobjective optimization still resort to parallel coordinates when a 4D (or higher) approximation set is to be shown. This was our motivation for presenting a new visualization method that reduces one dimension of the approximation set using prosection (projection of a section) and rotation. Although it is currently suitable only for 3D and 4D approximation sets, it trades-off generality for simplicity and clarity.

Let us first present the assumptions for this method. We assume (without loss of generality) that all the objectives of the multiobjective optimization problem need to be minimized. Furthermore, the approximation set must be normalized to the interval $[0, 1]^m$. Both assumptions are needed only to facilitate the presentation. In particular, we will see how to work around the second assumption in the examples (see Subsection 4.2).

3.1 Prosection

As mentioned before, a prosection is a projection of a section (the term was introduced in [7]). Here, the section on the 2D plane f_1f_2 is defined by the angle φ and width d (see Figure 1 (a)). Each vector within the section is orthogonally projected to the line crossing the origin and intersecting the f_1-axis at angle φ using:

$$p : (f_1, f_2) \mapsto (f_1', f_2'),$$

where

$$\begin{aligned} f_1' &= \cos\varphi(f_1 \cos\varphi + f_2 \sin\varphi), \\ f_2' &= \sin\varphi(f_1 \cos\varphi + f_2 \sin\varphi). \end{aligned}$$

Prosection projects all the vectors in the section to the chosen line, while all other vectors are ignored. This prosection can be uniquely denoted as a 3-tuple

$$m\text{D}(f_1f_2, \varphi, d),$$

where m is the number of objectives of the original approximation set, f_1f_2 indicates the plane on which the prosection takes place, φ denotes the chosen angle, and d is the width of the section.

3.2 Rotation

After prosection, the line with projected vectors needs to be rotated so that this truly becomes a reduction in dimension (see Figure 1 (b)):

$$r : (f_1', f_2') \mapsto \sqrt{(f_1')^2 + (f_2')^2}.$$

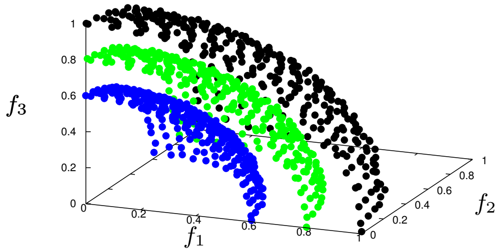


Figure 2: Three spherical approximation sets in 3D.

3.3 Method Features

When composing prosection and rotation, the transformation simplifies to:

$$r(p(f_1, f_2)) = f_1 \cos\varphi + f_2 \sin\varphi.$$

This yields a very simple algorithm for visualizing approximation sets with prosections:

1. Choose the angle φ and section width d. These parameters define the section.
2. All vectors within the section are projected using one of the following functions:

   3D case: $(f_1, f_2, f_3) \mapsto (f_1 \cos\varphi + f_2 \sin\varphi, f_3)$

   4D case: $(f_1, f_2, f_3, f_4) \mapsto (f_1 \cos\varphi + f_2 \sin\varphi, f_3, f_4)$

3. All vectors outside the section are ignored.

Note that prosection and rotation influence only two objectives (f_1 and f_2), while all the others remain intact. Moreover, for arbitrary angle φ and section width d, the following holds:

$$m\mathrm{D}(f_2 f_1, \varphi, d) \equiv m\mathrm{D}(f_1 f_2, (\pi/2 - \varphi), d),$$

which means, for example, that the prosection on the plane $f_1 f_2$ with angle 30° is equivalent to the prosection on the plane $f_2 f_1$ with angle 60°.

Let us demonstrate how this method works when projecting 3D approximation sets to 2D on the example of three spherical approximation sets from Figure 2. Plots (a), (b) and (c) in Figure 3 show the spherical approximation sets after prosection and before rotation for three different values of section width d. As expected, a wider section includes more vectors than a thiner one. To enable a good visualization of the approximation set, the selection width d should be chosen with care.

