# VISUALIZING SOLUTION SETS IN MULTIOBJECTIVE OPTIMIZATION

Tea Tušar

**Doctoral Dissertation**
**Jožef Stefan International Postgraduate School**
**Ljubljana, Slovenia**

**Supervisor:** Prof. Dr. Bogdan Filipič, Jožef Stefan Institute, Ljubljana, Slovenia

**Evaluation Board:**
Prof. Dr. Marko Bohanec, Chair, Jožef Stefan Institute, Ljubljana, Slovenia
Assist. Prof. Dr. Barbara Koroušić Seljak, Member, Jožef Stefan Institute, Ljubljana, Slovenia
Prof. Dr. Thomas Bartz-Beielstein, Member, Cologne University of Applied Sciences, Germany

**MEDNARODNA PODIPLOMSKA ŠOLA JOŽEFA STEFANA**
JOŽEF STEFAN INTERNATIONAL POSTGRADUATE SCHOOL

Tea Tušar

# VISUALIZING SOLUTION SETS IN MULTIOBJECTIVE OPTIMIZATION

**Doctoral Dissertation**

# VIZUALIZACIJA MNOŽIC REŠITEV V VEČKRITERIJSKI OPTIMIZACIJI

**Doktorska disertacija**

**Supervisor:** Prof. Dr. Bogdan Filipič

Ljubljana, Slovenia, June 2014

*To Dejan, Erika and Primož*

# Acknowledgments

First and foremost, sincere thanks to my supervisor, Prof. Bogdan Filipič. He taught me everything I know about being a researcher and his advice and support were invaluable to this work. He helped me stay in focus and encouraged me whenever I lost momentum. I consider working with him a privilege and I am looking forward to our future collaboration.

Special thanks to Prof. Marko Bohanec and Assist. Prof. Barbara Koroušić Seljak, who have followed my doctoral research from the start and contributed many insightful ideas that improved it. I am also very grateful to Prof. Thomas Bartz-Beielstein for his thorough review of the thesis.

Thanks to all participants of the AVN workshops, where most of this work was first presented. They always shared useful ideas and I was often inspired by our discussions. Many thanks also to Boris Naujoks for providing comments on an early draft of visualization of approximation sets and to Manuel López-Ibáñez and Prof. Richard Everson for discussions about visualization of empirical attainment functions. I am very grateful to Prof. Hisao Ishibuchi and Yuki Tanigaki for sending me their results of the polygon-based test problems. Further, I wish to acknowledge the support of Robert Vertnik and Miha Mlakar on the steel-casting use case.

I want to thank my colleagues from the Department of Intelligent Systems of the Jožef Stefan Institute (and especially those from the Computational Intelligence Group) for providing a friendly working environment. Thanks also to the department head, Prof. Matjaž Gams, for giving me the possibility to concentrate on the thesis in the last few weeks. Special thanks to my friend and colleague Bernard Ženko who always provided what I needed (be it chocolate or remote printing).

I thankfully acknowledge financial support by the Jožef Stefan Institute and the Ministry of Education, Science and Sport of the Republic of Slovenia covering tuition fees and conference attendance.

Finally, many thanks to my friends and family. They were always there for me and kept their “How is the research going?” questions to a minimum. I am especially grateful to Erika and Primož for being patient with their busy mum and to Dejan for taking on most of the parenting and household duties (it is only temporary, I promise). He is the one person I could not have done this without.

# Abstract

Many real-world optimization problems are inherently multiobjective, for example, searching for trade-off solutions of high quality and low cost. As no single optimal solution exists for such problems, multiobjective optimization algorithms provide a set of optimal (or near-optimal) trade-off solutions to choose from, called an approximation set. Because such algorithms are often stochastic, they return a different approximation set in every run. The Empirical Attainment Function (EAF) is able to describe the probabilistic distribution of multiple approximation sets and can be therefore used to analyze and compare the performance of such algorithms. As multiobjective optimization deals with vectors in the multidimensional objective space, it is very important to be able to visualize them. This thesis addresses two distinct tasks in visualization in multiobjective optimization—visualization of approximation sets and visualization of EAFs.

While scatter plots can be used for visualizing 2D and 3D approximation sets, more advanced approaches are needed to handle four or more objectives. This thesis presents a comprehensive review of the existing visualization methods used in evolutionary multiobjective optimization, showing their outcomes on two novel 4D benchmark approximation sets. In addition, a visualization method that uses prosection (projection of a section) to visualize 4D approximation sets is proposed. The method adequately reproduces the shape, range and distribution of vectors in the observed approximation sets and can handle multiple large approximation sets while being robust and computationally inexpensive. Even more importantly, for numerous vectors, the visualization with prosections preserves the Pareto dominance relation and relative closeness to reference vectors. Visualization with prosections is analyzed theoretically and demonstrated on several approximation sets.

While the visualization of EAFs is rather straightforward in two objectives, the three-objective case presents a great challenge as we need to visualize a large number of 3D cuboids. This thesis addresses the visualization of exact as well as approximated 3D EAF values and differences in these values provided by two competing multiobjective optimization algorithms. First, we compute the 3D cuboids. Then, we show that the exact EAFs can be visualized using Slicing and Maximum Intensity Projection, while the approximated EAFs can be visualized using Slicing, Maximum Intensity Projection and Direct Volume Rendering. In addition, the thesis demonstrates the use of the proposed visualization techniques on a real-world steel-casting optimization problem.

# Povzetek

Številni realni optimizacijski problemi so po naravi večkriterijski, saj zahtevajo iskanje kompromisov, na primer med kakovostjo rešitev in njihovo ceno. Ker tovrstni problemi nimajo ene same optimalne rešitve, večkriterijski optimizacijski algoritmi vrnejo množico (skoraj) optimalnih kompromisnih rešitev, ki se imenuje aproksimacijska množica. Takšni algoritmi so pogosto stohastični, zato v vsakem zagonu vrnejo drugačno aproksimacijsko množico. Empirična funkcija dosega (angl. Empirical Attainment Function, EAF) zna opisati verjetnostno porazdelitev več aproksimacijskih množic in se zato lahko uporablja za analizo in primerjavo takšnih algoritmov. Ker imamo pri večkriterijski optimizaciji opravka z vektorji v večdimenzionalnem prostoru kriterijev, je zelo pomembno, da jih znamo vizualizirati. Ta disertacija obravnava dve različni nalogi vizualizacije v večkriterijski optimizaciji – vizualizacijo aproksimacijskih množic in vizualizacijo EAF.

Medtem ko za vizualizacijo 2D in 3D aproksimacijskih množic lahko uporabljamo razsevne diagrame, je treba pri štirih ali več kriterijih poseči po naprednejših vizualizacijskih metodah. Disertacija predstavlja obsežno primerjalno študijo sorodnih metod za vizualizacijo aproksimacijskih množic tako, da prikaže njihove rezultate na dveh novih 4D testnih aproksimacijskih množicah. Poleg tega predlaga novo vizualizacijsko metodo, ki s prosekcijo (projekcijo sekcije) vizualizira 4D aproksimacijske množice v 3D. Metoda v veliki meri ohranja obliko, obseg in porazdelitev vektorjev v aproksimacijskih množicah ter je hkrati robustna in računsko nezahtevna. Še pomembneje, sposobna je ohranjati relacijo Pareto dominiranosti ter relativno bližino do referenčnih točk za številne vektorje. Lastnosti vizualizacije s prosekcijami smo bodisi formalno dokazali bodisi prikazali na praktičnih primerih.

Čeprav je vizualizacija EAF dokaj enostavna v primeru dveh kriterijev, je naloga s tremi kriteriji mnogo zahtevnejša, saj moramo vizualizirati veliko število kvadrov. Disertacija obravnava vizualizacijo tako eksaktnih kot aproksimiranih vrednosti EAF ter tudi razlik v vrednostih EAF med dvema optimizacijskima algoritmoma. Najprej kvadre izračunamo, nato pa pokažemo, da se eksaktne EAF da vizualizirati s prerezi ali metodo MIP (angl. Maximum Intensity Projection), aproksimirane EAF pa s prerezi ter z metodama MIP in DVR (angl. Direct Volume Rendering). Zaključimo s prikazom uporabe predlaganih vizualizacijskih metod na realnem optimizacijskem problemu ulivanja jekla.

# Contents

# List of Figures

# List of Tables

# List of Algorithms

# Abbreviations

| | | |
|---|---|---|
| BAS | ... | benchmark approximation set |
| DEB$m$DK | ... | a suite of benchmark optimization problems with knees in $m$ objectives by Deb |
| DEMO | ... | differential evolution for multiobjective optimization |
| DTLZ | ... | a suite of benchmark optimization problems by Deb, Thiele, Laumanns and Zitzler |
| DVR | ... | direct volume rendering |
| EAF | ... | empirical attainment function |
| EPH | ... | Edgeworth-Pareto hull |
| ES | ... | exhaustive search |
| IDM | ... | interactive decision map |
| MDS | ... | multi-dimensional scaling |
| MIP | ... | maximum intensity projection |
| MOEA | ... | multiobjective evolutionary algorithm |
| MOEA/D | ... | multiobjective evolutionary algorithm based on decomposition |
| NSGA-II | ... | nondominated sorting genetic algorithm II |
| PCA | ... | principal component analysis |
| RadViz | ... | radial coordinate visualization |
| RGBA | ... | red green blue alpha |
| SOM | ... | self-organizing map |
| SMS-EMOA | ... | $S$ metric selection evolutionary multiobjective optimization algorithm |
| SPEA2 | ... | improved strength Pareto evolutionary algorithm |
| WFG | ... | a suite of benchmark optimization problems by the Walking Fish Group |

# Symbols

| | |
|---|---|
| $\boldsymbol{f}$ | ... optimization function |
| $\boldsymbol{x}$ | ... decision vector |
| $X$ | ... decision space |
| $n$ | ... dimensionality of the decision space |
| $\boldsymbol{z}$ | ... objective vector |
| $F$ | ... objective space |
| $m$ | ... dimensionality of the objective space |
| $\prec$ | ... Pareto dominance relation |
| $\preceq$ | ... weak Pareto dominance relation |
| $\prec\!\!\prec$ | ... strict Pareto dominance relation |
| $\parallel$ | ... incomparability relation |
| $Z$ | ... approximation set |
| $\boldsymbol{m}$ | ... minimal element |
| $Z^{\min}$ | ... minimal set |
| $Z^{\max}$ | ... maximal set |
| $P_{\boldsymbol{f}}$ | ... Pareto front of optimization function $\boldsymbol{f}$ |
| $Z_{\mathrm{EPH}}$ | ... Edgeworth-Pareto hull of approximation set $Z$ |
| $\partial Z$ | ... boundary of approximation set $Z$ |
| $S_Z$ | ... attainment surface of approximation set $Z$ |
| $A_Z$ | ... attainment anchors of approximation set $Z$ |
| $\alpha_r^{\mathrm{A}}$ | ... empirical attainment function of algorithm $A$ run $r$ times |
| $I$ | ... indicator function |
| $L_{t/r}^{+}(\alpha_r^{\mathrm{A}})$ | ... the $t/r$-superlevel set of the empirical attainment function $\alpha_r^{\mathrm{A}}$ |
| $S_{t/r}$ | ... summary (or $t/r$)-attainment surface |
| $A_{t/r}$ | ... attainment anchors of the $t/r$-attainment surface |
| $\delta_r^{\mathrm{A-B}}$ | ... empirical attainment function difference between algorithms $A$ and $B$ (each run $r$ times) |
| $\boldsymbol{r}, \boldsymbol{r}^1, \boldsymbol{r}^2$ | ... reference vectors |
| $R$ | ... observed objective space |
| $O_Z^R$ | ... opposite of the approximation set $Z$ within the observed objective space $R$ |
| $I_{\mathrm{H}}$ | ... hypervolume indicator |
| $\boldsymbol{a}$ | ... prosection origin |
| $f_i f_j$ | ... prosection plane |
| $\varphi$ | ... prosection angle |
| $d$ | ... section width |
| $m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ | ... prosection of an $m$D approximation set with origin $\boldsymbol{a}$, plane $f_i f_j$, angle $\varphi$ and section width $d$ |
| $s_{\varphi,d,\boldsymbol{a}}$ | ... dimension reduction function in prosection |
| $N(\mu, \sigma)$ | ... normal distribution with mean $\mu$ and standard deviation $\sigma$ |

# Chapter 1

# Introduction

## 1.1 Problem Formulation

Many real-world optimization problems involve the optimization of several, often conflicting objectives. Consequently, instead of a single optimal solution, a set of optimal solutions exists for such problems. In the objective space, this set is called the *Pareto front*. While some popular traditional approaches to solving multiobjective optimization problems used to combine all objectives into a single one and solve the resulting singleobjective optimization problem, in the last two decades a lot of research has been directed into developing algorithms able to tackle these problems in their true multiobjective form. Most notably, Multiobjective Evolutionary Algorithms (MOEAs) have proven to be capable of finding high-quality solutions to such problems (Deb, 2001, 2012). A MOEA provides a set of trade-off solutions approximating the Pareto front where no solution from the set is better than any other in all objectives. This is called an *approximation set* and MOEAs typically return a different approximation set in every run. The probabilistic distribution of multiple approximation sets by a MOEA can be described by the Empirical Attainment Function (EAF) (Grunert da Fonseca, Fonseca, & Hall, 2001).

There exist many measures to assess the quality of approximation sets, i.e. how well they approximate the Pareto front in terms of convergence, spread and distribution of objective vectors (Deb & S. Jain, 2002; Zitzler, Thiele, Laumanns, Fonseca, & Grunert da Fonseca, 2003). However, no measure is as effective as visualization, especially if the Pareto front is known and can be visualized as well. Visualization in multiobjective optimization is essential in many aspects—it can be used to (Knowles, Corne, & Deb, 2008): estimate the location, range and shape of the Pareto front, assess conflicts and trade-offs between objectives, select preferred solutions, monitor the progress or convergence of an optimization run, assess the relative performance of different MOEAs, etc. However, note that the task of visualizing approximation sets is very different from the task of visualizing EAFs. This thesis addresses both.

### 1.1.1 Visualization of Approximation Sets

Based on the applications of visualization we can define some requirements for visualization methods. A method for visualizing approximation sets should be able to *preserve the Pareto dominance relation* between objective vectors. This means that any relation between objective vectors $\boldsymbol{z}^{\mathrm{A}}$ and $\boldsymbol{z}^{\mathrm{B}}$ ($\boldsymbol{z}^{\mathrm{A}}$ dominates $\boldsymbol{z}^{\mathrm{B}}$, $\boldsymbol{z}^{\mathrm{B}}$ dominates $\boldsymbol{z}^{\mathrm{A}}$, or $\boldsymbol{z}^{\mathrm{A}}$ and $\boldsymbol{z}^{\mathrm{B}}$ are incomparable) should be evident also from their visualization. This is crucial when comparing two or more approximation sets, since without dominance preservation a visualized approximation set may seem to dominate another one while this is not the case. Moreover, visualized approximation sets should *maintain their shape, range and distribution of vectors* as any large distortion of these features affects our perception of the approximation sets (Metzger, 2006). The shape of the approximation set might be of great importance to the decision maker as it presents the trade-offs among the objectives. Further, a visualization method should be *robust*, meaning that the addition or removal of a vector within the range of the approximation set should not produce a significantly different visualization. As approximation sets found by MOEAs are often large, a visualization method should be able to *handle large sets* in terms of visualization capability as well as computational complexity. Additionally, *simultaneous visualization of multiple approximation sets* is required if different approximation sets are to be compared. Finally, a visualization method should be *scalable* to multiple dimensions and *simple to understand and use*. These requirements imply that the task of visualizing an approximation set is fundamentally different from the task of visualizing general multidimensional sets.

While visualization of 2D and 3D approximation sets satisfying all these requirements is rather trivial (simple scatter plots can be used for this purpose), this is not the case for four- and more-dimensional approximation sets solving the so-called *many-objective* optimization problems. Because "standard" MOEAs designed to efficiently solve problems with two and three objectives perform poorly on many-objective optimization problems (Purshouse & Fleming, 2003; Khare, Yao, & Deb, 2003), more sophisticated MOEAs trying to address this issue are being proposed, for example (Zhang & Li, 2007; Deb & H. Jain, 2013). Visualization in many-objective optimization is especially important as it could help better understand the limitations of the algorithms and provide information to fuel further advances in this field.

