Initial Experiments in Visualization of Empirical Attainment Function Differences using Maximum Intensity Projection

Tea Tušar
Department of Intelligent Systems
Jožef Stefan Institute, and
Jožef Stefan International Postgraduate School
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
tea.tusar@ijs.si

Bogdan Filipič
Department of Intelligent Systems
Jožef Stefan Institute, and
Jožef Stefan International Postgraduate School
Jamova cesta 39
SI-1000 Ljubljana, Slovenia
bogdan.filipic@ijs.si

ABSTRACT

In multiobjective optimization, the Empirical Attainment Function (EAF) can be used to determine which areas of the objective space are attained by an optimization algorithm. If two algorithms are to be compared, differences in EAF values show which areas of the objective space are more often attained by one of the algorithms. While the visualization of EAF values and differences is rather straightforward in 2D, the 3D case presents a great challenge as we need to visualize a large number of 3D cuboids. This paper presents a method for computing the cuboids with constant EAF values and reports on initial experiments using Maximum Intensity Projection, a very-well known volume rendering technique used in medicine.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Multiobjective optimization, Empirical attainment function, Visualization, Maximum intensity projection

1. INTRODUCTION

Real-world optimization problems often consist of two, three or even more conflicting objectives that cannot be simply aggregated into a single value. Therefore, the optimization algorithms used for solving these problems offer a set of trade-off solutions, where no solution from the set is better than any other in all objectives. This is called an *approximation set*. If the algorithm is stochastic, as is the case with evolutionary multiobjective algorithms generally used for such problems [1], it returns a different approximation set in every run.

Whether constructing a new optimization algorithm or improving an existing one, researchers need tools to assist them in the comparison between algorithms. While the performance of algorithms can be (and most often is) measured using different performance metrics, visualization of their approximation sets can give important insight into the properties of the algorithms (or the problem at hand) and inspire ideas for further improvement. In two- and three-objective optimization, the easiest way to visualize an approximation set is by simply plotting each solution it contains[1]. Such plots can also be used to visually compare two or more approximation sets, for example, each from a different optimization algorithm. However, there is a limitation to this technique—if an optimization algorithm is run multiple times and the resulting approximation sets are to be visualized simultaneously, the produced output can become too cluttered and loose its interpretation potential.

This difficulty can be overcome with the use of the Empirical Attainment Function (EAF), which assigns to each vector in the objective space a value signifying how often the vector was attained by the given approximation sets [3]. In the 2D case, the performance of an algorithm represented by several approximation sets can be visualized by plotting rectangles of different colors (or gray shades) representing different EAF values. Similarly, two algorithms can be compared by showing differences in these values [4, 6]. Such plots are very informative, providing an easy to understand visualization of the areas of the objective space where one algorithm outperforms the other one (and vice versa).

In the 3D case, however, visualization of EAF values and differences is less trivial. The attained areas are not rectangles, but cuboids, or rather, unions of cuboids, which need to be computed first. If the approximation sets of one algorithm consistently outperform the approximation sets of another algorithm, the positive EAF differences between the two algorithms look similar to a 3D 'cloud' of cuboids where the cuboids with the highest values lie in the center of the cloud, surrounded by cuboids with gradually decreasing values. Direct volume rendering of such clouds would show only the outer cuboids that correspond to smaller differences be-

[1] Approximation sets of higher dimensions that require more sophisticated methods to be visualized will not be discussed in this paper. The interested reader is referred to [7] for a comprehensive review of such methods.

GECCO'14, July 12–16, 2014, Vancouver, BC, Canada.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2881-4/14/07 ...$15.00.
http://dx.doi.org/10.1145/2598394.2610548.

tween the algorithms and are thus less interesting than the inner cuboids. This paper tackles the task of visualizing clouds of cuboids so that the cuboids with highest values have precedence over those with lower values. In previous work, we have sliced through the clouds of EAF values and differences under different angles and visualized the slices obtained in this way [6]. Here, we limit ourselves to the visualization of EAF differences using the Maximum Intensity Projection (MIP) method [8], which gives us a kind of 'X-ray' visualization of the cloud, where the cuboids with highest EAF differences are best seen. The analogy with X-rays is not coincidental as MIP is mostly used in medical imaging [5]. The MIP visualization nicely complements the slices, and taken together, the visualizations give a good idea of the 3D cloud of cuboids.

The next section provides some background on the relations used in multiobjective optimization, the EAF and the MIP method. Section 3 is devoted to solving a subproblem in the visualization of EAF differences—the computation of the attained area, which in turn requires finding the *opposite* of an approximation set. Section 4 demonstrates how MIP can be used for visualizing EAF differences. Finally, Section 5 concludes the paper with ideas for future work.