Essentially, what the method does is slicing through the approximation set at an angle φ and looking at this slice in one less dimension. In the upper plots in Figure 3, rotation has not been performed yet to show how the prosection with the angle of 45° cuts the plane $f_1 f_2$ in the middle. The basic difference between this method and the slices and sections presented in Subsection 2.2 is in the angle φ, which was either 0 or $\pi/2$ in previous work. The fact that an angle φ different from 0 and $\pi/2$ can be used, leads to an important property of this method:

THEOREM 1. *If the angle* $\varphi \in (0, \pi/2)$ *and the vector* $(f_1^{\mathrm{A}}, f_2^{\mathrm{A}})$ *dominates the vector* $(f_1^{\mathrm{B}}, f_2^{\mathrm{B}})$*, then the projected vector* $r(p(f_1^{\mathrm{A}}, f_2^{\mathrm{A}}))$ *dominates* $r(p(f_1^{\mathrm{B}}, f_2^{\mathrm{B}}))$*.*

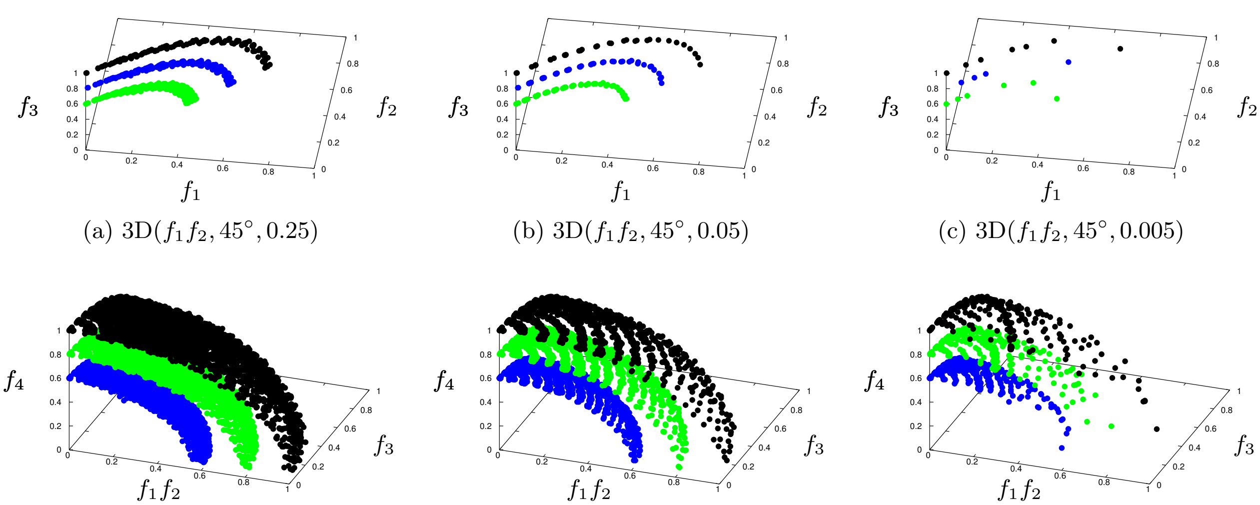
PROOF. Suppose the vector $(f_1^{\mathrm{A}}, f_2^{\mathrm{A}})$ dominates the vector $(f_1^{\mathrm{B}}, f_2^{\mathrm{B}})$. This means that $f_1^{\mathrm{A}} \leq f_1^{\mathrm{B}}$, $f_2^{\mathrm{A}} \leq f_2^{\mathrm{B}}$ and $(f_1^{\mathrm{A}}, f_2^{\mathrm{A}}) \neq (f_1^{\mathrm{B}}, f_2^{\mathrm{B}})$. Also, suppose that $\varphi \in (0, \pi/2)$. Then $\sin\varphi > 0$ and $\cos\varphi > 0$.

$$\begin{aligned} & r(p(f_1^{\mathrm{B}}, f_2^{\mathrm{B}})) - r(p(f_1^{\mathrm{A}}, f_2^{\mathrm{A}})) = \\ &= \underbrace{(f_1^{\mathrm{B}} - f_1^{\mathrm{A}})}_{\geq 0} \underbrace{\cos\varphi}_{>0} + \underbrace{(f_2^{\mathrm{B}} - f_2^{\mathrm{A}})}_{\geq 0} \underbrace{\sin\varphi}_{>0} \end{aligned}$$