Numerous methods for visualizing general multidimensional sets exist (dos Santos, 2004; Miettinen, 2014). However, they disregard the specific characteristic of approximation sets and do not try to preserve the Pareto dominance relation[1]. Even the visualization methods designed specifically for approximation sets fail to correctly visualize the relations among vectors. Only Pareto shells (Walker, Everson, & Fieldsend, 2010) are able to do so, using arrows to connect dominated vectors to those dominating them, but at the cost of other information on the approximation sets, such as their shape, range and distribution of vectors.

Each of the existing visualization methods serves some purpose and its effectiveness (or success) cannot be formally measured and compared to other methods. As a matter of fact, there is no established evaluation methodology that could be applied to methods visualizing approximation sets.

### 1.1.2 Visualization of Empirical Attainment Functions

In order to visually analyze the behavior of a MOEA solving a two- or three-objective optimization problem, a set of its approximation sets needs to be visualized. Because approximation sets of a MOEA are often very similar, visualization using scatter plots can prove to be difficult already for a small number of approximation sets. This difficulty can be overcome with the use of the EAF, which assigns to each vector in the objective space a value signifying how often the vector was attained by the given approximation sets (Grunert da Fonseca et al., 2001). 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

[1]Note that there exists no general Pareto-dominance preserving mapping from a higher-dimensional space to a lower-dimensional space (Köppen & Yoshida, 2007), meaning that it is not possible to preserve the Pareto dominance relation between *all* vectors of a 4D approximation set. Therefore, we can only strive to preserve the Pareto dominance relation between as many vectors as possible.

by showing differences in these values (López-Ibáñez, Paquete, & Stützle, 2010). 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 (and was to the best of our knowledge never tried before). 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. Generally, we are interested in the location of these clouds as well as their values. If exactness is not crucial, the 3D objective space can be approximated using a grid of *voxels*, i.e. values in a 3D grid, and it is enough to visualize approximated EAFs instead of exact ones.

It is straightforward to see that visualization of EAF values and differences (either exact or approximated) represents a completely different problem to the visualization of approximation sets as we need to visualize volumetric data, not just vectors in a 3D space.

## 1.2 Goals and Hypotheses

The purpose of this dissertation is to address the stated tasks of visualization in multiobjective optimization. The specific goals of the dissertation are to:

– create benchmark approximation sets for comparing visualization methods,

– present a comprehensive survey of the state-of-the-art in visualization of approximation sets using these benchmark sets,

– design and implement a new method for visualizing approximation sets, identify its properties and state their formal proofs, as well as demonstrate its usage on a variety of approximation sets,

– design and implement a method for computing exact attained areas in 3D (cuboids),

– visualize the exact and approximated EAF values and differences using different methods, discuss their properties and demonstrate their validity on a real-world multiobjective optimization problem.

The dissertation investigates the following two hypotheses:

- It is possible to visualize 4D approximation sets in 3D so that some of the idealistic requirements for visualization methods (such as the preservation of the Pareto dominance relation, maintenance of the shape, range and distribution of vectors in the approximation set, robustness, ability to handle multiple large approximation sets, scalability to multiple objectives and simplicity) are met as closely as possible.

- The exact and approximated 3D EAF values and differences can be computed and meaningfully visualized.

## 1.3 Original Contributions

The thesis leads to the following original scientific contributions:

- The design of benchmark approximation sets, not yet existing in the literature, which can provide a basis for establishing an evaluation and comparison methodology for visualization methods. Also, the use of these sets in a critical review of existing methods for visualizing approximation sets.

- A novel visualization method capable of simultaneously visualizing several 4D approximation sets while preserving the Pareto dominance relation between numerous vectors. Such a visualization method enables analysis and straightforward comparison of different 4D MOEAs, which has not been handled satisfactorily so far. In addition, some of the properties of this visualization method (partial preservation of the dominance relation and relative closeness to reference vectors) can be formally proven.

- The first attempt at visualizing 3D EAF values and differences, which covers the exact as well as the approximated case.

## 1.4 Organization of the Thesis

Chapter 2 provides the background required for proper understanding of terms used in multiobjective optimization with additional focus on the EAF. As the thesis addresses two distinct visualization tasks, it is divided into two parts—Part I is dedicated to the visualization of approximation sets, while Part II comprises chapters devoted to the visualization of EAFs.

Chapter 3 in Part I introduces the benchmark approximation sets, which are used in Chapter 4 to review the related methods for visualizing approximation sets. Part I concludes with the presentation of visualization with prosections, the newly proposed visualization method, in Chapter 5.

Part II starts by presenting related work on visualization of EAFs in Chapter 6. Next, Chapter 7 is dedicated to the visualization of exact and approximated 3D EAFs, while Chapter 8 shows how the proposed visualization methods can be used on a real-world optimization problem.

Finally, concluding remarks are presented in Chapter 9. In addition, Appendix A contains proofs of theorems from Chapter 5.

# Chapter 2

# Background

This chapter defines the terms used throughout the thesis. Section 2.1 lists the most important concepts from multiobjective optimization. In addition to illustrating the meaning of the empirical attainment function, Section 2.2 introduces formal definitions of some new terms related to the empirical attainment function, such as *attainment anchors* and *opposites*.

## 2.1 Multiobjective Optimization Concepts

**Definition 2.1 (Multiobjective optimization problem).** The *multiobjective optimization problem* consists of finding the optimum of a function:

$$\begin{aligned} \boldsymbol{f} &: X \to F \\ \boldsymbol{f} &: (x_1, \ldots, x_n) \mapsto (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)), \end{aligned} \tag{2.1}$$

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*.

We assume that $F \subseteq \mathbb{R}^m$ and all objectives $f_i : X \to \mathbb{R}$ are to be minimized. Note that we follow the established terminology in multiobjective optimization and use the term *vector* to denote a *position vector* with respect to the origin $\mathbf{0}$, which means it is equivalent to a *point* in the multidimensional Euclidean space. Therefore, mappings and projections of vectors as well as distances between vectors actually refer to mappings, projections and distances between the points that correspond to these vectors.

As we deal 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 2.2 (Pareto dominance of vectors).** The objective vector $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}}) \in F$ *dominates* the objective vector $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}}) \in F$, i.e. $\boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}}$, if

$$z_i^{\mathrm{A}} \leq z_i^{\mathrm{B}} \text{ for all } i \in \{1, \ldots, m\} \text{ and there exists a } j \in \{1, \ldots, m\} \text{ for which } z_j^{\mathrm{A}} < z_j^{\mathrm{B}}. \tag{2.2}$$

**Definition 2.3 (Weak Pareto dominance of vectors).** The objective vector $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}}) \in F$ *weakly dominates* the objective vector $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}}) \in F$, i.e. $\boldsymbol{z}^{\mathrm{A}} \preceq \boldsymbol{z}^{\mathrm{B}}$, if

$$z_i^{\mathrm{A}} \leq z_i^{\mathrm{B}} \text{ for all } i \in \{1, \ldots, m\}. \tag{2.3}$$

**Definition 2.4 (Strict Pareto dominance of vectors).** The objective vector $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}}) \in F$ *strictly dominates* the objective vector $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}}) \in F$, i.e. $\boldsymbol{z}^{\mathrm{A}} \prec\!\!\prec \boldsymbol{z}^{\mathrm{B}}$, if

$$z_i^{\mathrm{A}} < z_i^{\mathrm{B}} \text{ for all } i \in \{1, \ldots, m\}. \tag{2.4}$$

**Definition 2.5 (Incomparability of vectors).** The objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}}) \in F$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}}) \in F$ are *incomparable*, i.e. $\boldsymbol{z}^{\mathrm{A}} \parallel \boldsymbol{z}^{\mathrm{B}}$, if

$$\boldsymbol{z}^{\mathrm{A}} \not\preceq \boldsymbol{z}^{\mathrm{B}} \text{ and } \boldsymbol{z}^{\mathrm{B}} \not\preceq \boldsymbol{z}^{\mathrm{A}}. \tag{2.5}$$

**Definition 2.6 (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 distinct objective vectors $\boldsymbol{z}^{\mathrm{A}}, \boldsymbol{z}^{\mathrm{B}} \in Z$.

**Definition 2.7 (Weak Pareto dominance of approximation sets).** The approximation set $Z^{\mathrm{A}} \subseteq F$ *weakly dominates* the approximation set $Z^{\mathrm{B}} \subseteq F$, 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 $\boldsymbol{z}^{\mathrm{A}} \in Z^{\mathrm{A}}$.

**Definition 2.8 (Ideal objective vector).** The *ideal objective vector* $\boldsymbol{z}^* = (z_1^*, \ldots, z_m^*)$ represents the minimum possible value in each objective (typically, it cannot be achieved):

$$z_i^* = \min_{\boldsymbol{x} \in X} f_i(\boldsymbol{x}) \text{ for all } i \in \{1, \ldots, m\}. \tag{2.6}$$

**Definition 2.9 (Minimal element, minimal set).** Let $Z \subseteq F$ be a set of objective vectors. An objective vector $\boldsymbol{m} \in Z$ is a *minimal element* of $Z$ if

$$\text{for all } \boldsymbol{z} \in Z, \boldsymbol{z} \preceq \boldsymbol{m} \Rightarrow \boldsymbol{z} = \boldsymbol{m}. \tag{2.7}$$

All minimal elements of $Z$ constitute the *minimal set* of $Z$ denoted in this thesis by $Z^{\min}$.

*Maximal elements* and the *maximal set* $Z^{\max}$ can be defined dually.

**Definition 2.10 (Pareto front).** The set of Pareto optimal objective vectors known as the *Pareto front* can be formally defined as the minimal set of $\boldsymbol{f}(X)$:

$$P_{\boldsymbol{f}} = \boldsymbol{f}(X)^{\min}. \tag{2.8}$$

**Definition 2.11 (Edgeworth-Pareto hull).** Let $Z \subseteq F$ be an approximation set. The *Edgeworth-Pareto hull* (EPH) of $Z$ includes all objective vectors weakly dominated by $Z$:

$$Z_{\mathrm{EPH}} = \{\boldsymbol{z} \in F;\ \text{exists } \boldsymbol{z'} \in Z \text{ so that } \boldsymbol{z'} \preceq \boldsymbol{z}\}. \tag{2.9}$$

**Definition 2.12 (Pareto-dominance preserving mapping).** The mapping $\Pi : \mathbb{R}^m \to \mathbb{R}^n$, where $n < m$, is a *Pareto-dominance preserving mapping* if

$$\boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}} \iff \Pi(\boldsymbol{z}^{\mathrm{A}}) \prec \Pi(\boldsymbol{z}^{\mathrm{B}}) \tag{2.10}$$

for any two vectors $\boldsymbol{z}^{\mathrm{A}}, \boldsymbol{z}^{\mathrm{B}} \in \mathbb{R}^m$.

Note that Köppen and Yoshida (2007) have shown that such a mapping does not exist.

To illustrate the presented concepts, let us look at the example of two finite sets of vectors $X = \{\boldsymbol{x}^1, \ldots, \boldsymbol{x}^5\}$ and $F = \{\boldsymbol{z}^1, \ldots, \boldsymbol{z}^5\}$ and a function $\boldsymbol{f} : X \to F$, which maps each decision vector to the corresponding objective vector using $\boldsymbol{f}(\boldsymbol{x}^i) = \boldsymbol{z}^i$ for all $i \in \{1, \ldots, 5\}$. There are two objectives involved, both subject to minimization. Based on the values of the objective vectors shown in Figure 2.1, we can distinguish the following Pareto dominance relations among the objective vectors: $\boldsymbol{z}^1 \prec \boldsymbol{z}^2, \boldsymbol{z}^3 \prec \boldsymbol{z}^2, \boldsymbol{z}^4 \prec \boldsymbol{z}^2$ and $\boldsymbol{z}^4 \prec \boldsymbol{z}^5$. All these ordered pairs of vectors are also in the weak Pareto dominance

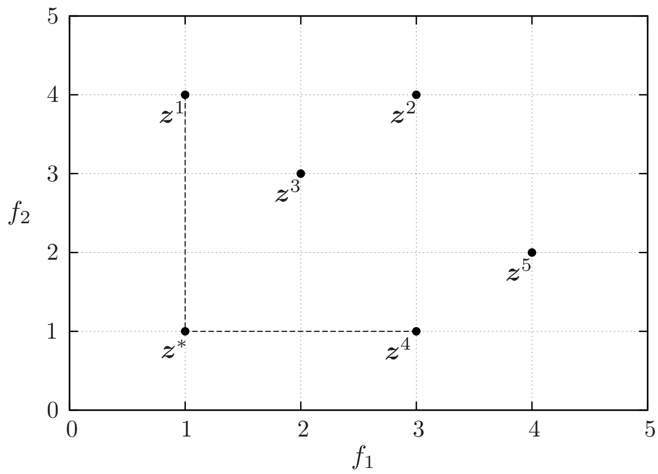



Figure 2.1: Example of vectors in the objective space illustrating some multiobjective optimization concepts.

relation. In addition, every objective vector weakly dominates itself, i.e. $\boldsymbol{z}^i \preceq \boldsymbol{z}^i$ for all $i \in \{1, \ldots, 5\}$. While there are only two strict Pareto dominance relations in this example, $\boldsymbol{z}^3 \prec\!\!\prec \boldsymbol{z}^2$ and $\boldsymbol{z}^4 \prec\!\!\prec \boldsymbol{z}^5$, many pairs of vectors are incomparable, $\boldsymbol{z}^1 \parallel \boldsymbol{z}^3, \boldsymbol{z}^1 \parallel \boldsymbol{z}^4, \boldsymbol{z}^1 \parallel \boldsymbol{z}^5, \boldsymbol{z}^2 \parallel \boldsymbol{z}^5, \boldsymbol{z}^3 \parallel \boldsymbol{z}^4, \boldsymbol{z}^3 \parallel \boldsymbol{z}^5$. The ideal objective vector is equal to $\boldsymbol{z}^* = (z_1^1, z_2^4)$. If we form two approximation sets, $Z^{\mathrm{A}} = \{\boldsymbol{z}^1, \boldsymbol{z}^3, \boldsymbol{z}^4\}$ and $Z^{\mathrm{B}} = \{\boldsymbol{z}^2, \boldsymbol{z}^5\}$, then we can see that $Z^{\mathrm{A}}$ weakly dominates $Z^{\mathrm{B}}$. In fact, $Z^{\mathrm{A}}$ is the Pareto front in this case. In addition, it holds that $Z^{\mathrm{A}}_{\mathrm{EPH}} = F$ and $Z^{\mathrm{B}}_{\mathrm{EPH}} = Z^{\mathrm{B}}$.

## 2.2 Empirical Attainment Function

Based on the multiobjective optimization concept of goal attainment (Grunert da Fonseca et al., 2001) we say that an objective vector is *attained* when it is weakly dominated by an approximation set, i.e. when it is contained in the set's EPH. The surface delimiting the attained vectors from the ones that have not been attained by an approximation set is called the *attainment surface* and its minimal elements are called the *attainment anchors*.

**Definition 2.13 (Set boundary).** The boundary of a set $Z \subseteq F$ is the intersection of the closure of $Z$ with the closure of its complement:

$$\partial Z = \overline{Z} \cap \overline{(F \setminus Z)}. \tag{2.11}$$

**Definition 2.14 (Attainment surface, attainment anchors).** Let $Z \subseteq F$ be an approximation set. The *attainment surface* of $Z$ is the boundary of the EPH of $Z$:

$$S_Z = \partial Z_{\mathrm{EPH}}. \tag{2.12}$$

The minimal elements of $S_Z$ are called *attainment anchors* and are equal to the original approximation set $Z$:

$$A_Z = S_Z^{\min} = Z. \tag{2.13}$$

Now, imagine that an optimization algorithm is run $r$ times, producing $r$ approximation sets. Then, each objective vector can be attained between 0 and $r$ times.