2. BACKGROUND

2.1 Relations in Multiobjective Optimization

The multiobjective optimization problem consists of finding the optimum of a function

$$\boldsymbol{f}\colon X \to F$$
$$\boldsymbol{f}\colon (x_1, \dots, x_n) \mapsto (f_1(x_1, \dots, x_n), \dots, f_m(x_1, \dots, x_n)),$$

where X is an n-dimensional decision space, and F is an m-dimensional objective space ($m \geq 2$). Each solution $\boldsymbol{x} \in X$ is called a decision vector, while the corresponding element $\boldsymbol{z} = \boldsymbol{f}(\boldsymbol{x}) \in F$ is an objective vector. Without loss of generality we assume that $F \subseteq \mathbb{R}^m$ and all objectives $f_i\colon X \to \mathbb{R}$ are to be minimized.

As this paper deals with visualization in the objective space, which can be viewed rather independently from the decision space, the following definitions are confined to the objective space.

Definition 1 (**Weak Pareto dominance of vectors**). The objective vector $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \dots, z_m^{\mathrm{A}})$ *weakly dominates* the objective vector $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \dots, z_m^{\mathrm{B}})$, i.e. $\boldsymbol{z}^{\mathrm{A}} \preceq \boldsymbol{z}^{\mathrm{B}}$, if

$$z_i^{\mathrm{A}} \leq z_i^{\mathrm{B}} \text{ for } \forall i \in \{1, \dots, m\}.$$

Definition 2 (**Strict Pareto dominance of vectors**). The objective vector $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \dots, z_m^{\mathrm{A}})$ *strictly dominates* the objective vector $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \dots, z_m^{\mathrm{B}})$, i.e. $\boldsymbol{z}^{\mathrm{A}} \prec\prec \boldsymbol{z}^{\mathrm{B}}$, if

$$z_i^{\mathrm{A}} < z_i^{\mathrm{B}} \text{ for } \forall i \in \{1, \dots, m\}.$$

Definition 3 (**Incomparability of vectors**). The objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \dots, z_m^{\mathrm{A}})$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \dots, z_m^{\mathrm{B}})$ are *incomparable*, i.e. $\boldsymbol{z}^{\mathrm{A}} \parallel \boldsymbol{z}^{\mathrm{B}}$, if

$$\boldsymbol{z}^{\mathrm{A}} \not\preceq \boldsymbol{z}^{\mathrm{B}} \text{ and } \boldsymbol{z}^{\mathrm{B}} \not\preceq \boldsymbol{z}^{\mathrm{A}}.$$

Definition 4 (**Approximation set**). A set of objective vectors $Z \subseteq F$ is called an *approximation set* if $\boldsymbol{z}^{\mathrm{A}} \parallel \boldsymbol{z}^{\mathrm{B}}$ for any two objective vectors $\boldsymbol{z}^{\mathrm{A}}, \boldsymbol{z}^{\mathrm{B}} \in Z$.

Definition 5 (**Weak Pareto dominance of approximation sets**). The approximation set Z^{A} *weakly dominates* the approximation set Z^{B}, i.e. $Z^{\mathrm{A}} \preceq Z^{\mathrm{B}}$, if every $\boldsymbol{z}^{\mathrm{B}} \in Z^{\mathrm{B}}$ is weakly dominated by at least one $\boldsymbol{z}^{\mathrm{A}} \in Z^{\mathrm{A}}$.

2.2 Empirical Attainment Function

The attainment function is based on the multiobjective concept of goal-attainment [3]: an objective vector is *attained* when it is weakly dominated by the approximation set returned by an optimization algorithm. If the algorithm is run r times, each objective vector can be attained between 0 and r times. The *empirical attainment function* of the objective vector $\boldsymbol{z}$ gives the frequency of attaining $\boldsymbol{z}$ by the approximation sets $Z_1, \dots, Z_r$:

$$\alpha_r(\boldsymbol{z}) = \frac{1}{r} \sum_{i=1}^{r} \boldsymbol{I}\left(Z_i \preceq \{\boldsymbol{z}\}\right),$$

where $\boldsymbol{I}$ is the indicator function, defined as

$$\boldsymbol{I}(b) = \begin{cases} 1 & \text{if } b \text{ is true,} \\ 0 & \text{otherwise,} \end{cases}$$

and $\preceq$ is the weak Pareto dominance relation between sets.

Every vector in the objective space is thus assigned an EAF value from the interval $[0, 1]$. Of course, in practice the EAF cannot be computed for every objective vector, but only the boundaries of the regions of the objective space with a constant EAF value are of interest. They are called *k %-attainment surfaces* (also *summary attainment surfaces*) and consist of the tightest objective vectors that have been attained in at least k % of the runs. The attainment surfaces are well-defined by a finite set of objective vectors that we call *attainment anchors*[2]. Attainment anchors include other vectors beside the vectors from the approximation sets. Fonseca et al. [2] provided efficient algorithms for computing attainment anchors and their EAF values for the 2D and 3D cases.

Consider the example shown in Figure 1, where three approximation sets contain four objective vectors each (denoted with circles). The lines represent three attainment surfaces that partition the objective space into areas with distinct EAF values. The attainment anchors define these surfaces and are shown with circles and dots (the dots are those that are not present in the approximation sets).