Since $(f_1^{\mathrm{B}} - f_1^{\mathrm{A}})$ and $(f_2^{\mathrm{B}} - f_2^{\mathrm{A}})$ cannot both be equal to 0,

$$r(p(f_1^{\mathrm{B}}, f_2^{\mathrm{B}})) - r(p(f_1^{\mathrm{A}}, f_2^{\mathrm{A}})) > 0,$$

which means that $r(p(f_1^{\mathrm{A}}, f_2^{\mathrm{A}})) < r(p(f_1^{\mathrm{B}}, f_2^{\mathrm{B}}))$, or in other words, $r(p(f_1^{\mathrm{A}}, f_2^{\mathrm{A}}))$ dominates $r(p(f_1^{\mathrm{B}}, f_2^{\mathrm{B}}))$. □



(a) $3\mathrm{D}(f_1 f_2, 45^\circ, 0.25)$ (b) $3\mathrm{D}(f_1 f_2, 45^\circ, 0.05)$ (c) $3\mathrm{D}(f_1 f_2, 45^\circ, 0.005)$

(d) $4\mathrm{D}(f_1 f_2, 45^\circ, 0.25)$ (e) $4\mathrm{D}(f_1 f_2, 45^\circ, 0.05)$ (f) $4\mathrm{D}(f_1 f_2, 45^\circ, 0.005)$

Figure 3: Prosections of the three spherical approximation sets with different section width.

This means that if one vector dominates the other, the dominance relation is retained after prosection and rotation. However, this is not the case for nondominance: if two vectors are nondominated with regard to each other, after the transformation one vector can dominate the other. Note that the new 'objective' f_1f_2 that is formed in the prosection still needs to be minimized.

Now, let us focus on the 4D case. Plots (d), (e) and (f) in Figure 3 show 3D prosections of three 4D spherical approximation sets. Note that they resemble very much the 3D spherical approximation sets, thus showing that using prosections we can achieve an intuitive visualization of high dimensional approximation sets. It is again obvious how the width of the section influences the number of vectors included in the visualization.

In both examples (prosection of the 3D and 4D spherical approximation sets), the value of $d = 0.05$ found good results—it gives just enough projected vectors for representing the shape of the approximation set without overcrowding it with redundant vectors. Therefore, we will use a fixed d of 0.05 in the rest of the experiments.

3.4 Prosection Matrix

So far we always predefined the plane in which the prosection takes place to f_1f_2. Since in general other planes might be of interest, it is sensible to show a prosection matrix where all possible planes are taken into account. For this purpose, half of the prosection matrix suffices. However, by using the whole matrix we can explore two different angles φ at the same time. Such prosection matrices will be presented in Section 4.

3.5 Section Variants

When defining the section, two alternatives can be used in place of the fixed width d.

The section can be defined using the number of vectors N to be included in the projection. In this way, only the N vectors that are closest to the chosen line are projected using the function p, while all other vectors are ignored. This alternative can be denoted with

$$m\mathrm{D}(f_1f_2, \varphi, N).$$

The section can be bounded by the angle δ (see Figure 1 (c)). The function p is the same as before. Note that this alternative is biased since the section gets wider further away from the origin. This means that close to the origin, the projection would capture less vectors than further away. This alternative can be denoted with

$$m\mathrm{D}(f_1f_2, \varphi, \delta).$$

Throughout this paper, only the original definition of the section with width d (see Subsection 3.1) is used.

4. EXAMPLES

Visualization of approximation sets using prosections is further investigated using two examples.

4.1 Two Approximation Sets

In the first example, we use two approximation sets of different shapes (see Figure 4): one linear and one spherical (following the shapes of fronts of DTLZ test problems [4]). The two approximation sets are intertwined—in some regions the spherical approximation set dominates the linear one and vice versa. With this example we want to show that this method can be used to visually compare two (or more) approximation sets at the same time.

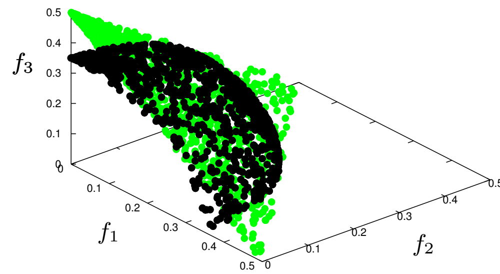


Figure 4: A linear and a spherical approximation set in 3D.

Note that the approximation sets are not normalized to the $[0, 1]^m$ interval. Since all objectives have the same range, there is no need to alter the prosection and rotation functions. With the help of the prosection matrix, we can explore many views at once: all possible planes are used for prosection and since the whole matrix is used, two angles φ are explored: $10°$ and $45°$. The width of the section d is set to 0.05. Figures 5 and 6 present prosections matrices of the 3D and 4D approximation sets, respectively.