**Definition 2.15 (Empirical attainment function).** For algorithm A the *empirical attainment function* (EAF) of the objective vector $\boldsymbol{z} \in Z$ gives the frequency of attaining $\boldsymbol{z}$ by its approximation sets $Z_1^{\mathrm{A}}, \ldots, Z_r^{\mathrm{A}}$:

$$\alpha_r^{\mathrm{A}}(\boldsymbol{z}) = \alpha(Z_1^{\mathrm{A}}, \ldots, Z_r^{\mathrm{A}}; \boldsymbol{z}) = \frac{1}{r} \sum_{i=1}^{r} \boldsymbol{I}\left(Z_i^{\mathrm{A}} \preceq \{\boldsymbol{z}\}\right), \tag{2.14}$$

where $I$ is the indicator function, defined as

$$I(b) = \begin{cases} 1 & \text{if } b \text{ is true,} \\ 0 & \text{otherwise,} \end{cases} \tag{2.15}$$

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

This means that, given an algorithm A and its approximation sets $Z_1^{\mathrm{A}}, \ldots, Z_r^{\mathrm{A}}$, an EAF value from the set of frequencies $\{0, 1/r, 2/r, \ldots, 1\}$ can be assigned to every vector in the objective space. Of course, in practice the EAF cannot be computed for every objective vector and only attainment surfaces with a constant EAF value (and their corresponding attainment anchors) are of interest. They are called $k$ %-*attainment surfaces* (also *summary attainment surfaces*) in order to be distinguished from the "ordinary" attainment surfaces of single approximation sets. Vectors in the $k$ %-attainment surface are the tightest objective vectors that have been attained in at least $k$ % of the runs.

**Definition 2.16 (Summary attainment surfaces).** Let $Z_1, \ldots, Z_r$ be $r$ approximation sets of algorithm $A$ and $\alpha_r^{\mathrm{A}}(\boldsymbol{z})$ its EAF. Let the $t/r$-superlevel set of $\alpha_r^{\mathrm{A}}$ be defined as

$$L_{t/r}^{+}(\alpha_r^{\mathrm{A}}) = \{\boldsymbol{z} \in F; \alpha_r^{\mathrm{A}}(\boldsymbol{z}) \geq t/r\} \text{ for all } t \in \{1, \ldots, r\}. \tag{2.16}$$

Then the *summary* (or $t/r$-)*attainment surface* $S_{t/r}$ is equal to the boundary of $L_{t/r}^{+}(\alpha_r^{\mathrm{A}})$:

$$S_{t/r} = \partial L_{t/r}^{+}(\alpha_r^{\mathrm{A}}) \text{ for all } t \in \{1, \ldots, r\}. \tag{2.17}$$

We will use $A_{t/r}$ to denote the attainment anchors of $S_{t/r}$:

$$A_{t/r} = S_{t/r}^{\min} \text{ for all } t \in \{1, \ldots, r\}. \tag{2.18}$$

The summary attainment surfaces that correspond to 1/r %, $\sim 50\%$ and 100%-attainment surfaces are called the *best*, *median* and *worst summary attainment surfaces*, respectively.

Note how the definition of an attainment surface coincides with the definition of a summary attainment surface if $r = 1$. In general, the number of all attainment anchors (including duplicates), $n_A = \sum_{t=1}^{r} |A_{t/r}|$, is much larger than the number of objective vectors contained in the approximation sets, $n_Z = \sum_{t=1}^{r} |Z_t|$. It has been shown by Fonseca, Guerreiro, López-Ibáñez, and Paquete (2011) that $n_A \in O(r^2 n_Z)$ for three-objective optimization problems[1].

We illustrate these concepts in the two-objective case. Figure 2.2 shows an example of four approximation sets $Z_1, \ldots, Z_4$ (each consisting of four objective vectors indicated with crosses) and their summary attainment surfaces. Dots denote all attainment anchors and colors are used to emphasize areas with equal EAF values (darker colors correspond to higher values).

If two algorithms A and B are run $r$ times each, they can be compared by computing and visualizing their EAF values $\alpha_r^{\mathrm{A}}$ and $\alpha_r^{\mathrm{B}}$ separately. However, there exists a straightforward way to directly compare the algorithms by computing *EAF differences*, i.e. differences in their EAF values (López-Ibáñez et al., 2010).

[1]In the context of the EAF computation (Fonseca et al., 2011), vectors from the approximation sets and attainment anchors were called *input* and *output points*, respectively.

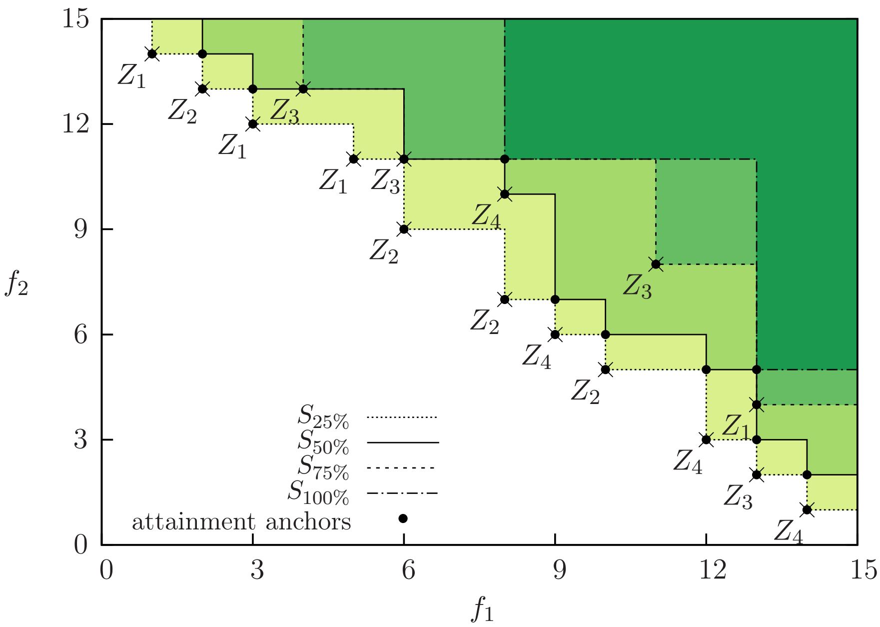



Figure 2.2: EAF values. The plot presents four approximation sets in a two-objective optimization problem, their summary attainment surfaces, and all attainment anchors. Colors denote areas with equal EAF values (darker colors correspond to larger values).

**Definition 2.17 (EAF differences).** Assume that algorithms A and B are run $r$ times each. The *EAF differences* between algorithms A and B are defined for each objective vector $\boldsymbol{z} \in Z$ as

$$\delta_r^{\mathrm{A-B}}(\boldsymbol{z}) = \alpha_r^{\mathrm{A}}(\boldsymbol{z}) - \alpha_r^{\mathrm{B}}(\boldsymbol{z}). \tag{2.19}$$

Note that EAF differences need to be computed for all attainment anchors of the combined $2r$ approximation sets. Defined in this way, the differences can adopt values from the set $\{-1, -(r-1)/r, \dots, 0, 1/r, \dots, 1\}$. Positive EAF differences correspond to areas in 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.

Let us focus on the example from Figure 2.3, where algorithms A and B solving a two-objective optimization problem produced two approximation sets each. The lines show the overall summary attainment surfaces. The colored areas emphasize areas of the objective space where algorithm A outperformed algorithm B (green hues) and areas where algorithm B outperformed algorithm A (red hues). We can readily notice that algorithm A is more successful in minimizing the first objective, while algorithm B better attains lower values of the second objective. This small example nevertheless shows that visualization of 2D EAF differences can be very helpful when analyzing and comparing the results of two algorithms.

We can safely assume that we are interested in a limited portion of the objective space, for example, the one enclosed by reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$, where $\boldsymbol{r}^1 \prec\!\!\prec \boldsymbol{r}^2$. Then, areas with equal EAF values/differences correspond to unions of rectangles in 2D and unions of cuboids in 3D. If we allow overlapping, then each rectangle or cuboid between the $t/r$- and $(t+1)/r$-attainment surfaces, where $t \in \{1, \dots, r-1\}$, can be defined by two vertices—one from the $t/r$-attainment surface and the other from the $(t+1)/r$-attainment

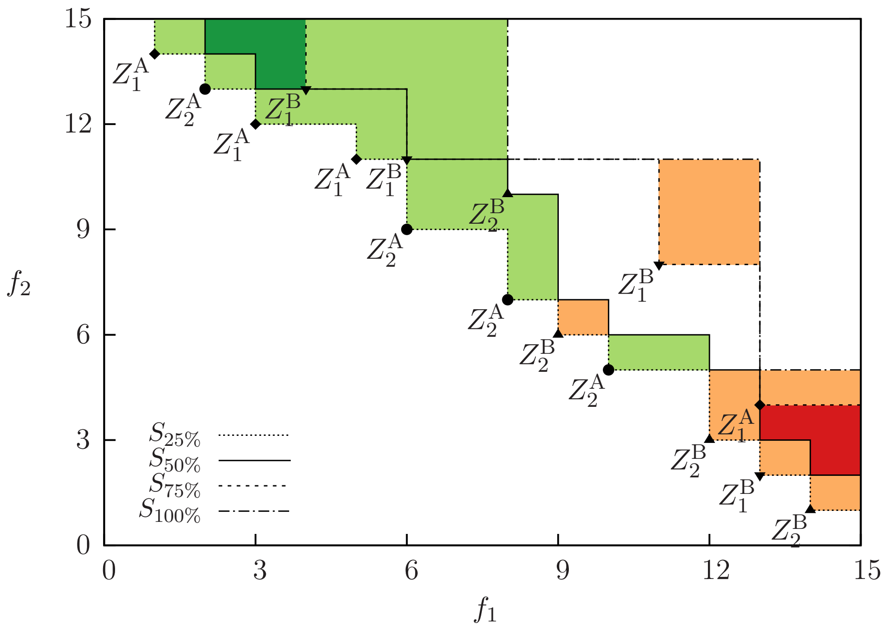



Figure 2.3: EAF differences. The plot presents four approximation sets in a two-objective optimization problem—two produced by algorithm A and two by algorithm B. Colored areas show where algorithm A outperformed algorithm B (green hues) and algorithm B outperformed algorithm A (red hues) Darker colors correspond to larger differences.

surface (Figure 2.4 shows 2D areas between 25%- and 50%-attainment surfaces). While the lower vertices correspond to the attainment anchors of the $t/r$-attainment surface, the upper vertices are not the attainment anchors of the $(t+1)/r$-attainment surface, but their *opposite*. The opposite can be defined for an arbitrary approximation set as follows.

**Definition 2.18 (Approximation set opposite).** Let reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$, where $\boldsymbol{r}^1 \prec\!\!\prec \boldsymbol{r}^2$, define the boundaries of the observed objective space:

$$R = \{\boldsymbol{z} \in F; z_i \in [r_i^1, r_i^2] \text{ for all } i \in \{1, \dots, m\}\}. \tag{2.20}$$

Let $Z \subseteq R$ be an approximation set enclosed in this space with the attainment surface $S_Z$. The *opposite* $O_Z^R$ of the approximation set $Z$ within the observed objective space $R$ is the maximal set of $S_Z \cap R$:

$$O_Z^R = (S_Z \cap R)^{\max}. \tag{2.21}$$

Finding the opposites of attainment anchors in 2D is rather trivial (Tušar & Filipič, 2014a). The computation of opposites (and the corresponding cuboids) for the 3D case is more demanding and is presented in detail in Section 7.2.1.

Finally, let us recall the formal definition of the *hypervolume indicator* (Zitzler, Brockhoff, & Thiele, 2001), which is often used to measure the quality of approximation sets.

**Definition 2.19 (Hypervolume indicator).** Let again reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$, where $\boldsymbol{r}^1 \prec\!\!\prec \boldsymbol{r}^2$, define the boundaries of the observed objective space

$$R = \{\boldsymbol{z} \in F; z_i \in [r_i^1, r_i^2] \text{ for all } i \in \{1, \dots, m\}\}.$$

The *hypervolume indicator* $I_{\mathrm{H}}$ of an approximation set $Z \subseteq R$ can be formulated via the empirical attainment function of $Z$ as

$$I_{\mathrm{H}}(Z) = \int_{\boldsymbol{z} \in R} \alpha(Z; \boldsymbol{z}) d\boldsymbol{z}.$$

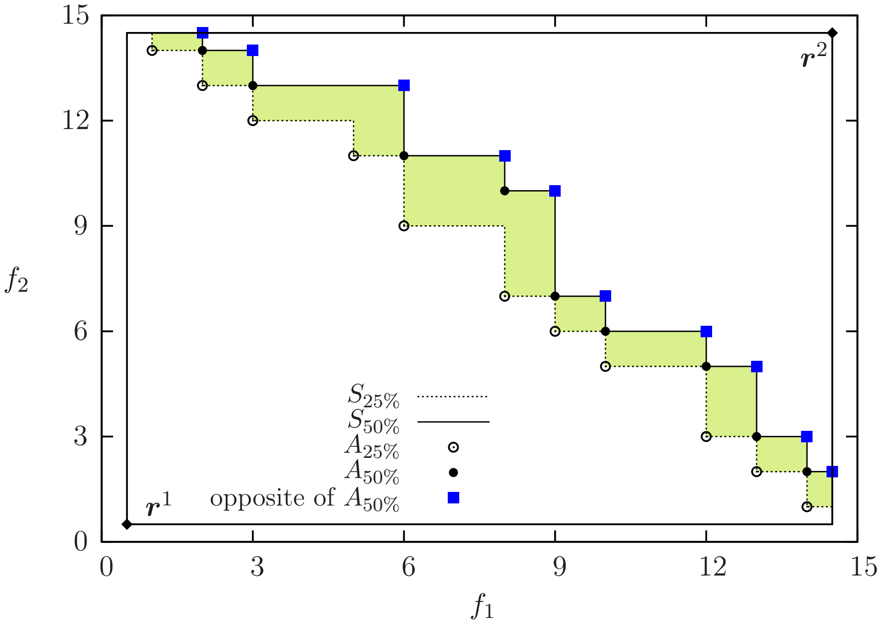



Figure 2.4: Opposite of an approximation set. The plot presents the area between the 2D 25%- and 50%-attainment surfaces from the example in Figure 2.2. Vectors from the opposite of the attainment anchors of the 50%-attainment surface are denoted with squares.

# Part I

# Visualization of Approximation Sets

# Chapter 3

# Benchmark Approximation Sets

This chapter presents the motivation for proposing benchmark approximation sets (Section 3.1) and two instances of such sets, the linear and spherical benchmark approximation sets (Sections 3.2 and 3.3, respectively).

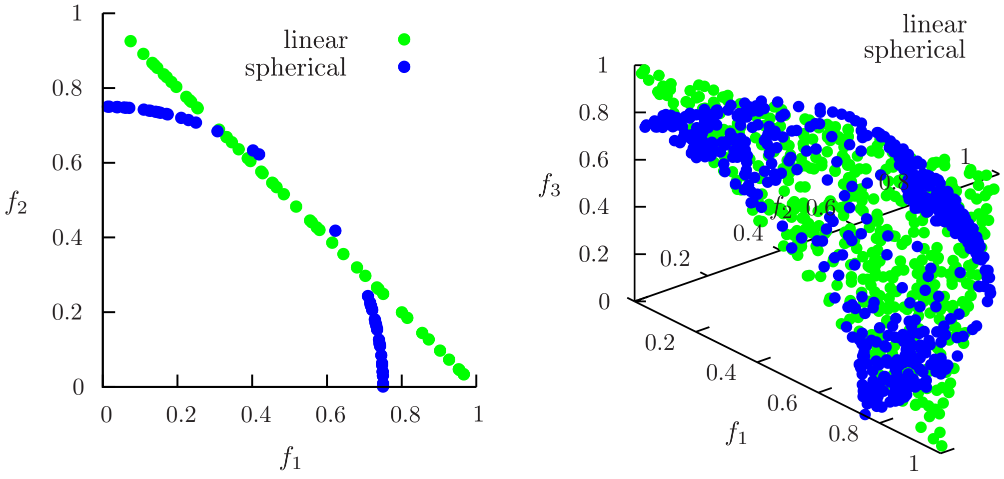
## 3.1 Motivation

In the field of evolutionary multiobjective optimization, many benchmark problems exist, such as the DTLZ (Deb, Thiele, Laumanns, & Zitzler, 2005) and WFG (Huband, Hingston, Barone, & While, 2006) test suites, which are used for comparing the performance of MOEAs. However, no benchmark sets exist that could analogously be used for comparing visualization methods. In fact, no serious attempt to compare visualization methods has been made in this field so far. For this purpose, we introduce the concept of *benchmark approximation sets* (BASes) to be used when comparing visualization methods.