Now imagine that two algorithms, A and B, are run r times each. This yields two sets of approximation sets. The algorithms can be compared by computing their EAF values α_r^{A} and α_r^{B} separately. However, there exists a straightforward way to directly compare the algorithms by joining their approximation sets [4]. First, the overall attainment surfaces for the $2r$ approximation sets need to be computed. Then, for each attainment anchor $\boldsymbol{z}$, the *difference in EAF values* can be calculated by counting the number of times each of the algorithms managed to attain that anchor and subtracting the two numbers:

$$\delta^{\mathrm{A-B}}(\boldsymbol{z}) = r\alpha_r^{\mathrm{A}}(\boldsymbol{z}) - r\alpha_r^{\mathrm{B}}(\boldsymbol{z}).$$

Defined in this way, the differences can adopt values from $\{-r, \dots, 0, \dots, r\}$. Positive EAF differences denote areas in

[2] These vectors are called *output points* in the context of the EAF computation [2], where the *input points* represent the objective vectors from the approximation sets.

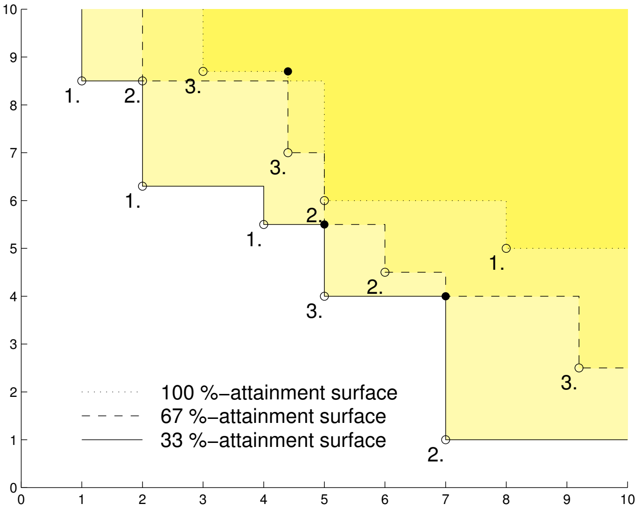

Figure 1: Attainment surfaces for three approximation sets. The circles labels denote the different approximation sets. All circles and dots represent the attainment anchors.

the objective space where the algorithm A outperforms the algorithm B, while negative EAF differences denote areas in the objective space where the algorithm B outperforms the algorithm A. Naturally, where the differences are zero, both algorithms attain the area equally well.

In order to visualize the EAF differences, the attained areas need to be computed. As presented later, this is easier to do in 2D, where these areas are unions of rectangles, than in 3D, where the areas are unions of cuboids, and require the computation of a opposite of an approximation set. Additional practical difficulties in visualizing EAF differences arise from the high number of attainment anchors l. As proven in [2], in the 2D case, $l \in \Theta(ra)$, while in the 3D case, $l \in O(r^2 a)$, where r is the number of approximation sets (runs) and a the total number of objective vectors contained in the approximation sets. In Section 4 we will see on an example how these bounds on the number of attainment anchors translate to actual figures.

2.3 Maximum Intensity Projection

Maximum intensity projection (MIP) is a volume rendering method for spatial data represented by voxels, i.e. values in a 3D grid. The method inspects voxels in direction of parallel rays traced from the viewpoint to the projection plane, and takes the maximum value encountered in the voxels along a ray as the projection value for the ray (see the schematic presentation in Figure 2).

The method, originally called maximum activity projection (MAP) [8], was proposed for 3D image rendering in nuclear medicine and tested in tomographic studies. It was later accepted not only in medical imaging, but in scientific data visualization in general.

The advantages of MIP are its simplicity and efficiency, and the ability of achieving high contrast, which arises from the fact that maximum voxel values are projected. On the other hand, as a limitation, the resulting projections lack the sense of depth of the original data. Moreover, the viewer cannot distinguish between left or right and front and back.

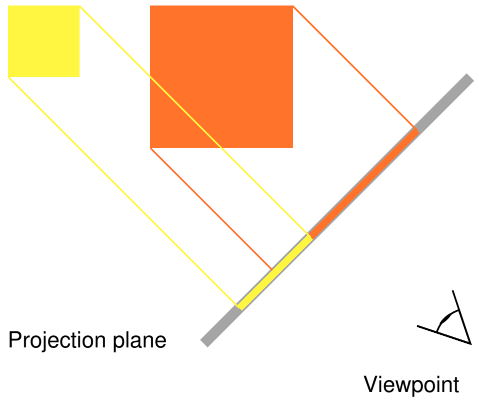

Figure 2: Schematic presentation of MIP.