The prosection matrix of the 3D approximation sets shows three important things. First, since the shapes of both approximation sets are symmetrical, there is virtually no difference among the plots with different chosen planes. Second, when the angle $\varphi = 10°$ for the line is chosen, the approximation set of linear shape crosses this line at a small angle, producing a prosection with more spread out vectors than when the angle $\varphi = 45°$. This indicates that for some approximation sets, the section definition influences the outcome of the projection. This influence will be investigated in more detail in further work. Finally, the plots are very clear and informative and can easily be used for comparing the two approximation sets.

Similar conclusions can be drawn for the 4D case. The plots are very similar independently of the plane used for prosection and even the choice of a different angle does not produce much different plots. Correspondingly to the 3D approximation sets, the plots are very explanatory. Because it is easy to see in which regions one approximation set dominates the other, such prosections can help in a comparison analysis.

4.2 A Discontinuous Approximation Set

In the second example, a single approximation set of discontinuous shape (as in the DTLZ test suite) is visualized. We use colors to help differentiate between its regions (see Figure 7). In 3D the approximation set consists of four discontinuous regions (with vectors in black, green, blue and red full circles).

Moreover, the objectives in this example are disproportionate. In 3D, $f_1, f_2 \in [0, 0.9]$, while $f_3 \in [2.6, 6]$, and in 4D, $f_1, f_2, f_3 \in [0, 1]$ and $f_4 \in [2.9, 8]$. To show the versatility of this method, we move the origin of the observed plot $\mathbf{a}$ to $[0, 0, 2.6]$ in 3D and $[0, 0, 0, 2.9]$ in 4D. This entails the following altering of the transformation:

$$r(p(f_1, f_2)) = (f_1 - a_i)\cos\varphi + (f_2 - a_j)\sin\varphi,$$

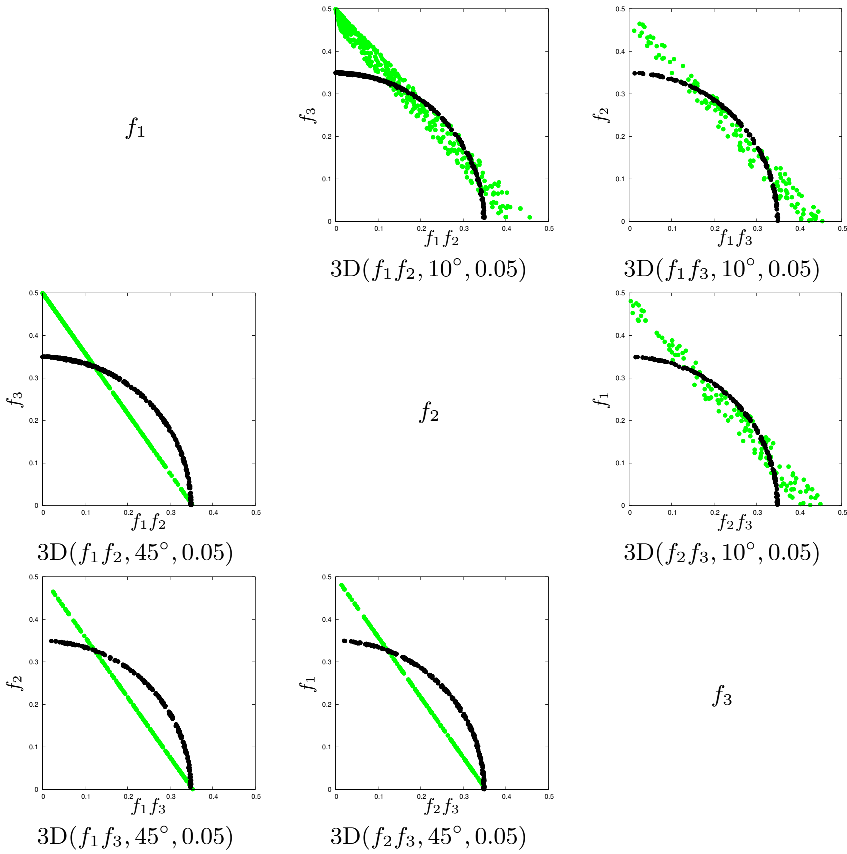


Figure 5: Prosection matrix for the 3D linear and spherical approximation sets.