Based upon the requirements for visualization methods from Section 1.1.1, we can list some specific demands a suite of BASes should conform to. The idea is that BASes should have some distinct properties that can be used when assessing how visualization methods fulfill the aforementioned requirements. As any BAS can consist only of mutually nondominated vectors, BASes in the same suite need to *dominate each other entirely or in part*. This property is important if we wish to inspect whether the visualization methods manage to (partially) preserve the Pareto dominance relation. Next, to be able to assess the preservation of the shape of approximation sets, BASes should be of *different shapes*, such as linear, concave, convex, mixed, degenerated, discontinuous, with knees (and possibly others). In order to visualize a different distribution of vectors, BASes in the same suite should have *uniform as well as different kinds of nonuniform distributions of vectors*. Also, BASes should be *scalable to many dimensions* to check the scalability of the visualization methods. While no specific requirements are needed to inspect the preservation of the objective range, robustness and simplicity of the visualization methods, in order to assess their capability to visualize multiple large sets, the BASes should be of a *large size*—appropriate to their dimensionality.

Following these guidelines, we could come up with a considerable suite of BASes, much like the existing suites of benchmark problems. However, visualization methods cannot be compared as efficiently as optimization algorithms because their outcome on each BAS cannot be measured but must be visualized. Therefore, the size of such a suite of BASes is limited by the number of visualization methods we wish to compare and the amount of space we have to present the results. It is for this reason that the number of BASes to be used throughout this thesis is limited to two.

The two BASes will be denoted as *linear* and *spherical* according to their shapes.




(a) 2D BASes (b) 3D BASes

Figure 3.1: 2D and 3D instances of benchmark approximation sets.

These two (rather simple) shapes were chosen among others as they appear most often in benchmark problems used in the field of evolutionary multiobjective optimization. The two BASes can be instantiated in any dimension (see Figure 3.1 for their 2D and 3D instances) and have different distributions and ranges of vectors. In addition, for any dimension $m$ they are intertwined—in one region, the linear BAS dominates the spherical one, while in others, the spherical dominates the linear one. In the thesis we deal with the BASes in four instances of different dimensionality and/or cardinality: 2D with 50 vectors in each BAS, 3D with 500 vectors in each BAS, and 4D with 300 and 3000 vectors in each BAS.

## 3.2 Linear BAS

The first BAS is linear with all objective vectors satisfying the following constraint:

$$\sum_{i=1}^{m} z_i = 1, \tag{3.1}$$

where each $z_i \in [0, 1]$ and $m$ is the number of objectives. The vectors in the linear BAS are created using Algorithm 3.1 and are uniformly randomly distributed (Rubinstein & Kroese, 2008).

## 3.3 Spherical BAS

The second BAS is spherical with all objective vectors satisfying the following constraint:

$$\sum_{i=1}^{m} z_i^2 = 0.75^2, \tag{3.2}$$

where each $z_i \in [0, 0.75]$ and $m$ is the number of objectives. The vectors in the spherical BAS have a nonuniform distribution—only few vectors are located in the middle of the approximation set, while most of them are near its corners. Therefore an $m$D spherical BAS has exactly $m$ regions with a high density of vectors. With this property we try

**Algorithm 3.1:** Generating a vector in the linear BAS. The uniformRand() function returns a random number from the interval $[0, 1]$ with a uniform distribution. The sortAscending($w_0$, $w_1, \ldots, w_m$) function sorts the values $w_i$ so that $w_0 \leq w_1 \leq \cdots \leq w_m$.

**Input**: Number of objectives $m$
**Output**: The objective vector $(z_1, \ldots, z_m)$

$w_0 \leftarrow 0$;
**for** $i = 1, \ldots, m-1$ **do**
  $w_i \leftarrow$ uniformRand();
**end**
$w_m \leftarrow 1$;
sortAscending$(w_0, w_1, \ldots, w_m)$;
**for** $i = 1, \ldots, m$ **do**
  $z_i \leftarrow w_i - w_{i-1}$;
**end**

to achieve two goals: 1) a nonuniform distribution of vectors different from the uniform one of the linear BAS, and 2) a few almost unconnected regions aimed at mimicking the discontinuous fronts.

Algorithms 3.2 and 3.3 present the algorithms used to create the spherical BAS. Lines 4 to 8 of Algorithm 3.2 assure that $m$ regions with a high density of vectors are created. These are then projected on the sphere with radius 0.75 in Lines 9 to 17. Algorithm 3.3 shows how the nonuniform "U-shaped" distribution is created from the Gaussian distribution[1].

**Algorithm 3.2:** Generating a vector in the spherical BAS. The nonuniformRand() function is presented in Algorithm 3.3.

**Input**: Number of objectives $m$
**Output**: The objective vector $(z_1, \ldots, z_m)$

1 **for** $i = 1, \ldots, m$ **do**
2   $w_i \leftarrow 0$;
3 **end**
4 **while** $\sum_{i=1}^{m} w_i^2 > 1$ or ($w_i < 0.5$ for all $i \in \{1, \ldots, m\}$) **do**
5   **for** $i = 1, \ldots, m$ **do**
6     $w_i \leftarrow$ nonuniformRand();
7   **end**
8 **end**
`// Project` $(w_1, \ldots, w_m)$ `onto the sphere`
9 **for** $i = 1, \ldots, m-2$ **do**
10   $\theta_i \leftarrow \arctan \frac{\sqrt{w_{i+1}^2 + \cdots + w_m^2}}{w_i}$;
11 **end**
12 $\theta_{m-1} \leftarrow 2 \arctan \frac{w_m}{\sqrt{w_{m-1}^2 + w_m^2} + w_{m-1}}$;
13 $z_1 \leftarrow 0.75 \cos(\theta_1)$;
14 **for** $i = 1, \ldots, m-1$ **do**
15   $z_i \leftarrow 0.75 \prod_{j=1}^{i-1} \sin(\theta_j) \cos(\theta_i)$;
16 **end**
17 $z_m \leftarrow 0.75 \sin(\theta_1) \cdots \sin(\theta_{m-2}) \sin(\theta_{m-1})$;

[1] This distribution was preferred to the beta distribution (which is also "U-shaped") because its resulting regions with a high density of vectors are more distinct than those achieved with the beta distribution.

**Algorithm 3.3:** Generating a random value in the interval $[0, 1)$ with a nonuniform distribution. The gaussianRand() function returns a random number with a normal distribution.

**Output**: The value $r$

```
r ← 3; // Set to a value outside [-2, 2]
while r < −2 or r > 2 do
    r ← gaussianRand();
end
if r ≥ 0 then
    r ← r/4;
else
    r ← 1 + r/4;
end
```

# Chapter 4

# Review of Related Visualization Methods

Numerous methods exist for visualizing multidimensional data, see for example (dos Santos, 2004; Miettinen, 2014). As we are interested only in methods suitable for visualizing approximation sets, the review in this chapter is restricted to visualization methods that were previously used for this purpose. We divide these methods into two groups: *general* (Section 4.1) and *specific* (Section 4.2), according to their ability to handle the unique features of approximation sets. Chapter 3 declared the usage of 4D BASes in two instances: small BASes containing only 300 vectors and large BASes with 3000 vectors. This review provides for each related method its visualization of the small 4D BASes, because visualizing the large BASes represents a difficulty for most of the methods.

In addition, Table 4.1 summarizes the properties of all methods in view of the requirements for visualization methods presented in Section 1.1.1. The review of related methods ends with the presentation of orthogonal prosections as used for visualizing abstract mathematical models (Section 4.3).

## 4.1 General Multidimensional Data Visualization Methods

The general multidimensional data visualization methods are those introduced outside the field of evolutionary multiobjective optimization and thus make no effort to preserve the Pareto dominance relation between vectors or any other feature specific to evolutionary multiobjective optimization. Therefore we review these methods mainly with regard to their ability to distinguish between the different shape and distribution of vectors of the two BASes.

### 4.1.1 Scatter Plot Matrix

A straightforward visualization method is to project all vectors to a lower-dimensional space by disregarding all the dimensions of the vector that are beyond those that can be visualized. If this is done for all possible combinations of these lower-dimensional spaces, a scatter plot matrix is obtained (see Figure 4.1). The scatter plot matrix is a very fast, simple and robust visualization method that in our case retains some information on the shape of the approximation sets (it is easy to distinguish between the spherical and linear BAS) as well as the different distribution of vectors.

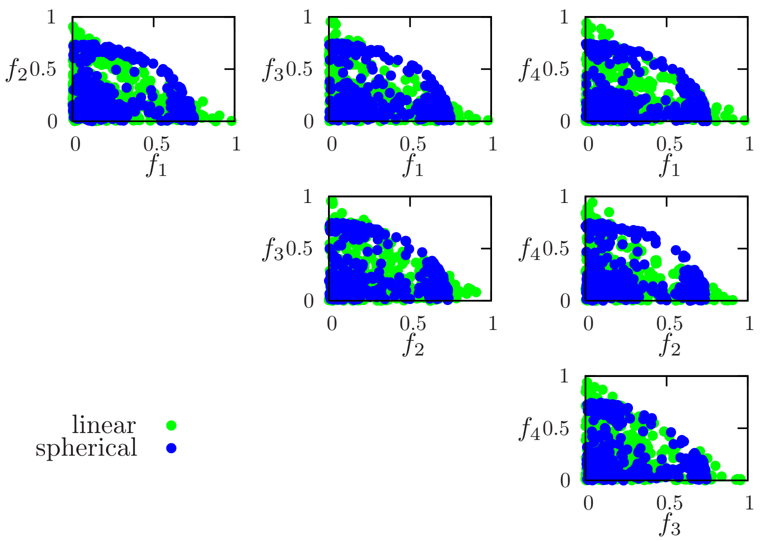



Figure 4.1: Scatter plot matrix.

### 4.1.2 Bubble Chart

In a scatter plot, additional dimensions can be visualized using size (4D) and color (5D), thus obtaining a bubble chart (Ashby, 2000; Poles, Geremia, Campos, Weston, & Islam, 2007). Liebscher, Witowski, and Goel (2009) call it a *trade-off plot*. From Figure 4.2 we can observe that the bubble chart has the same advantages and disadvantages as the scatter plot matrix. The main benefit of the bubble chart over the scatter plot matrix is that all the information is given in a single plot.

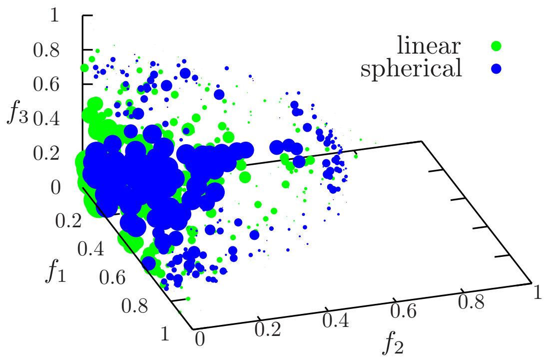



Figure 4.2: Bubble chart.

### 4.1.3 Radial Coordinate Visualization

The idea for Radial Coordinate Visualization (RadViz) comes from physics (Hoffman, Grinstein, Marx, Grosse, & Stanley, 1997). The objectives (called dimensional anchors) are distributed evenly on the circumference of the unit circle. Imagine that each objective vector is held with springs that are attached to the anchors and the spring force is proportional to the value in the corresponding objective/anchor. The position of the objective vector is the one where the spring forces are in equilibrium. For example, objective vectors that are placed close to $f_i$ have a higher value in $f_i$ than in any other objective, while objective vectors with all equal values are placed exactly in the center of the circle. RadViz was used under the name of *barycentric coordinates* to visualize approximation sets by Walker, Fieldsend, and Everson (2012, 2013). While the RadViz of our two BASes (see Figure 4.3) is able to preserve well the distribution of vectors of both sets, we cannot distinguish their shape.

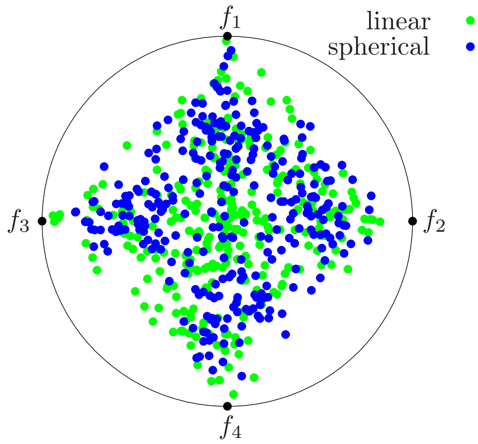



Figure 4.3: Radial coordinate visualization.

### 4.1.4 Parallel Coordinates

Using parallel coordinates (Inselberg, 2009), each $m$-dimensional objective vector is represented as a polyline with vertices on the parallel axes, where the position of the vertex on the $i$th axis corresponds to the $i$th coordinate of the objective vector. Parallel coordinates are very useful for representing (in)dependences between objectives. In our case (see Figure 4.4), the clutter created by numerous polylines conceals the distribution of objective vectors, which could be seen otherwise. Although not able to show the shape of approximation sets, parallel coordinates are frequently used for visualizing results in evolutionary multiobjective optimization.

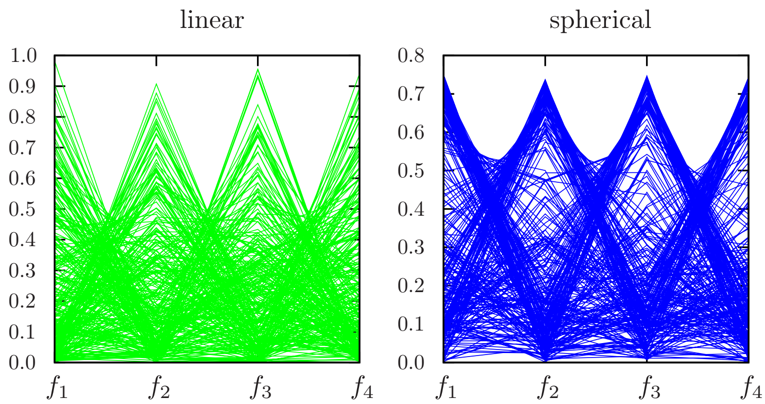



Figure 4.4: Parallel coordinates.

### 4.1.5 Heatmaps

In a heatmap, objective values are shown using color (Pryke, Mostaghim, & Nazemi, 2007). Similarly to the parallel coordinates plot, heatmaps can show (in)dependences between objectives. See Figure 4.5, where the objective vectors in each heatmap are sorted by the value of the first objective. Although with this visualization no information is lost, for our two BASes not much information could be gained either.

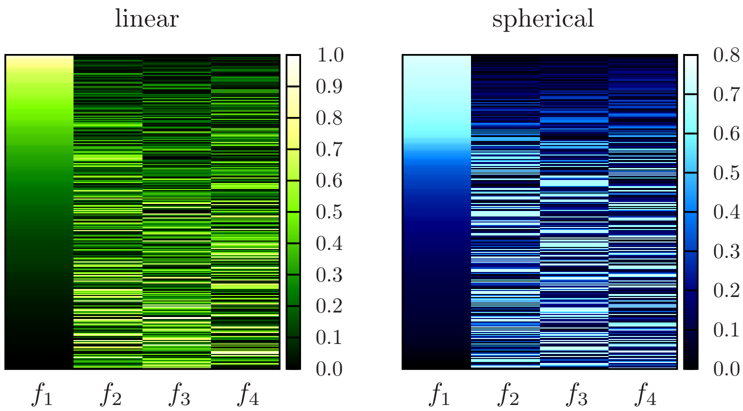



Figure 4.5: Heatmaps.

The five methods presented so far are very simple to understand and compute—they do not require sophisticated mappings of objective vectors—and are therefore very fast. RadViz, parallel coordinates and heatmaps can also be easily scaled in many dimensions and the latter two are able to visualize the decision space together with the objective space. Moreover, the scatter plot matrix and parallel coordinates can be enhanced using the *brushing and linking* interaction techniques (Martin & Ward, 1995). Brushing means selecting a subset of data (in a single scatter plot or on a single parallel coordinate), while linking causes this data to be highlighted also on the other views (other scatter plots or other parallel coordinates).