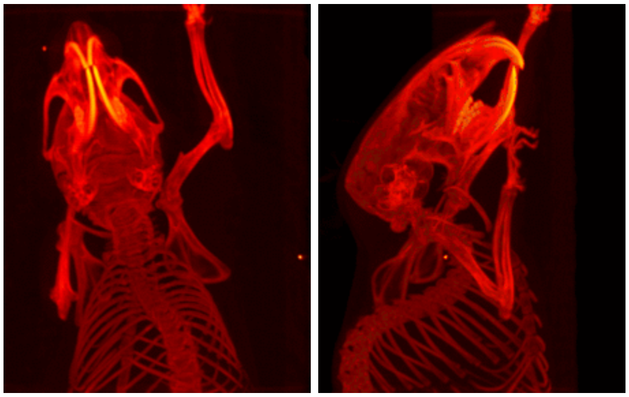

Figure 3: CT of a mouse visualized using MIP at two different viewpoints, ©Christian Lackas.

As an improvement, animations are usually provided, consisting of a sequence of MIP renderings at slightly different viewpoints, which results in the illusion of rotation (see an example of visualization using MIP at two different viewpoints in Figure 3).

3. COMPUTING THE ATTAINED AREA

Before tackling the visualization of EAF differences, let us focus on the problem of computing the areas attained by the anchors, which are unions of rectangles in 2D, and unions of cuboids in 3D. Consider two consecutive attainment surfaces Z_1 and Z_2. Each rectangle (or cuboid) between the two attainment surfaces is defined by exactly two vectors—one attainment anchor from Z_1 and a dominated vector from Z_2. We call these 'dominated vectors' the *opposite* of the set of anchors.

3.1 Opposite of an Approximation Set

The opposite O of an approximation set Z depends only on Z and two reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$ denoting the boundaries of the observed objective space ($\{\boldsymbol{r}^1\} \preceq Z \preceq \{\boldsymbol{r}^2\}$). The attainment surface of the approximation set Z divides the observed objective space in two areas—vectors in the first area are not attained by Z, while vectors in the second area are attained by Z. The opposite O is the set of vectors that would yield the same attainment surface, i.e.

Algorithm 1: Computing the opposite of a 2D approximation set.

```
Input: Approximation set $Z$ and reference vectors $\boldsymbol{r}^1$
       and $\boldsymbol{r}^2$ such that $\{\boldsymbol{r}^1\} \preceq Z \preceq \{\boldsymbol{r}^2\}$
Output: The opposite $O$

Sort the vectors from $Z$ in ascending order of the first
objective to obtain $Z = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^k\}$;
$O \leftarrow \{\}$;
if $z_1^1 \neq r_1^1$ then
    $O \leftarrow \{(z_1^1, r_2^2)\}$;
end
for $i \leftarrow 2$ to $k$ do
    $O \leftarrow O \cup \{(z_1^i, z_2^{i-1})\}$;
end
if $z_2^k \neq r_2^2$ then
    $O \leftarrow O \cup \{(r_1^2, z_2^k)\}$;
end
```

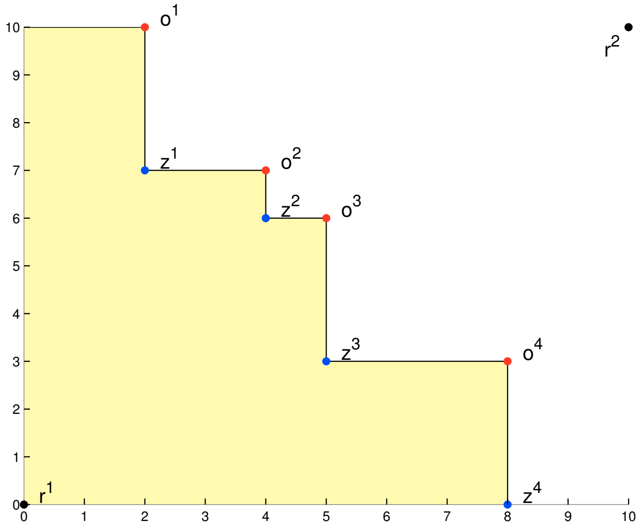

Figure 4: Example of a 2D approximation set $Z = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^4\}$ and its opposite $O = \{\boldsymbol{o}^1, \ldots, \boldsymbol{o}^4\}$ with regard to reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$.

divide the observed objective space in exactly the same areas, if the minimization problem would be transformed into a maximization problem.

In 2D, computing the opposite of an approximation set Z given two reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$ is very simple (see Algorithm 1 and Figure 4 for an example). First, the vectors in the approximation set Z are sorted with regard to their values in the first objective. Then for each vector $\boldsymbol{z}^i \in Z$, a vector $\boldsymbol{o}^i$ in the opposite O is created by 'raising' $\boldsymbol{z}^i$ to match the second objective value of its predecessor $\boldsymbol{z}^{i-1}$. The first vector has no predecessor, therefore it is raised to the level of the second reference vector $\boldsymbol{r}^2$. In addition, the last vector of the opposite is created by combining the first objective value of the $\boldsymbol{r}^2$ and the second objective value of the last vector from Z. The first and last vectors are retained in the opposite only if they are strictly dominated by the first reference vector $\boldsymbol{r}^1$ (this is the reason why there is no $\boldsymbol{o}^5 = (10, 0)$ in the example from Figure 4).