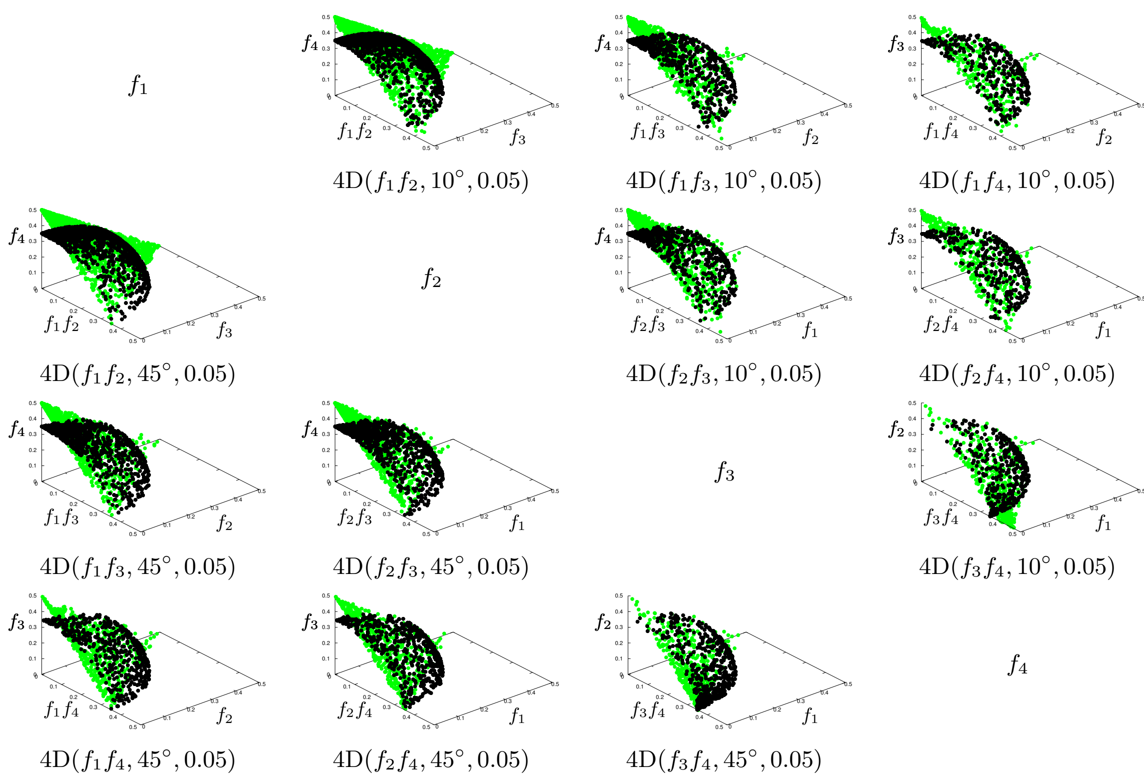


Figure 6: Prosection matrix for the 4D linear and spherical approximation sets.

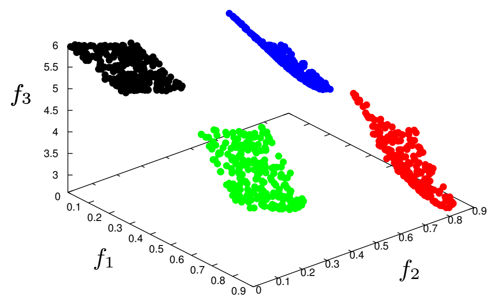


Figure 7: A discontinuous approximation set in 3D. The regions are colored for facilitating comprehension of the prosections.

where $i, j = 1, \ldots, m$ and $i \neq j$. Because the last objective has a different range than the others, when the last objective is included in the prosection, a line at 45° does not cut the plane in the middle. Therefore, the angles need to be adjusted. In this example, instead of the 10° and 45° angles, we use 33.7° and 75.2° in the 3D case and 42° and 78.9° in the 4D case, so that the new angles correspond to 10° and 45° if the approximation sets would have been normalized to $[0, 1]^m$.