In our case, the combination of the bubble chart and RadViz provides most information. The next five methods use some more sophisticated mapping to perform dimension reduction to the 2D space.

### 4.1.6 Sammon Mapping

Sammon mapping (Sammon, 1969) aims to minimize the stress function, which emphasizes the preservation of the local distances. This means that distances between the mapped vectors[1] are required to be as similar as possible to distances between the original vectors. The minimization can be performed either by gradient descent, as proposed initially, or by other means, usually involving iterative methods. See (Valdés & Barton, 2007) for the use of Sammon mapping in evolutionary multiobjective optimization. In our case (Figure 4.6) the Sammon mapping preserves very well the distribution of vectors—all four regions with a high density of vectors of the spherical BAS are manifested.

[1] Recall that vectors are regarded as position vectors and the distances are computed between their corresponding points in the Euclidean space.

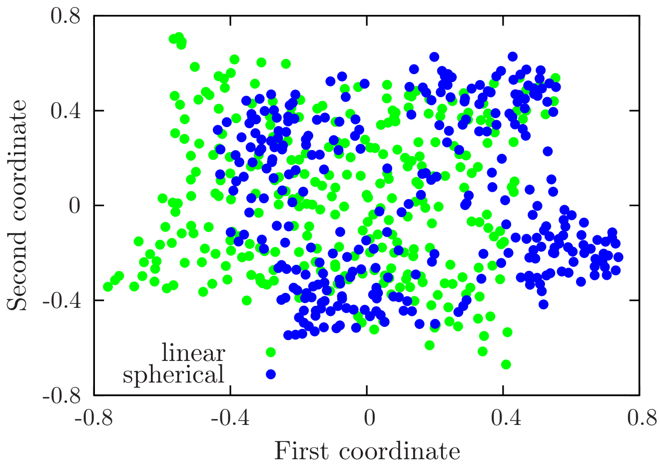



Figure 4.6: Sammon mapping.

### 4.1.7 Neuroscale

Neuroscale (Lowe & Tipping, 1996) pursues the same goal as Sammon mapping (the preservation of distances) using a radial basis function neural network to minimize the stress function. See (Everson & Fieldsend, 2006) for neuroscale representations of an approximation set and Figure 4.7 for the visualization of our BASes. Neuroscale does not differentiate well between the two BASes. Moreover, it is the only method that skews the distribution of vectors in the linear BAS.

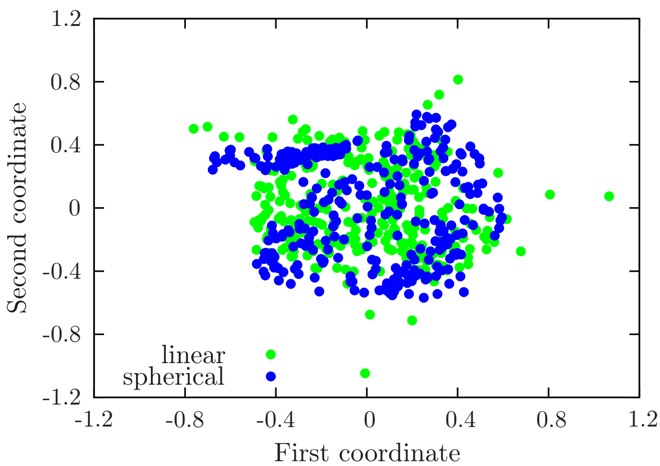



Figure 4.7: Neuroscale.

### 4.1.8 Self-Organizing Maps

Self-Organizing Maps (SOMs) by Kohonen (2001) are artificial neural networks that provide a topology-preserving mapping from $m$D to a lower dimension (usually 2D). This means that nearby vectors in the input space are mapped to nearby units (called neurons) in the SOM. While different arrangements of neurons are possible, we use the hexagonal grid as was done by Obayashi and Sasaki (2003). When trained, the SOMs can be visualized using different methods. One of the most popular is the U-matrix (unified distance matrix), where the distance between adjacent neurons is presented with different colorings. Light areas represent clusters of similar neurons and dark areas indicate cluster boundaries. Figure 4.8 presents the U-matrices of our two BASes. The SOM of the linear BAS correctly puts all neurons in a single cluster, while it is difficult to interpret the SOM of the spherical BAS (the exact number of clusters is hard to establish).

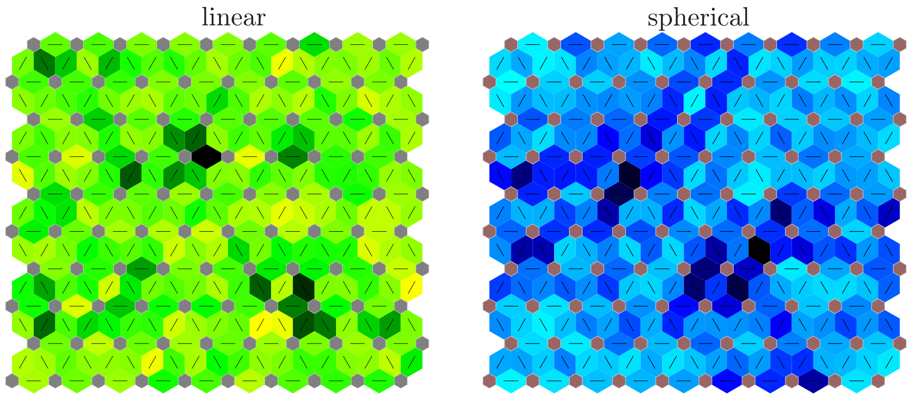



Figure 4.8: Self-organizing maps.

### 4.1.9 Principal Component Analysis

Principal Component Analysis (PCA) finds a new lower-dimensional set of coordinates (the principal components) so that projection onto the principal components captures the maximum variance among all linear projections. The principal components are easily found as the eigenvectors with the highest eigenvalues of the covariance matrix of a set of vectors. PCA was used for visualization in the field of evolutionary multiobjective optimization by Yamamoto, Yoshikawa, and Furuhashi (2010). The objective vectors from our BASes are mapped to the space of the principal components as shown in Figure 4.9. Two of the regions with a high density of vectors in the spherical BAS are visualized as one.

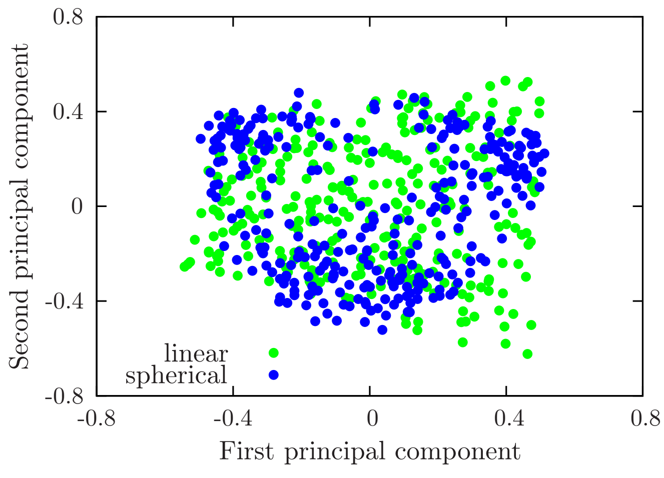



Figure 4.9: Principal component analysis.

### 4.1.10 Isomap

The basic idea behind Isomap (Tenenbaum, de Silva, & Langford, 2000) is to preserve the intrinsic geometry of the data when mapping to the 2D space using multidimensional scaling (Borg & Groenen, 2005). In a graph of vectors, where each vector is linked only to its closest neighbors, the *geodesic distance* between two vectors is calculated as the sum of Euclidean distances of the shortest path between the two vectors in the graph. The presumption of Isomap is that the vectors lie on some low-dimensional manifold and the distances between vectors along this manifold should be preserved. Isomap was used to visualize approximation sets by Walker et al. (2012), while Kudo and Yoshikawa (2012) propose to construct the graph of vectors using distances in the objective space and calculating the geodesic distances in the decision space. As no decision space is given for our BASes, Figure 4.10 shows the usual Isomap. We can see all four regions with a high density of vectors of the spherical BAS.

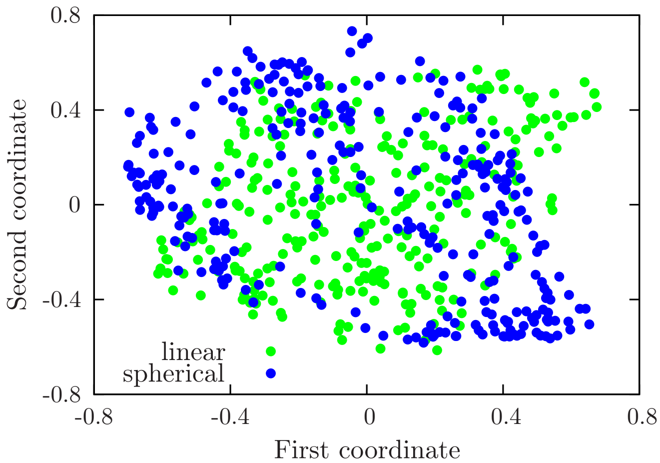



Figure 4.10: Isomap.

The latter five visualization methods are also scalable to many dimensions. However, they are more elaborate, difficult to understand and implement and computationally more expensive than the previous methods. They are also less robust than the previous methods as the mapping used for visualization depends on the values of the objective vectors in the approximation sets. Among these five methods, Sammon mapping was the best at distinguishing the distribution of vectors of the two BASes.

## 4.2 Methods Specifically Designed for Visualization of Approximation Sets

This section reviews the methods that are tailored to the visualization of multidimensional approximation sets.

### 4.2.1 Distance and Distribution Charts

Ang, Chong, and Li (2002) propose to plot the objective vectors against their distance to some approximation of the Pareto front (distance chart) and their distance between each other (distribution chart). As the exact computation of the distribution of vectors is very time-consuming when the number of objectives is high, Ang et al. (2002) suggest to use a simpler computation that does not produce exact results. Using the latter version, the distance and distribution charts of our BASes are presented in Figure 4.11. Here, the approximation of the Pareto front consists of nondominated vectors from both sets. The distance chart correctly shows how most of the vectors from the spherical approximation set are dominated, while this holds for only a few vectors from the linear approximation set. However, the distribution chart fails to differentiate between two BASes with a very different distribution of vectors. This might be due to the non-exact computation of the distribution metric, however, despite possible reasons, in our view such charts fail to provide an intuitive presentation of the 4D BASes.

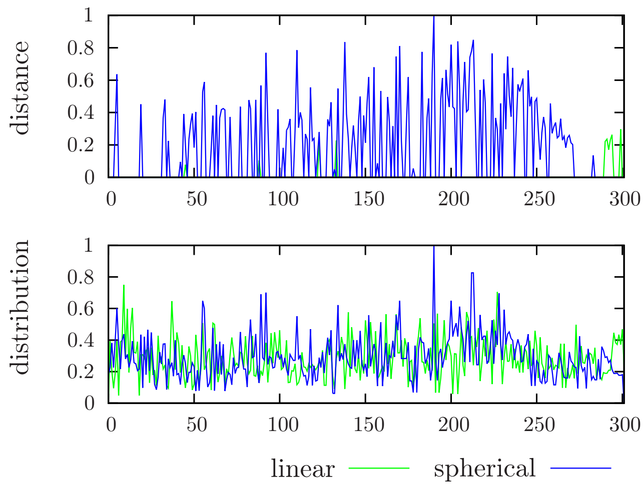



Figure 4.11: Distance and distribution charts.

### 4.2.2 Interactive Decision Maps

In contrast to other methods, Interactive Decision Maps (IDMs) visualize the Edgeworth-Pareto hull of an approximation set (Lotov, Bushenkov, & Kamenev, 2004; Lotov & Miettinen, 2008). This means that instead of visualizing only a finite number of vectors of an approximation set, IDMs visualize a number of axis-aligned sampling surfaces of the EPH. As this is possible only for visualizing 2D and 3D approximation sets, interactive choice of the value of the fourth objective is used for visualizing decision maps of 4D approximation sets. See the example of the IDMs with $f_4$ fixed to 0.5 of our two BASes in Figure 4.12. They give a good idea about the shape of the approximation set and also somewhat about the distribution of vectors. However, from the plots it is impossible to infer any dominance relation between the vectors of the two sets.

### 4.2.3 Hyper-Space Diagonal Counting

This method builds upon the idea that the set of natural numbers $\mathbb{N}$ has the same cardinality as the set $\mathbb{N}^m$, where $m \in \mathbb{N}$. Therefore, the set $\mathbb{N}^m$ can be mapped into $\mathbb{N}$ using the hyper-space diagonal counting as described by Agrawal et al. (2004). Consider now the case of an approximation set in 4D. Its visualization using hyper-space diagonal counting is performed as follows (Agrawal, Bloebaum, & Lewis, 2005). First, each objective is divided into a predefined number of bins. The bins of a pair of objectives are then counted using hyper-space diagonal counting producing indices for this pair of objectives. These indices are plotted on two axes (one for each pair of objectives), while the third axis is

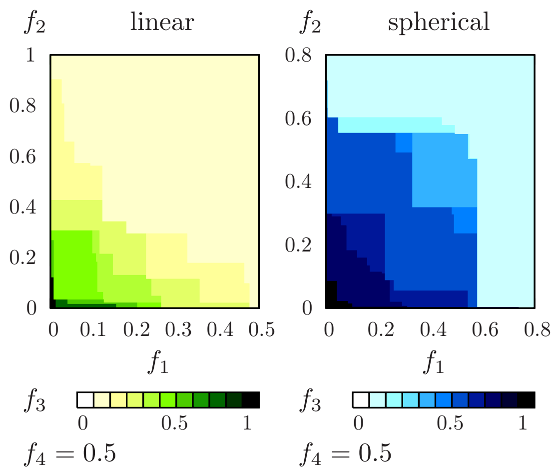



Figure 4.12: Interactive decision maps.

used to plot the number of vectors of the approximation set that fall in the same set of indices. See Figure 4.13, where hyper-space diagonal counting is used to visualize the two BASes. Arguably, the plot better captures the distribution of vectors than the shape of the approximation sets. Again, this method does not maintain the dominance relations between objective vectors.

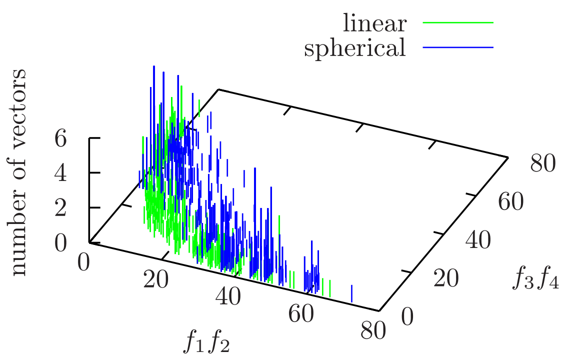



Figure 4.13: Hyper-space diagonal counting.

### 4.2.4 Two-Stage Mapping

The two-stage mapping by Köppen and Yoshida (2007) aims to preserve (as much as possible) the Pareto dominance relations and distances among vectors. In the first stage, all nondominated vectors are mapped onto a quarter-circle. A MOEA (in their case, NSGA-II (Deb, Pratap, Agrawal, & Meyarivan, 2002)) is used to find a permutation of these vectors so that both relations among vectors (Pareto dominance and distance) are preserved as much as possible. When a good permutation is found, the nondominated vectors are mapped onto the circle in the order given by this permutation (and with distances proportional to their mutual distances). In the second stage, each dominated vector is mapped to the minimal vector of all nondominated vectors that dominate it. Figure 4.14 shows the result of the two-stage mapping for the two BASes. Unfortunately, other than the split to dominated and nondominated vectors, not much information can be gathered from this plot, while the visualization method is rather complex (requiring itself to solve a multiobjective optimization problem). The two-stage mapping builds its visualization upon the Pareto dominance relations among vectors, which means that in case of addition of an objective vector to one of the approximation sets (or deletion of an objective vector from an approximation set), the visualization of the sets might change considerably—depending on how well the inherent multiobjective optimization problem is solved. In other words, this method is not robust.