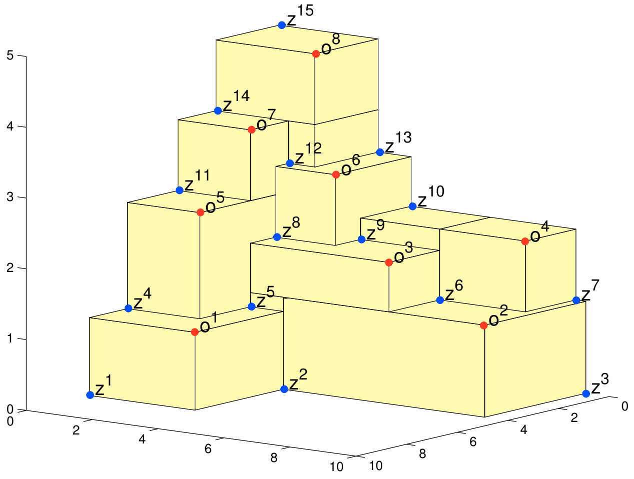

Figure 5: Example of a 3D approximation set $Z = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^{15}\}$ and its opposite $O = \{\boldsymbol{o}^1, \ldots, \boldsymbol{o}^8\}$ with regard to reference vectors $\boldsymbol{r}^1 = (0, 0, 0)$ and $\boldsymbol{r}^2 = (10, 10, 5)$ (not pictured).

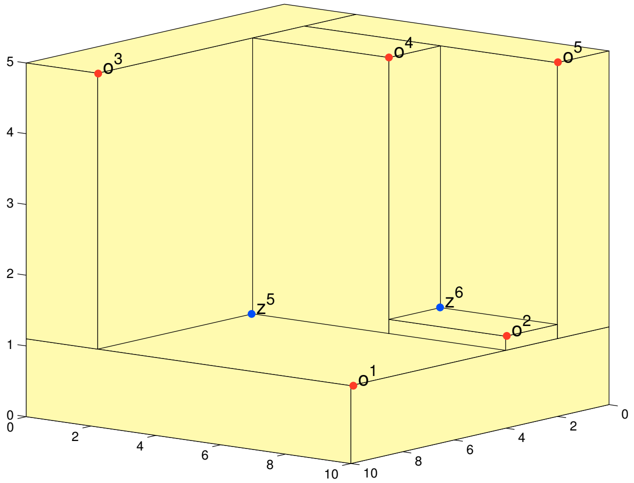

Figure 6: An intermediate step in the construction of the opposite of set Z from Figure 5 showing the opposite for the set $\{\boldsymbol{z}^5, \boldsymbol{z}^6\}$.

While finding the opposite of a 3D approximation set is more demanding, there exists an elegant way to do it. We will explain it with the help of Figures 5, 6 and 7. Again, we know the approximation set Z and two reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$. The desired outcome is shown in Figure 5. It can be constructed gradually, by taking into consideration one vector from Z at a time. Note that the vectors from Z do not need to be sorted.

Imagine a single cuboid defined by the reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$. The only vector in the opposite at this point is $\boldsymbol{r}^2$. Every time a vector from Z is 'cut into' the existing cuboid, new vectors are added into the opposite. Assume we have already performed this step for vectors $\boldsymbol{z}^5$ and $\boldsymbol{z}^6$ (see Figure 6) and vector $\boldsymbol{z}^9$ is next in line (see Figure 7). First, we delete from the opposite all vectors that are strictly dominated by the current vector $\boldsymbol{z}$ (this means that in our

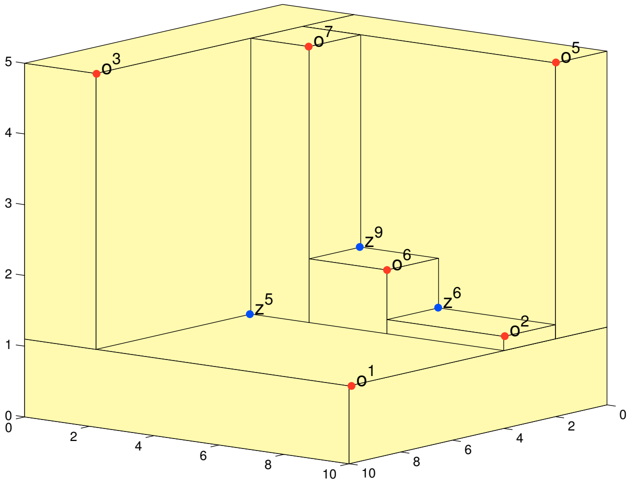

Figure 7: An intermediate step in the construction of the opposite of set Z from Figure 5 showing the opposite for the set $\{\boldsymbol{z}^5, \boldsymbol{z}^6, \boldsymbol{z}^9\}$.