Again, we use a prosection matrix to display the results of the transformation (see Figures 8 and 9). Let us first focus on the 3D case. The prosection matrix enables a clear and understandable insight into the way the method works. For example, take the prosection $3D(f_1 f_3, 75.2°, 0.05)$ shown in the left plot in the third row of the matrix. The line at angle 75.2° cuts the plane $f_1 f_3$ in the middle and intersects the green and red regions. If the smaller angle of 33.7° is used (see the right plot in the first row of the matrix), only the red region is caught in the prosection. The approximation set is not symmetric as in the first example, therefore the prosections are very different from each other.

In the 4D case, the approximation set consists of eight regions. They are distinguished using vectors of different colors and marks (black, green, blue and red full and empty circles). While all the prosections without the fourth objective cut through four regions, with the help of colors and marks it is clear that these are not always the same regions. Although it is still hard to imagine how the approximation set looks in 4D, this visualization method makes it possible to view the range, spread and distribution of vectors in the approximation set in a straightforward way.

5. DISCUSSION

To discuss the proposed visualization method, let us look at its advantages and disadvantages.

5.1 Advantages

First of all, the visualization is simple, intuitive and informative. If we lean on the purposes of visualization in multiobjective optimization that were presented in the Introduction, we can see that the presented visualization serves them all. It can be used to: present various features of the problem, estimate the location, range and shape of the Pareto optimal front, assess conflicts and trade-offs between objectives, select preferred solutions, monitor the progress or convergence of an optimization run, and assess the relative performance of different optimizers.

While we focused mainly on the section with fixed width, two other alternatives for defining the section were presented. The second alternative, which uses the angle δ to bound the section is biased and should probably not be used. However, the first alternative with the fixed number of vectors in the section can be very useful, especially if the number of vectors to be visualized is known in advance.

The method is very fast. For an approximation set of n vectors, the prosection and rotation can be achieved with computational complexity of $O(n)$.

Instead of using a static prosection matrix, the visualization can be made even more effective with the use of a computer program that enables interactively changing the angle and section width[1].

5.2 Disadvantages

Some existing visualization methods, such as parallel coordinates [6] and heatmaps [13], can be used to visualize the decision space as well as the objective space. This is not the case with visualization using prosections as it can be reasonably applied only in the objective space.

So far, the method can only be used to visualize up to 4D approximation sets. While it is clear that a 5D approximation set can be visualized by applying prosection twice, further work is needed to analyze how to do this so that the simplicity and versatility of the method are not compromised. Although currently applicable only up to 4D, we believe the method gives a new insight into this space, which is an important contribution by itself.

Because we are visualizing only a slice of an mD approximation set, a lot of information is disregarded when looking at a single prosection plot. To get a complete picture, it is important to view the whole prosection matrix under many angles. Also, as seen from the example of two approximation sets (Subsection 4.1), for some approximation sets the visualization result depends on how the section is chosen. Further work is needed to gain additional knowledge on this issue. Moreover, when using the section of a fixed width, the width must be determined. On the basis of the performed experiments, we suggest to first try with a width of 0.1 and then adjust it according to the results.

As Tweedie et al. stated in [14]: 'there is a trade-off between the amount of information, simplicity and accuracy'. When conceiving this method, we willingly sacrificed some information to get a simple and accurate visualization.

6. CONCLUSION

Visualization in multiobjective optimization is very important. It can be used to present various features of the problem or to reveal information about the multiobjective optimizer being used. While it is easy to visualize 2D and 3D approximation sets, an intuitive representation for approximation sets in higher dimensions is harder to achieve. Although there are many techniques for visualizing 4D approximation sets, none of them can be regarded as a scaled scatter plot with all of its benefits—a clear and informative

[1]See `http://dis.ijs.si/tea/GECCO2011/prosection.htm` for additional visualizations of the approximation sets used in this paper, which include animation using `gnuplot`.