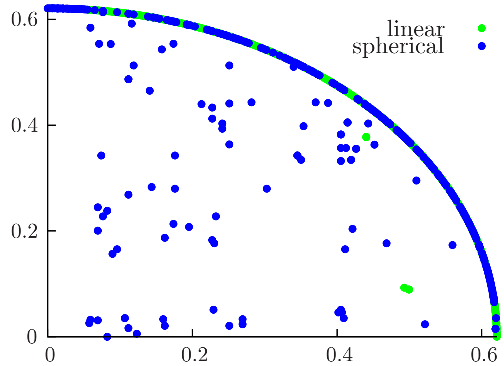



Figure 4.14: Two-stage mapping.

### 4.2.5 Level Diagrams

Blasco, Herrero, Sanchis, and Martínez (2008) propose to plot the approximation sets on a set of $m$ level diagrams, where $m$ is the number of objectives (the decision space can be visualized using this method, too). In each such diagram, objective vectors are sorted according to their value of the corresponding objective and plotted against their distance to the ideal objective vector (different norms can be used). Therefore, each vector has the same $y$ position in all diagrams. See the level diagrams of our two BASes in Figure 4.15, where the Euclidean norm is used for computing distances. While the shape

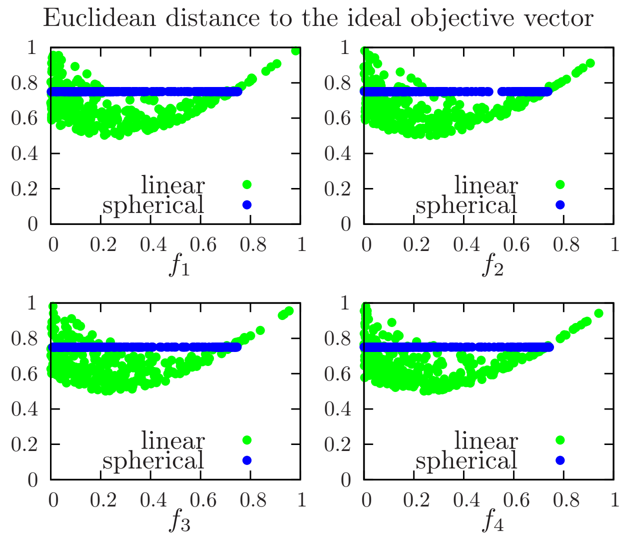



Figure 4.15: Level diagrams.

of the approximation sets can be inferred from this diagrams, this is not the case for the Pareto dominance relations and the distribution of vectors (particularly for the spherical BAS). Nevertheless, this method is simple, computationally inexpensive and can help the decision maker, especially if color is added to show user preferences.

### 4.2.6 Hyper-Radial Visualization

Somewhat similar to level diagrams is the hyper-radial visualization (Chiu & Bloebaum, 2010). Here too, the objective vectors preserve their distance to the ideal objective vector (their hyper-radius), but separately for two subsets of objectives. The resulting visualization of our two BASes (see Figure 4.16) is able to maintain well the shape of the approximation sets, while the distribution of vectors is correctly represented for the linear BAS, but not for the spherical one. The findings from level diagrams can be applied here too: while the Pareto dominance relations are mostly not preserved, the method is simple, computationally inexpensive and valuable for the decision maker if user preferences are color-coded.

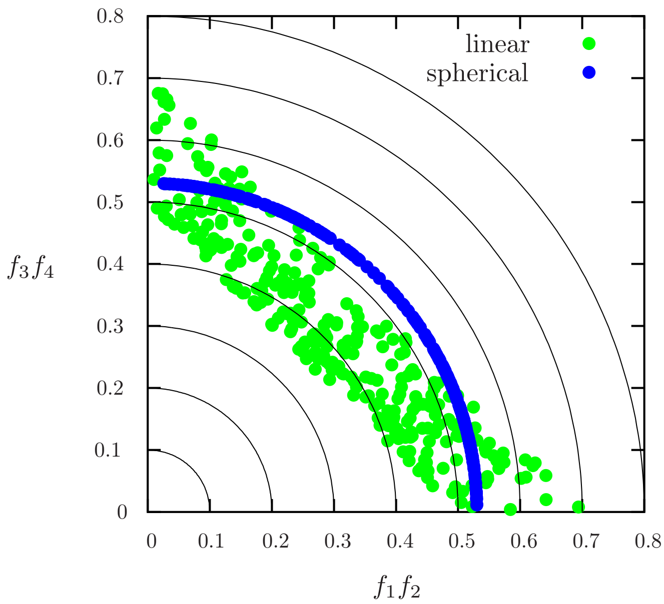



Figure 4.16: Hyper-radial visualization.

### 4.2.7 Pareto Shells

Using the nondominated sorting procedure from NSGA-II, vectors from different approximation sets can be sorted into Pareto shells of mutually nondominated vectors. These shells can be visualized using a graph where nodes represent vectors (arranged according to the shell they belong to) and directed edges represent the Pareto dominance relation between the connected vectors (Walker et al., 2010). Our two BASes are visualized using Pareto shells in Figure 4.17. While this method is somewhat cumbersome for visualizing large approximation sets (in the plot we only draw one edge for each dominated vector as drawing all edges would make the plot too crowded), it clearly shows the Pareto dominance relations between vectors. Of course, all other information (objective ranges, distributions of vectors and the shape of the approximation set) cannot be shown using this method. This is also a non-robust method.

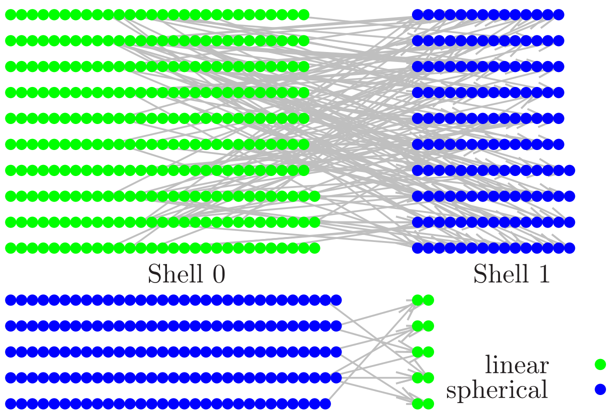



Figure 4.17: Pareto shells.

### 4.2.8 Seriated Heatmaps

As the amount of information that can be retrieved from a heatmap heavily depends on the order of vectors in the heatmap, Walker et al. (2013) propose to seriate heatmaps so that similar vectors (and similar objectives) are placed together. Instead of showing actual objective values, seriated heatmaps present ranks that are assigned to each vector component depending on its objective value. The seriated heatmaps for our two BASes are shown in Figure 4.18. While seriation rearranged the objectives and vectors of both sets, we cannot conclude that seriated heatmaps give us any more information than the regular ones (already presented in Figure 4.5). Note also that because of ranking, seriated heatmaps are not as robust as their predecessors.

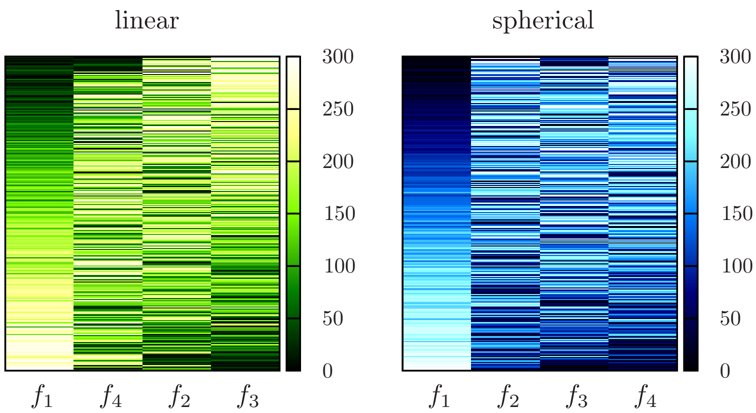



Figure 4.18: Seriated heatmaps.

### 4.2.9 Multi-Dimensional Scaling

The classical Multi-Dimensional Scaling (MDS) tries to find a linear mapping to the 2D space that preserves similarities between vectors (Borg & Groenen, 2005). Simply put, the classical MDS is equivalent to performing PCA on similarities between vectors instead of their distances. Walker et al. (2013) do it using *dominance similarity*, which defines two vectors as similar if they dominate the same vectors. Figure 4.19 shows the MDS of our two BASes using this dominance similarity. Since the dominance similarity takes into account only the relative dominance relations between vectors, the distribution of vectors is lost in such a visualization and the method is even less robust than PCA.

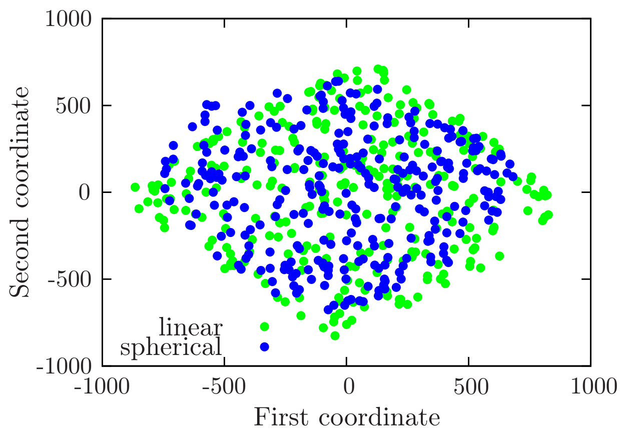



Figure 4.19: Multi-dimensional scaling using dominance similarity.

These nine visualization methods are very different from each other and therefore hard to compare. However, in our opinion, the most useful information on our BASes comes from the interactive decision maps and the hyper-radial visualization.

## 4.3 Orthogonal Prosections

More than the visualization methods described so far, our method resembles the *orthogonal prosections* (Tweedie, Spence, Dawkes, & Su, 1996), which were used for visualizing abstract mathematical models[2]. Their idea is very simple (see Figure 4.20 for the 3D case): instead of projecting the whole set of solutions to the orthogonal plane $p_1p_2$, only the solutions from the chosen section are projected. Because multiple planes can be chosen for the projection (as in the scatter plot matrix), a prosection matrix is used to visualize all orthogonal prosections simultaneously. In addition, the authors used color coding for distinguishing between feasible and infeasible solutions.

To our best knowledge, orthogonal prosections were never before used to visualize approximation sets.

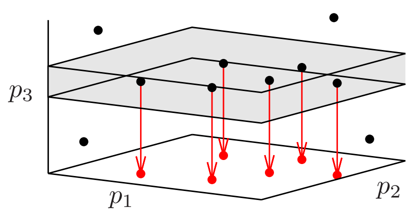



Figure 4.20: Orthogonal prosection. A section of $p_3$ is projected to the orthogonal $p_1p_2$ plane.

[2]Tweedie et al. (1996) call them *prosections*, but we added the adjective *orthogonal* to make a clear distinction between their method and ours.

Table 4.1: Summary analysis of methods for visualizing approximation sets.

| Method | Preservation[a] of the | | | | Robustness | Handling of large sets | Simultaneous visualization | Scalability | Simplicity |
|---|---|---|---|---|---|---|---|---|---|
| | dominance relation | front shape | objective range | distribution of vectors | | | | | |
| Scatter plot matrix | × | ≈ | ✓ | ≈ | ✓ | ≈[b] | ✓ | × | ✓ |
| Bubble chart | × | ≈ | ✓ | ≈ | ✓ | ≈[b] | ✓ | × | ✓ |
| Radial coordinate visual. | × | × | × | ≈ | ✓ | ≈[b] | ✓ | ✓ | ✓ |
| Parallel coordinates | ≈[c] | × | ✓ | ≈[c] | ✓ | × | × | ✓ | ✓ |
| Heatmaps | × | × | ✓ | × | ✓ | × | × | ✓ | ✓ |
| Sammon mapping | × | × | × | ✓ | ≈ | ≈[b] | ✓ | ✓ | × |
| Neuroscale | × | × | × | × | ≈ | ≈[b] | ✓ | ✓ | × |
| Self-organizing maps | × | × | × | × | ≈ | ✓ | × | ✓ | × |
| Principal component analysis | × | × | × | × | ≈ | ≈[b] | ✓ | ✓ | × |
| Isomap | × | × | × | ≈ | ≈ | ≈[b] | ✓ | ✓ | × |
| Distance and distrib. charts | ≈ | × | × | × | ✓ | × | ✓ | ≈ | × |
| Interactive decision maps | × | ≈ | ✓ | ≈ | ✓ | ✓ | × | × | ✓ |
| Hyper-space diagonal count. | × | × | × | ≈ | ✓ | ✓ | ✓ | ≈ | × |
| Two-stage mapping | ≈ | × | × | × | × | ≈[b] | ✓ | ≈ | × |
| Level diagrams | × | ≈ | ✓ | × | ✓ | ≈[b] | ✓ | ✓ | ✓ |
| Hyper-radial visualization | × | ≈ | ✓ | × | ✓ | ≈[b] | ✓ | ✓ | ✓ |
| Pareto shells | ✓ | × | × | × | × | × | ✓ | ✓ | ✓ |
| Seriated heatmaps | × | × | × | × | ≈ | × | × | ✓ | × |
| Multi-dimensional scaling | × | × | × | × | × | ≈[b] | ✓ | ✓ | × |
| Prosections | ✓ | ✓ | ≈ | ✓ | ✓ | ✓ | ✓ | × | ≈ |

[a] Note that these properties cannot be fully preserved (we are therefore assessing only their partial preservation).
[b] Difficulties with very large sets (containing thousands of vectors).
[c] Possible when only a few vectors are visualized.

# Chapter 5

# Visualization with Prosections

As we have seen in the previous chapter, numerous ways to visualize 4D approximation sets exist. However, 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 sets that preserves the Pareto dominance relation, and the ability of handling multiple large approximation sets while being robust and computationally inexpensive. Moreover, despite all these new visualization possibilities, most researchers in the field of evolutionary multiobjective optimization still resort to parallel coordinates when a 4D (or higher) approximation set is to be shown. This was our motivation for developing a new visualization method that reduces one dimension of the approximation set using projection of a section and rotation (Tušar & Filipič, 2014b).

This chapter is devoted to the thorough description of visualization with prosections. First, Section 5.1 presents the method, while Section 5.2 shows its output for the BASes from Chapter 3. Next, the parameters of the method are discussed in Section 5.3 and some of its properties are formally proven in Section 5.4. An alternative way of visually exploring the objective space is discussed in Section 5.5. Finally, the chapter concludes with several usage examples in Section 5.6 and a summary in Section 5.7.

## 5.1 Prosections

A *prosection* is a projection of a section (the term was introduced by Furnas and Buja (1994)). Here, the section on the 2D plane $f_1 f_2$ with origin $\boldsymbol{a} = (a_1, a_2)$ is defined by the angle $\varphi$ and width $d$ (see (Tušar & Filipič, 2011) for two alternative section definitions). Each objective vector $\boldsymbol{z} = (z_1, z_2)$ within the section

$$|(z_1 - a_1)\sin\varphi - (z_2 - a_2)\cos\varphi| \leq d, \tag{5.1}$$

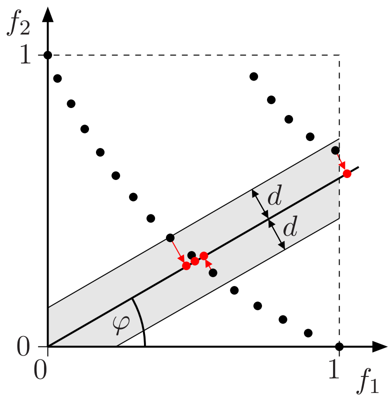
where $z_1 \geq a_1$ and $z_2 \geq a_2$, is orthogonally projected to the line crossing the origin $\mathbf{0} = (0, 0)$ and intersecting the $f_1$-axis at angle $\varphi$ using the mapping $p_{\varphi,d,\boldsymbol{a}}$,

$$p_{\varphi,d,\boldsymbol{a}} : (z_1, z_2) \mapsto (z'_1, z'_2), \tag{5.2}$$

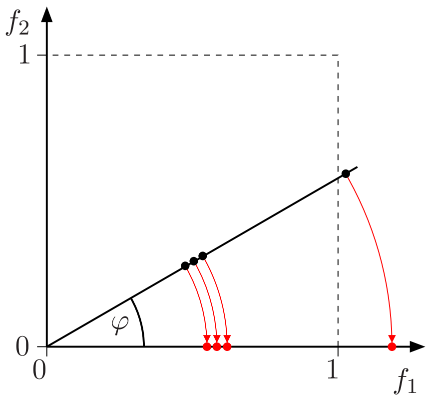
where

$$\begin{aligned} z'_1 &= \cos\varphi((z_1 - a_1)\cos\varphi + (z_2 - a_2)\sin\varphi), \\ z'_2 &= \sin\varphi((z_1 - a_1)\cos\varphi + (z_2 - a_2)\sin\varphi). \end{aligned} \tag{5.3}$$

All the objective vectors in the section are projected to the mentioned line, while all other objective vectors are ignored (see Figure 5.1a).