Algorithm 2: Computing the opposite of a 3D approximation set.

```
Input: Approximation set $Z$ and reference vectors $\boldsymbol{r}^1$
       and $\boldsymbol{r}^2$ such that $\{\boldsymbol{r}^1\} \preceq Z \preceq \{\boldsymbol{r}^2\}$
Output: The opposite $O$

$O \leftarrow \{\boldsymbol{r}^2\}$;
foreach $\boldsymbol{z} \in Z$ do
    $O' \leftarrow \{\boldsymbol{o} \in O; \boldsymbol{z} \prec\prec \boldsymbol{o}\}$;
    $O \leftarrow O - O'$;
    foreach $\boldsymbol{o} \in O'$ do
        $\boldsymbol{o}^{n_1} \leftarrow (z_1, o_2, o_3)$;
        $\boldsymbol{o}^{n_2} \leftarrow (o_1, z_2, o_3)$;
        $\boldsymbol{o}^{n_3} \leftarrow (o_1, o_2, z_3)$;
        for $i \leftarrow 1$ to $3$ do
            if $\neg$ coplanar($\boldsymbol{o}^{n_i}, \boldsymbol{r}^1$) $\wedge \neg$ collinear($\boldsymbol{o}^{n_i}$, $O$)
            then
                $O \leftarrow O \cup \{\boldsymbol{o}^{n_i}\}$;
            end
        end
    end
end
```

Algorithm 3: Computing the attained areas between two attainment surfaces.

```
Input: Attainment surfaces $Z_1$ and $Z_2$, for which
       $Z_1 \preceq Z_2$, and reference vector $\boldsymbol{r}^2$
Output: The set of attained areas $A$

$\boldsymbol{r}^1 \leftarrow$ component-min($Z_1$);
$O \leftarrow$ opposite($Z_1, \boldsymbol{r}^1, \boldsymbol{r}^2$);
foreach $\boldsymbol{z} \in Z_1$ do
    $O' \leftarrow \{\boldsymbol{o} \in O; \boldsymbol{z} \prec\prec \boldsymbol{o}\}$;
    foreach $\boldsymbol{o} \in O'$ do
        $A \leftarrow A \cup$ new-area($\boldsymbol{z}, \boldsymbol{o}$, value($\boldsymbol{z}$));
    end
end
```

example we delete vector $\boldsymbol{o}^4$, but not $\boldsymbol{o}^5$). Next, for every deleted vector $\boldsymbol{o}$ we create three new vectors in the following way:

$$\boldsymbol{o}^{n_1} = (z_1, o_2, o_3)$$
$$\boldsymbol{o}^{n_2} = (o_1, z_2, o_3)$$
$$\boldsymbol{o}^{n_3} = (o_1, o_2, z_3)$$

Finally, each of these vectors is added to the opposite if it is not coplanar with $\boldsymbol{r}^1$ and not collinear with any of the vectors from the current opposite. In our example, we add to the opposite vectors $\boldsymbol{o}^6$ and $\boldsymbol{o}^7$, but not $\boldsymbol{o}^8 = (2, 6.5, 5)$ (not pictured in the figure) as it is collinear with $\boldsymbol{o}^5$. When these steps have been taken for every vector from Z, the resulting set O represents the opposite of Z. See also Algorithm 2 for the algorithmic notation of the described procedure.

Note that the same idea of cutting into the objective space could also be used for the 2D case. However, Algorithm 1 is so simple that there is no need to look for an alternative.

3.2 From Opposites to Areas

Let us return to the example of two consecutive attainment surfaces Z_1 and Z_2. If the opposite O of Z_2 is known, it is rather straightforward to compute the attained areas between Z_1 and Z_2 and can be done in a uniform way for both the 2D and 3D case (see Algorithm 3). For every attainment anchor $\boldsymbol{z} \in Z_1$ and every vector $\boldsymbol{o} \in O$ that is strictly dominated by $\boldsymbol{z}$, an area enclosed by these two vectors is added to the set of attained areas.

In addition, each area can be associated with a value—either the EAF value of the difference between EAF values of the attainment anchor $\boldsymbol{z}$. While some areas might overlap, this is not a problem since all overlapping areas have the same value (if they did not have the same value, another anchor vector would have already split the area).

The Algorithm 3 computes the attained areas between two consecutive attainment surfaces. In order to compute the areas for the whole objective space, we need to iterate the algorithm over all consecutive pairs of attainment surfaces.

4. VISUALIZING EAF DIFFERENCES USING MAXIMUM INTENSITY PROJECTION

The procedure for visualizing EAF differences between two sets of approximation sets consists of the following steps:

1. Combine all approximation sets and calculate the EAF values and the differences in those values.
2. Compute the cuboids between all pairs of attainment surfaces.
3. Plot separately the cuboids with positive and negative EAF differences using MIP.