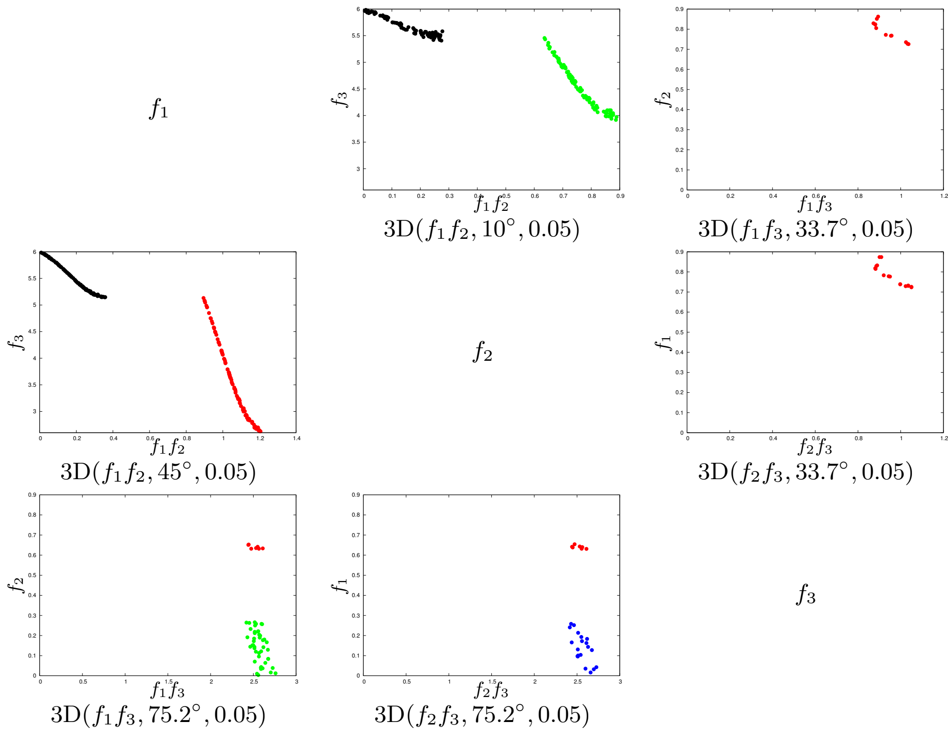


Figure 8: Prosection matrix for the 3D discontinuous approximation set.