(a) Projection




(b) Rotation

Figure 5.1: Graphical presentation of the prosection with origin $\mathbf{0}$.

After this mapping, the line with projected vectors needs to be rotated so that this truly becomes a reduction in dimension (see Figure 5.1b):

$$r : (z'_1, z'_2) \mapsto \sqrt{(z'_1)^2 + (z'_2)^2}. \tag{5.4}$$

When composing these two functions, the transformation simplifies to:

$$s_{\varphi,d,\boldsymbol{a}}(z_1, z_2) = r(p_{\varphi,d,\boldsymbol{a}}(z_1, z_2)) = (z_1 - a_1)\cos\varphi + (z_2 - a_2)\sin\varphi \tag{5.5}$$

for all objective vectors in the section. The function $s_{\varphi,d,\boldsymbol{a}}$ performs dimension reduction from a 2D to a 1D space. Let us now show how this can be employed to reduce the $m$D space to $(m-1)$D.

When the number of objectives $m > 2$, other planes beside $f_1 f_2$ are possible. Therefore we denote with $ijk_1 \dots k_{m-2}$ a permutation of objective indices $1, \dots, m$, so that $k_1 < \dots < k_{m-2}$. The prosection is always performed on the $f_i f_j$ plane, and because of the previous condition all other objectives are kept in ascending order. Performing prosection on $m$D approximation sets consists of the following steps:

1. Select the origin $\boldsymbol{a} = (a_1, \dots, a_m)$ and prosection plane $f_i f_j$, where $i, j \in \{1, \dots, m\}$ and $i \neq j$.

2. Define the section by choosing the angle $\varphi$ and section width $d$. The section contains all vectors $\boldsymbol{z} = (z_1, \dots, z_m)$, $\boldsymbol{a} \preceq \boldsymbol{z}$, for which $|(z_i - a_i)\sin\varphi - (z_j - a_j)\cos\varphi| \leq d$.

3. All vectors within the section are projected using the following function:

$$(z_1, \dots, z_m) \mapsto ((z_i - a_i)\cos\varphi + (z_j - a_j)\sin\varphi, z_{k_1}, \dots, z_{k_{m-2}}) \tag{5.6}$$

4. All vectors outside the section are ignored.

The prosection affects only two objectives ($f_i$ and $f_j$), while all the others remain intact and in the same order as before prosection. The new "objective" that is formed in the prosection (denoted simply by $f_i f_j$) still needs to be minimized, i.e. lower values of $f_i f_j$ are preferred to higher ones.

A prosection of an $m$D approximation set with origin $\boldsymbol{a}$, prosection plane $f_i f_j$, angle $\varphi$ and section width $d$ will be denoted with

$$m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d) \tag{5.7}$$

from now on. Note that for any $\varphi \in [0°, 90°]$:

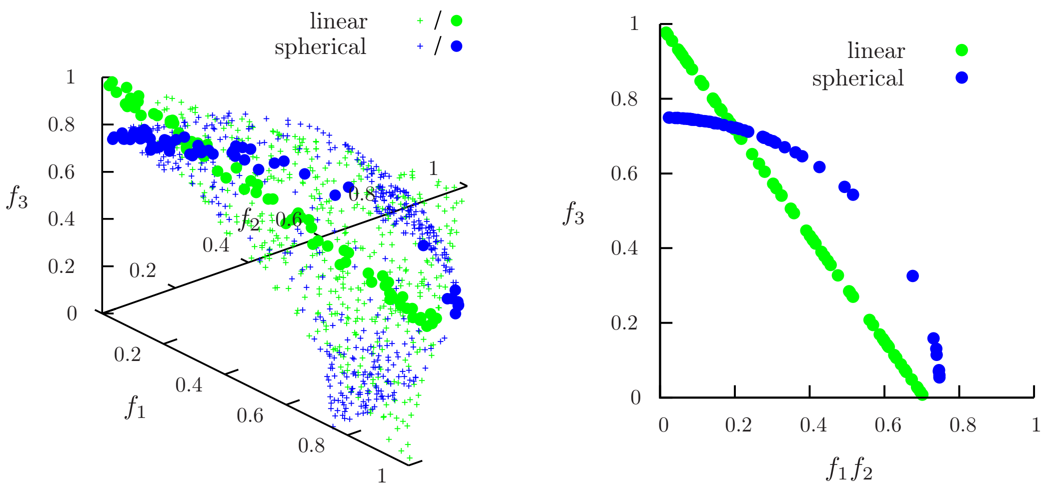
$$m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d) \equiv m\mathrm{D}(\boldsymbol{a}, f_j f_i, (90° - \varphi), d). \tag{5.8}$$

This means, for example, that the prosection on the plane $f_i f_j$ with angle 30° is equivalent to the prosection on the plane $f_j f_i$ with angle 60°. As a consequence, there is no need to explore both prosections on the plane $f_i f_j$ and on the plane $f_j f_i$.

While this transformation can be performed on an $m$D approximation set for an arbitrary $m \geq 2$, the resulting $(m-1)$D set can be easily visualized only if $m \leq 4$.

## 5.2 Visualization of BASes

Let us demonstrate how this method works when projecting 3D approximation sets to 2D on the example of the two 3D BASes from Figure 3.1b. Assume the section is defined by the angle $\varphi = 45°$ and the section width $d = 0.05$. This section cuts the plane $f_1 f_2$ in the middle. Figure 5.2a shows which of the objective vectors fall in the specified section, while the same objective vectors after prosection are presented in Figure 5.2b. Essentially, what the method does is slice through approximation set at angle $\varphi$ and project this slice so that dimension reduction is achieved. Prosections of the 3D BASes are very similar to the scatter plot of the 2D BASes (see Figure 3.1a). This is desirable since the 3D BASes are a generalization of the 2D ones.



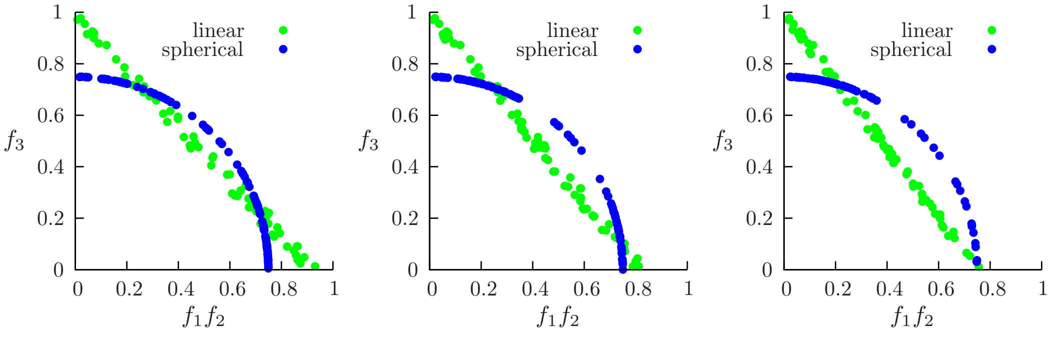

(a) Before prosection (vectors in the section are emphasized)

(b) After prosection $3\mathrm{D}(\mathbf{0}, f_1 f_2, 45°, 0.05)$

Figure 5.2: Prosection of the 3D BASes under angle $\varphi = 45°$.

In addition, Figure 5.3 presents the results of prosections under different angles $\varphi$. Angles over 45° are not shown as the BASes are nearly symmetric—therefore each prosection plot $3\mathrm{D}(\mathbf{0}, f_1 f_2, \varphi, 0.05)$ is very similar to the corresponding $3\mathrm{D}(\mathbf{0}, f_1 f_2, 90° - \varphi, 0.05)$ one. Depending on the angle, the prosections capture either one or two regions of the spherical BAS with a high density of vectors. We can see that the range, shape and distribution of vectors from the visualized part of the objective space are well preserved, while the preservation of the Pareto dominance relation will be discussed in more detail later.

Now, let us focus on the 4D case. Figure 5.4 shows the $4\mathrm{D}(\mathbf{0}, f_1 f_2, 45°, 0.25)$ visualization with prosection of the small 4D BASes. As the small BASes contain only 300 vectors




(a) 3D($\mathbf{0}, f_1 f_2, 5°, 0.05$) (b) 3D($\mathbf{0}, f_1 f_2, 15°, 0.05$) (c) 3D($\mathbf{0}, f_1 f_2, 25°, 0.05$)

(d) 3D($\mathbf{0}, f_1 f_2, 35°, 0.05$) (e) 3D($\mathbf{0}, f_1 f_2, 45°, 0.05$)

Figure 5.3: Prosections of the 3D BASes under different angles $\varphi$.

each, a larger section ($d = 0.25$) is chosen in order to show the preservation of the linear and spherical shape of the approximation sets. The plot clearly shows the differences in the distribution of vectors from both sets (uniform vs. nonuniform distribution). Similar visualizations are achieved also on the large 4D BASes under different angles $\varphi$ and with the section width $d$ set to 0.05 (see Figure 5.5). Depending on the angle, two or three dense regions of the spherical BAS are visualized. Note again that these prosections resemble very much the 3D BASes from Figure 3.1b , showing that they achieve an intuitive visualization of the high dimensional approximation sets.

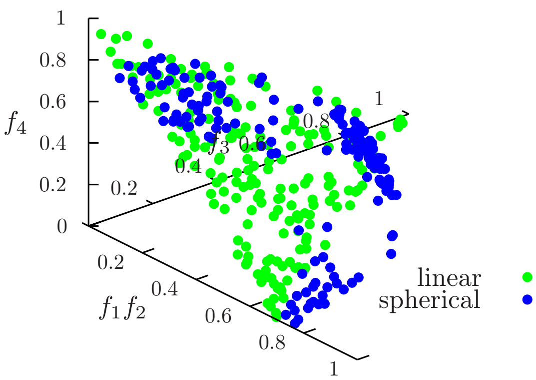
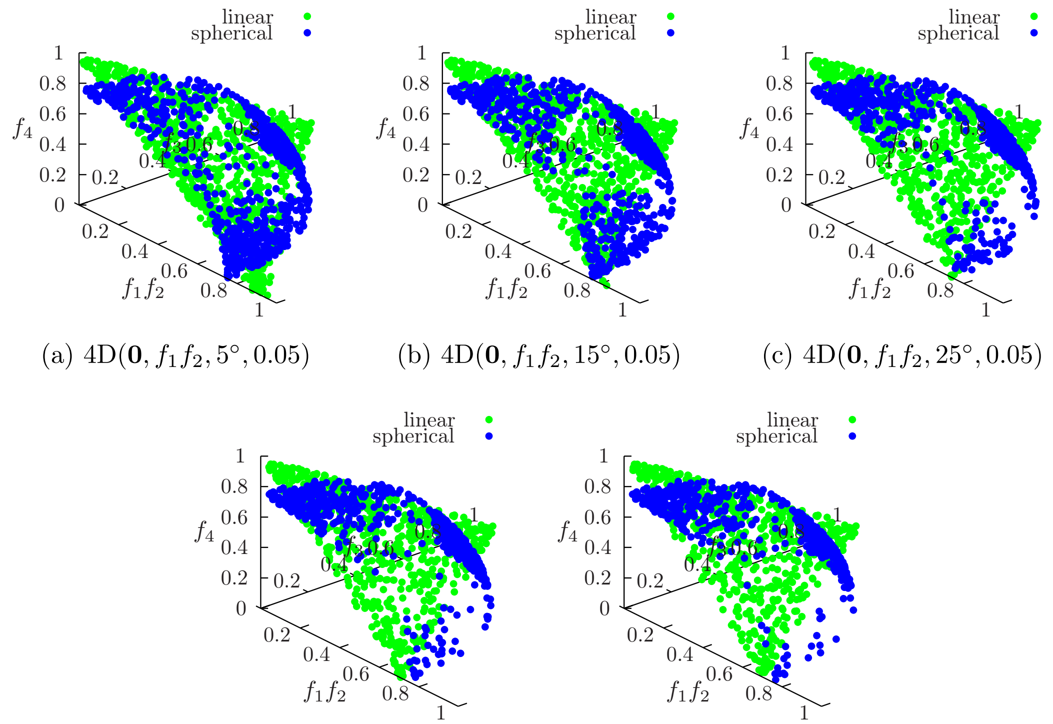



Figure 5.4: Prosection 4D($\mathbf{0}, f_1 f_2, 45°, 0.25$) of the small 4D BASes.




(a) 4D($\mathbf{0}, f_1 f_2, 5°, 0.05$) (b) 4D($\mathbf{0}, f_1 f_2, 15°, 0.05$) (c) 4D($\mathbf{0}, f_1 f_2, 25°, 0.05$)

(d) 4D($\mathbf{0}, f_1 f_2, 35°, 0.05$) (e) 4D($\mathbf{0}, f_1 f_2, 45°, 0.05$)

Figure 5.5: Prosections of the 4D BASes under different angles $\varphi$.

## 5.3 Parameters

The visualization with prosections depends on four parameters: plot origin $\boldsymbol{a}$, prosection plane $f_i f_j$, angle $\varphi$ and section width $d$.

### 5.3.1 Plot Origin and Range of Objectives

For a reasonable result, the plot origin should be set to a vector that dominates all objective vectors to be visualized. The ideal objective vector is thus a sensible (but not obligatory) choice for the plot origin. If the origin is chosen to be far better than the ideal objective vector, prosections under extreme angles (near 0° and 90°) and narrow sections might turn out empty.

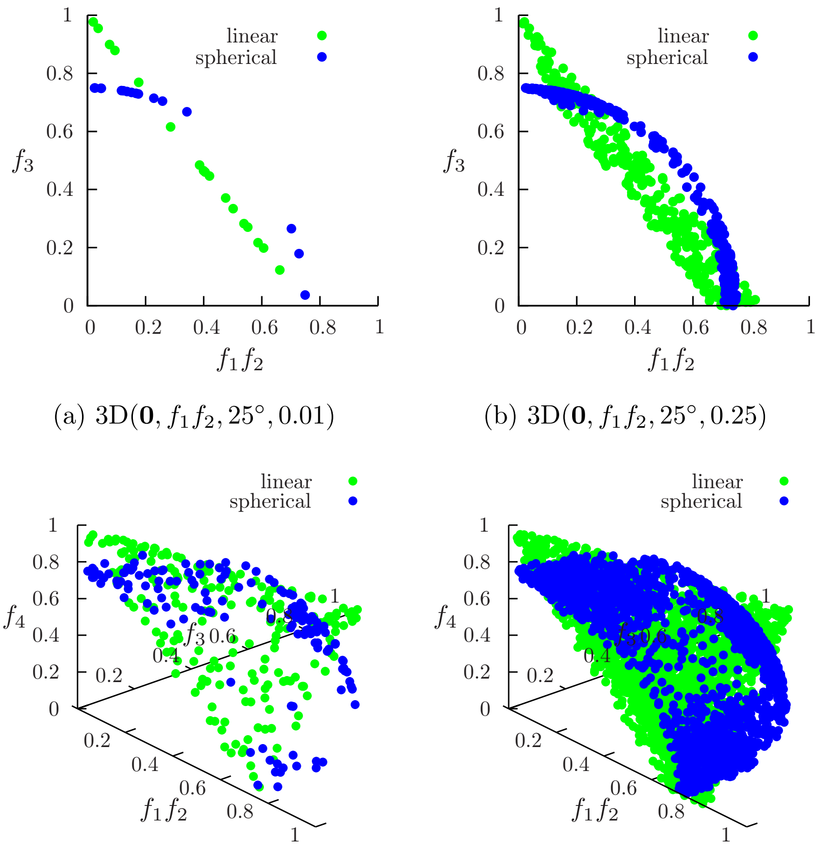
While the prosection is well defined for any range of objectives, objectives that have ranges of different magnitude affect the "meaning" of the angle $\varphi$. For example, the angle $\varphi = 45°$ does not cut the rectangle $[a_i, b_i] \times [a_j, b_j]$ exactly in half if $(b_i - a_i) \neq (b_j - a_j)$. Also, in extreme cases of disproportionate objectives, the size of the section depends heavily on the chosen angle. Therefore, in cases with a large difference between the ranges of objectives, it is best to normalize the objectives prior to visualization.