Let us assume we wish to plot the cuboids using a standard plotting tool able to visualize 3D data. Usually, the 3D rendering tools do not allow for cuboids with larger EAF differences to cover those with smaller EAF differences, as is the principle of MIP, but visualize the cuboids in a sequence that maintains some notion of depth (cuboids near the viewpoint are shown in front of the cuboids further away). Therefore, MIP can be performed only after the viewpoint has been set and the cuboids are plotted onto 2D. Now we can choose to

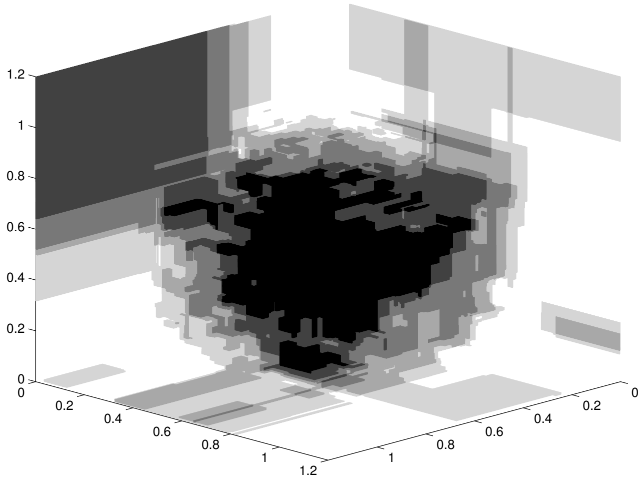

Figure 8: Areas in the objective space where the linear approximation sets outperform the spherical ones as shown by MIP (darker areas denote larger differences).

visualize them in a predefined order from the smallest to the largest EAF difference. If plotted in this way, cuboids with larger EAF differences cover those with smaller differences, effectively achieving the MIP visualization of the cloud of cuboids.

Now, let us demonstrate the visualization of EAF differences by means of MIP on an example. The attainment surfaces used in this example are not the product of optimization algorithms, but rather benchmark approximation sets with known properties used for visualization purposes [7].

The first set consists of five linear approximation sets with a uniform distribution of vectors that satisfy the constraint

$$\sum_{i=1}^{3} z_i = c_\mathrm{L}.$$

In order to simulate the behavior of a stochastic algorithm, the values of c_L for different sets follow the normal distribution with mean 1 and variance 0.05. The second set contains five spherical approximation sets with a non-uniform distribution of vectors where only few vectors are located in the middle of the approximation set, while most of them are near its corners. The vectors from the spherical approximation sets satisfy the constraint

$$\sum_{i=1}^{3} z_i^2 = c_\mathrm{S}^2,$$

where the values of c_S for each of the five sets follow the normal distribution with mean 0.75 and variance 0.05. Each individual approximation set contains 100 objective vectors.

The values of c_L and c_S were chosen so that the sets are always intertwined—in one region, the linear approximation sets dominate the spherical ones, while in others, the spherical sets dominate the linear ones. This means that there will be visible differences in EAF values for the two sets of approximation sets.

Figure 8 shows the areas of the objective space where the linear approximation sets outperform the spherical ones.

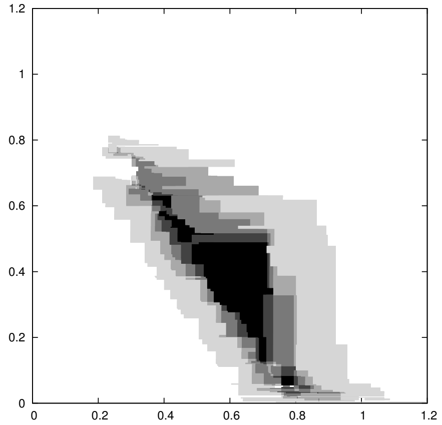

Figure 9: The intersection between the cuboids showing outperformance of the linear approximation sets with the plane crossing the origin and cutting the xy plane at the angle of $45°$.

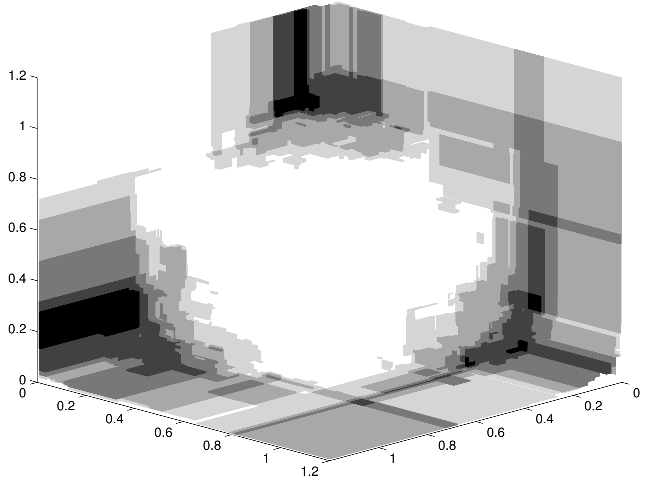

Figure 10: Areas in the objective space where the spherical approximation sets outperform the linear ones as shown by MIP (darker areas denote larger differences).