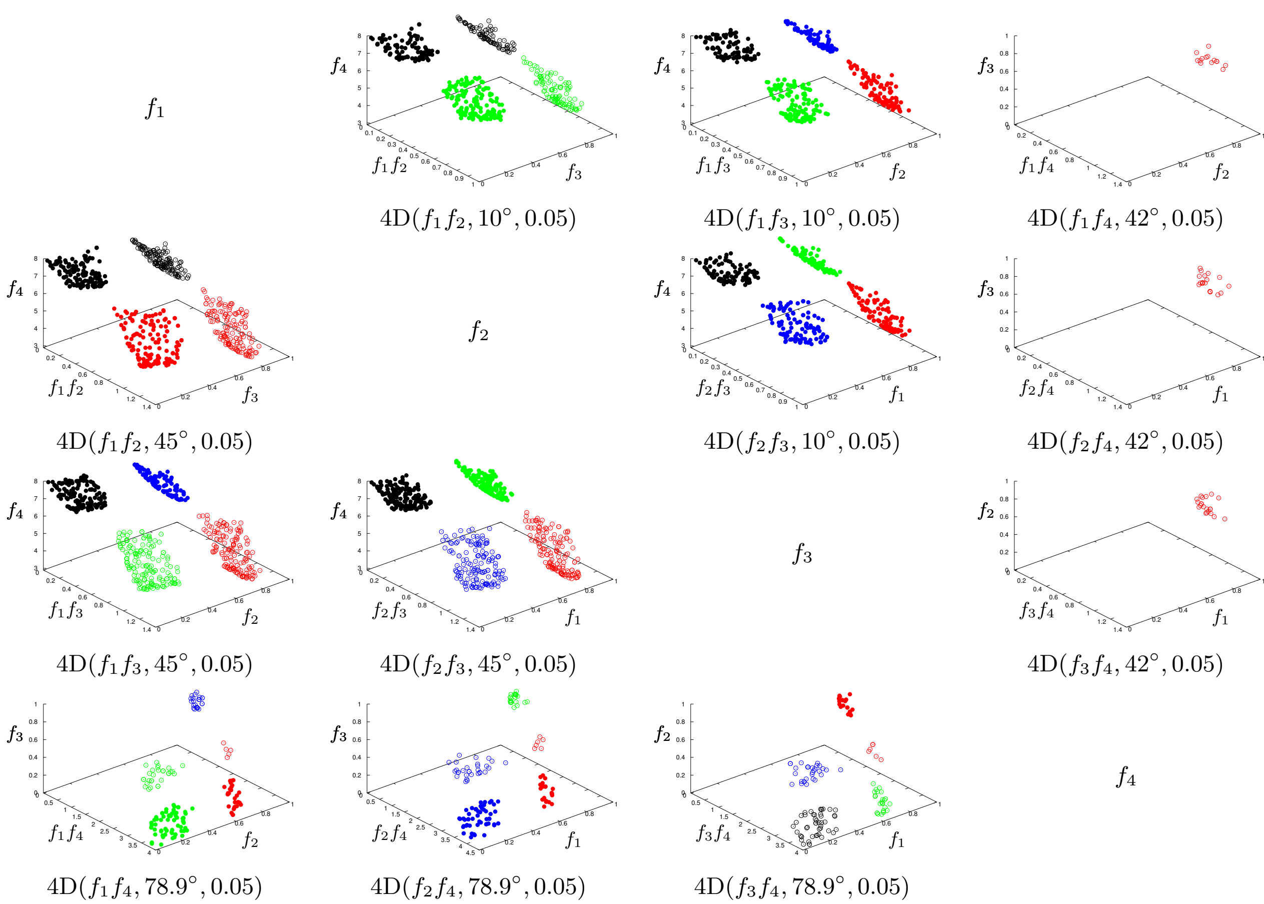


Figure 9: Prosection matrix for the 4D discontinuous approximation set.

presentation of the shape, range and distribution of vectors in the observed approximation set.

We presented a new method for visualizing 4D approximation sets that trades wholeness for clarity. Using prosections to a line that is not necessarily parallel to any of the coordinates the method reduces one dimension of the approximation set while keeping many of its features intact. Specifically, if this line intersects the coordinates with an angle $\varphi \in (0, \pi/2)$, the prosection preserves the dominance relation between all pairs of vectors where one vector dominates the other.

Experiments on approximation sets of different shapes have shown that the method can effectively visualize the shape, range and distribution of vectors in the observed approximation sets. It can be used for analyzing the performance of a single multiobjective optimizer as well for comparison of two or more multiobjective optimizers.

Since the research on this method is still in an early stage, further work in several directions is needed. Most importantly, the best way to employ prosections to efficiently tackle problems in more than 4D has to be investigated. Also, an analysis of the influence of section definition on visualization results for differently shaped approximation sets is needed. Furthermore, the presented alternatives for section definition as well as guidelines for suggested parameter settings should be studied in greater detail. Finally, the practical use of this method would benefit from an accompanying software supporting exploration of the entire approximation set by simply scrolling through the prosections from angle 0 to $\pi/2$.

7. ACKNOWLEDGMENTS

The work presented in this paper has been carried out under research programme P2-0209 and research project L2-3651, both funded by the Slovenian Research Agency. The authors wish to thank the Walking Fish Group for making the approximation sets for the DTLZ test problems available online (`http://www.wfg.csse.uwa.edu.au/`).

8. REFERENCES

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

[2] K. H. Ang, G. Chong, and Y. Li. Visualization technique for analyzing non-dominated set comparison. In *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02)*, volume 1, pages 36–40, 2002.

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

[4] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, R. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, chapter 6, pages 105–145. Springer, 2005.

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

[6] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 416–423, 1993.

[7] G. W. Furnas and A. Buja. Prosection views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics*, 3(4):323–353, 1994.

[8] J. Knowles, D. Corne, and K. Deb. Introduction: Problem solving, EC and EMO. In J. Knowles, D. Corne, and K. Deb, editors, *Multiobjective Problem Solving from Nature: From Concepts to Applications*, pages 1–28. Springer, 2008.

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

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

[11] S. Obayashi and D. Sasaki. Visualization and data mining of Pareto solutions using self-organizing map. In *Proceedings of the Second International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science 2632*, pages 796–809, 2003.

[12] S. Poles, P. Geremia, F. Campos, S. Weston, and M. Islam. MOGA-II for an automotive cooling duct optimization on distributed resources. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science 4403*, pages 633–644, 2007.

[13] A. Pryke, S. Mostaghim, and A. Nazemi. Heatmap visualisation of population based multi objective algorithms. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science 4403*, pages 361–375, 2007.

[14] L. Tweedie, R. Spence, H. Dawkes, and H. Su. Externalising abstract mathematical models. In *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, CHI '96, pages 406–412, 1996.

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

[16] J. J. van Wijk and R. van Liere. Hyperslice: Visualization of scalar functions of many variables. In *Proceedings of the 4th conference on Visualization '93*, VIS '93, pages 119–125, 1993.

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