Here the cloud of cuboids is nicely visualized. We can see that the largest values are in its center, surrounded by decreasing values. Nevertheless, this MIP visualization shares the same disadvantages as MIP visualizations in medicine—because we can see through the cloud of cuboids, we loose the sense of depth. While it is inevitable to lose some information when projecting 3D structures onto 2D, this can be amended by combining two visualization techniques—MIP and slicing through the cloud, which visualizes the intersection between the cuboids and the cutting plane (see [6] for details and the example of such a cut in Figure 9). Now we can see that the cloud is not round at its side closest to the origin, but flat as the best approximation sets in that area of the objective space are the linear ones.

Similarly, we can use MIP to visualize the areas of the objective space where the spherical approximation sets outperform the linear ones (see Figure 10). As it could be expected,

Table 1: **The number of attainment anchors and cuboids depending on the number of approximation sets.**

No. of sets	No. of all vectors	No. of attainment anchors	No. of cuboids with $\delta^{\text{L-S}} \neq 0$	No. of cuboids with $\delta^{\text{L-S}} > 0$	No. of cuboids with $\delta^{\text{L-S}} < 0$
10	1 000	8 093	22 577	14 470	8 107
20	2 000	49 044	91 088	58 392	32 696

we can see that the spherical sets dominate the linear ones only near the corners of the objective space.

Finally, a note on the complexity of such visualizations. In our example we have used two sets of five approximation sets, each containing 100 objective vectors. In addition, we have performed the same experiments on two sets with ten approximation sets, again containing 100 objective vectors each. Table 1 collects the number of objective vectors, attainment anchors and cuboids in each experiment. As we can see, the number of attainment anchors significantly increases with the growing number of approximation sets (as already mentioned in Subsection 2.2). What these numbers mean in practice is that it is very difficult to visualize all areas with non-zero EAF differences when dealing with ten or more runs per algorithm since this entails visualizing approximately 50 000 cuboids. As we are usually interested mainly in the areas where the differences between algorithms are large (or even statistically significant), one possibility is to remove the cuboids with smaller EAF differences before visualization with MIP, thus decreasing their number.

The Algorithm 3 constructing the cuboids allows them to overlap. As we can see from Table 1, this happens very often (there are between two and three cuboids per attainment anchor). It might be interesting to explore if posing a restriction on overlapping would bring better results in terms of faster rendering of the cuboids. While non-overlapping cuboids would be smaller (and therefore easier to visualize), they might increase in number, which makes it hard to predict if such a procedure would be advantageous.

5. CONCLUSION

We have tackled the visualization of 3D EAF differences between two optimization algorithms, which consists of two subproblems. In order to show the EAF differences, the objective space needs to be first divided into cuboids with constant differences. Second, to compute these cuboids, the opposites of all attainment anchors must be found. We have presented algorithms to solve these subproblems and shown some preliminary results of the visualization with MIP on two sets of benchmark approximation sets.

In the future, we intend to provide more combined MIP and slicing visualization examples, including those from real-world optimization problems. Also, the algorithms presented in this paper could probably be further improved, for example, by storing the already computed relations between objective vectors. Additionally, it might be interesting to explore a completely different approach where the objective space would not contain exact cuboids, but would be discretized into voxels with values approximating the EAF differences in the covered area. While any discretization would surely yield only an approximation of the true cloud of cuboids, it could provide better efficiency. Moreover, existing MIP rendering software could then be used to visualize the voxels (including interactive rotations).

6. ACKNOWLEDGMENTS

The work presented in this paper was carried out as part of research program P2-0209 and research projects L2-3651 and J2-4120, all funded by the Slovenian Research Agency.

The first author wishes to acknowledge Richard Everson and Manuel López-Ibáñez for discussing some ideas presented in this paper at GECCO'13.

7. REFERENCES

[1] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001.

[2] C. M. Fonseca, A. P. Guerreiro, M. López-Ibáñez, and L. Paquete. On the computation of the empirical attainment function. In *Proceedings of the 6th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2011*, volume 6576 of *Lecture Notes in Computer Science*, pages 106–120, 2011.

[3] V. D. Grunert da Fonseca, C. M. Fonseca, and A. O. Hall. Inferential performance assessment of stochastic optimisers and the attainment function. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization, EMO 2001*, volume 1993 of *Lecture Notes in Computer Science*, pages 213–225, 2001.

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

[5] S. Schreiner and R. L. Galloway. A fast maximum-intensity projection algorithm for generating magnetic resonance angiograms. *IEEE Transactions on Medical Imaging*, 12(1):50–57, March 1993.

[6] T. Tušar and B. Filipič. An approach to visualizing the 3D empirical attainment function. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1367–1372, 2013.

[7] T. Tušar and B. Filipič. Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*, 2014. In press.

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