### 5.3.2 Prosection Plane

In the examples shown so far, we have always performed prosection on a single prosection plane—$f_1 f_2$ (this was not particularly problematic considering the symmetric nature of our BASes). However, in the general case more prosection planes need to be explored to gain a complete "mental picture" of the approximation sets. This can be done simultaneously using a *prosection matrix* as in (Tweedie et al., 1996), where each prosection plane results in one plot. As the prosections are symmetric, half of the matrix suffices (as with the scatter plot matrix). Two examples of prosection matrices are provided in Section 5.6. Note also that a chosen prosection plot (or even the whole matrix) can be animated by showing how prosections transform when the angle $\varphi$ changes. This can further help to construct a mental picture about the trade-off among objectives when the approximation sets are not symmetric.

### 5.3.3 Section Definition

The section is defined with the angle $\varphi$ and width $d$. The choice of these two parameters influences greatly the resulting visualization. The angle determines which part of the approximation set is visualized, while the section width regulates the amount of vectors that will be included in the visualization—larger sections produce more crowded plots. See for example the influence of the section width in Figure 5.6. The section width should be chosen so that it includes enough vectors to visualize the shape of the approximation set and the distribution of vectors (not the case if $d = 0.01$), while at the same time not overcrowding it with too many vectors ($d = 0.25$ is too wide). For our BASes this means choosing the section width near 0.05 (see the Figures 5.3c and 5.5c).




(a) 3D$(\mathbf{0}, f_1 f_2, 25°, 0.01)$ (b) 3D$(\mathbf{0}, f_1 f_2, 25°, 0.25)$

(c) 4D$(\mathbf{0}, f_1 f_2, 25°, 0.01)$ (d) 4D$(\mathbf{0}, f_1 f_2, 25°, 0.25)$

Figure 5.6: Section width effect. The plots show prosections of the 3D and the 4D BASes using different values for section width $d$ ($d = 0.01$ for plots (a) and (c), and $d = 0.25$ for plots (b) and (d)).

Figure 5.6b demonstrates that approximation sets after prosection with a wide section can be indistinct. This depends not only on the width of the section, but also on the chosen angle $\varphi$ and the *shape of the approximation set*. We will explain the reasons for this shortly.

## 5.4 Properties

The basic difference between the prosections proposed here and the orthogonal prosections mentioned in Section 4.3 is in the angle $\varphi$, which was either $0°$ or $90°$ in previous work (hence we called those prosections orthogonal). The fact that an angle $\varphi$ different from $0°$ and $90°$ is used leads to two important properties of this method, formulated in Theorems 5.1 and 5.2 (their proofs are in Appendix A).

**Theorem 5.1.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \geq 2$ and $\varphi \in (0°, 90°)$. Then for any two objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \dots, z_m^{\mathrm{A}})$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \dots, z_m^{\mathrm{B}})$ inside the section the following holds:

$$\boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}} \Rightarrow \sigma(\boldsymbol{z}^{\mathrm{A}}) \prec \sigma(\boldsymbol{z}^{\mathrm{B}}). \tag{5.9}$$

This means that if one vector dominates the other, the dominance relation is retained after prosection. While it is beneficial that a visualization method is capable of correctly showing the dominance relations among vectors, the other way around (being able to infer the dominance relations from the visualization) is even more important for the correct understanding of the visualized approximation sets.

As shown by Köppen and Yoshida (2007), no Pareto-dominance preserving mapping exists[1]. Nevertheless, for prosections we can prove that if one projected vector dominates another projected vector and the two are apart enough, the first vector indeed dominates the second one.

**Theorem 5.2.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \geq 2$ and $\varphi \in (0°, 90°)$. Then for any two objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \dots, z_m^{\mathrm{A}})$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \dots, z_m^{\mathrm{B}})$ inside the section the following holds:

$$\sigma(\boldsymbol{z}^{\mathrm{A}}) \prec \sigma(\boldsymbol{z}^{\mathrm{B}}) \wedge s_{\varphi,d,\boldsymbol{a}}(z_i^{\mathrm{B}}, z_j^{\mathrm{B}}) - s_{\varphi,d,\boldsymbol{a}}(z_i^{\mathrm{A}}, z_j^{\mathrm{A}}) \geq 2d \max\left\{\tan\varphi, \tan^{-1}\varphi\right\} \Rightarrow \boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}}. \tag{5.10}$$

"Apart enough" is thus any distance in the new objective $f_i f_j$ that is greater than $2d \max\left\{\tan\varphi, \tan^{-1}\varphi\right\}$. Unfortunately, $\max\left\{\tan\varphi, \tan^{-1}\varphi\right\}$ strongly depends on the chosen angle $\varphi$. If $\varphi = 45°$, then $\tan\varphi = \tan^{-1}\varphi = 1$, which is the smallest possible value. For all other values of $\varphi$, the value of $\max\left\{\tan\varphi, \tan^{-1}\varphi\right\}$ is higher, getting unpractical high values in the proximity of $\varphi = 0°$ and $\varphi = 90°$. Figure 5.7 shows how the value of $2d \max\left\{\tan\varphi, \tan^{-1}\varphi\right\}$ depends on the values of $\varphi$ and $d$.

Theorem 5.2 tells us that we cannot completely trust the visualized dominance relations. While some vectors may appear (non)dominated in the prosection plot, this might not be the case in the original objective space. Only those vectors that are dominated according to Theorem 5.2 are truly dominated, while for the rest we cannot be certain. Figure 5.8 shows on the example of our BASes which vectors from the spherical BASes are dominated by the vectors from the linear BASes according to Theorem 5.2.

[1] It is easy to see why the right-to-left direction from Definition 2.12 does not hold for prosections: if the objective vectors are incomparable, after prosection the one projected vector might dominate the other projected vector.

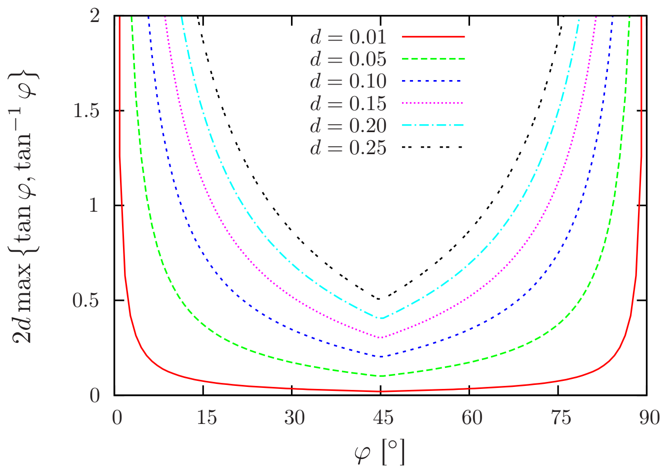
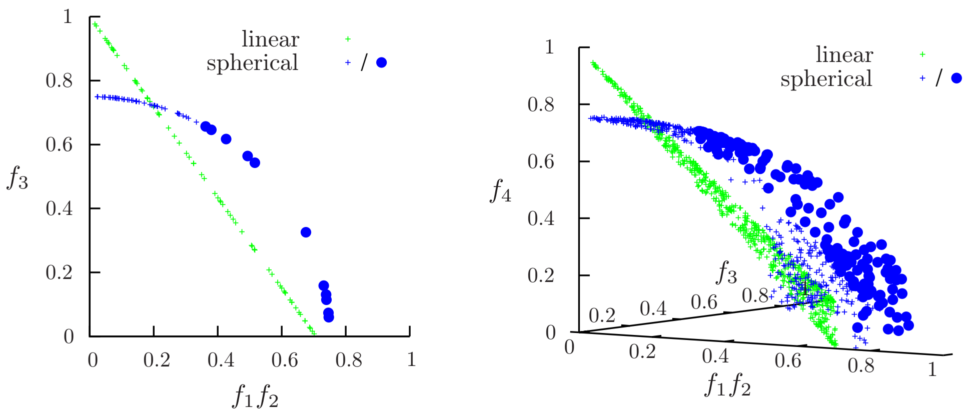



Figure 5.7: Dependence of $2d \max \left\{\tan\varphi, \tan^{-1}\varphi\right\}$ on angle $\varphi$ and section width $d$.



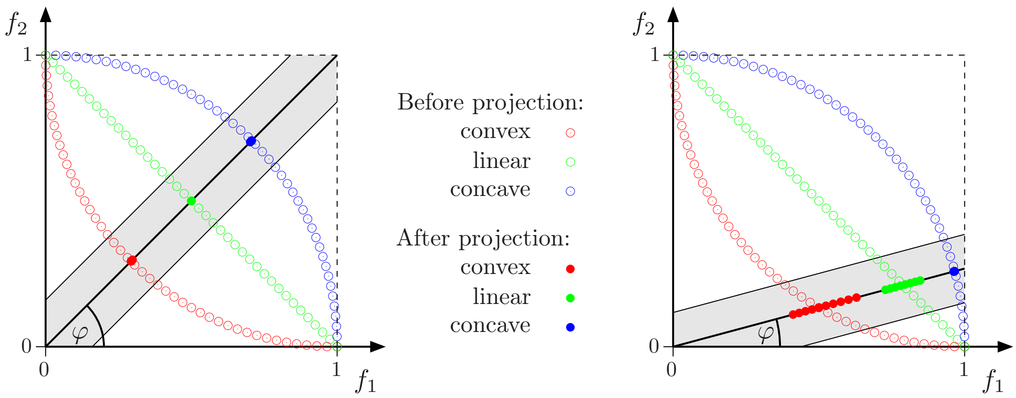

(a) $3\text{D}(\mathbf{0}, f_1 f_2, 45°, 0.05)$ (b) $4\text{D}(\mathbf{0}, f_1 f_2, 45°, 0.05)$

Figure 5.8: Visualization of preserved dominance relations. Vectors from the spherical BASes that are dominated by the vectors from the linear BASes according to Theorem 5.2 are emphasized (drawn using dots instead of small crosses).

Additionally, Theorem 5.2 explains the indistinctness mentioned before, since the value $2d \max\{\tan\varphi, \tan^{-1}\varphi\}$ is exactly the maximum possible width of indistinctness that an approximation set can achieve in the new objective. The actual indistinctness of an approximation set after prosection (which is often much smaller than the maximum one) depends also on its shape—see the example of three 2D approximation sets of different shape (convex, linear and concave) in Figure 5.9. When $\varphi = 45°$, all approximation sets after prosection are distinct, since they are all almost perpendicular to the section. On the other hand, after prosection with $\varphi = 15°$, the concave approximation is still distinct, but this is not the case for the linear (some indistinctness) and the convex (a lot of indistinctness) ones.

Note that the assertions from Theorems 5.1 and 5.2 are not true if $\varphi$ is equal to 0° or 90°, which means that simple orthogonal prosections do not share these useful properties.

Next, let us explore the interpretation of the new objective $f_i f_j$. Because of the projection, much of the information on $f_i$ and $f_j$ is lost, but not all. Assume the $m\text{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$




(a) $\varphi = 45°$ (b) $\varphi = 15°$

Figure 5.9: Effect of the prosection angle and approximation set shape on the actual indistinctness. The plots show 2D convex, linear and concave approximation sets before and after projection (without applying rotation) using different values for angle $\varphi$.

prosection is performed and we are interested in the original values in objectives $f_i$ and $f_j$ of the projected vector with value $A$ in objective $f_i f_j$. Then we know that the original values of $f_i$ and $f_j$ lie on the line segment $\overline{A'A''}$, where

$$\begin{aligned} A' &= (a_i + A\cos\varphi - d\sin\varphi, a_j + A\sin\varphi + d\cos\varphi), \\ A'' &= (a_i + A\cos\varphi + d\sin\varphi, a_j + A\sin\varphi - d\cos\varphi). \end{aligned} \tag{5.11}$$

For example, if we are interested in the vector $A$ with value 0.5 in objective $f_1 f_2$ of the prosection $4\mathrm{D}(\mathbf{0}, f_1 f_2, 45°, 0.05)$, we know that the original values in $f_i$ and $f_j$ lie on the line segment between vectors $(0.318, 0.389)$ and $(0.389, 0.318)$.

Finally, a note on the transformation of distances. As a projection from an $m$D space to a $(m-1)$D space is performed, the distances among arbitrary vectors cannot be preserved. However, it is trivial to show that the distance between two vectors after prosection is never greater than the distance between the original vectors. The distance is preserved when the line segment bounded by the two original vectors is parallel to the line intersecting the plane $f_i f_j$ at the angle $\varphi$.

**Theorem 5.3.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \geq 2$. Then for any two objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}})$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}})$ inside the section the following holds:

$$\left\|\sigma(\boldsymbol{z}^{\mathrm{A}}) - \sigma(\boldsymbol{z}^{\mathrm{B}})\right\| \leq \left\|\boldsymbol{z}^{\mathrm{A}} - \boldsymbol{z}^{\mathrm{B}}\right\|. \tag{5.12}$$

The equality holds iff

$$\frac{z_j^{\mathrm{A}} - z_j^{\mathrm{B}}}{z_i^{\mathrm{A}} - z_i^{\mathrm{B}}} = \tan\varphi. \tag{5.13}$$

More importantly, we are able to show that prosections preserve the relative closeness to the reference vector for some vectors. This is especially useful when (one or more) reference vectors are given and we wish to visualize them together with the approximation set.

**Theorem 5.4.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \geq 2$. Let $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}})$, $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}})$ and $\boldsymbol{r} = (r_1, \ldots, r_m)$ be three vectors inside the section and let us assume that $\left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|<\left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|$. Then

$$\left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|^{2}-\left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|^{2}>4d^{2} \Rightarrow\left\|\sigma(\boldsymbol{z}^{\mathrm{A}})-\sigma(\boldsymbol{r})\right\|<\left\|\sigma(\boldsymbol{z}^{\mathrm{B}})-\sigma(\boldsymbol{r})\right\|. \tag{5.14}$$

On the other hand,

$$\left\|\sigma(\boldsymbol{z}^{\mathrm{B}})-\sigma(\boldsymbol{r})\right\|^{2}-\left\|\sigma(\boldsymbol{z}^{\mathrm{A}})-\sigma(\boldsymbol{r})\right\|^{2}>4d^{2} \Rightarrow\left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|<\left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|. \tag{5.15}$$

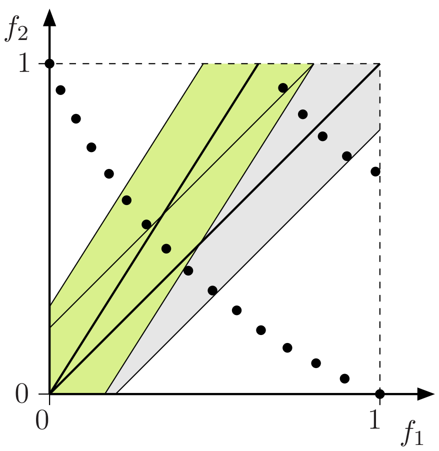
According to Theorem 5.4, if vectors $\boldsymbol{z}^{\mathrm{A}}$ and $\boldsymbol{z}^{\mathrm{B}}$, where $\boldsymbol{z}^{\mathrm{A}}$ is closer to the reference vector $\boldsymbol{r}$ than $\boldsymbol{z}^{\mathrm{B}}$, are "apart enough" from the perspective of the reference vector ($\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\|^2-\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\|^2>4d^2$), the vector $\boldsymbol{z}^{\mathrm{A}}$ remains closest to the reference vector also after prosection. And vice versa, from the distances between the vectors and the reference vector after prosection we can infer on their closeness in the original space. This means that prosections are able to truthfully visualize the closeness to the reference vector for some vectors.

Note that the properties described in Theorems 5.3 and 5.4 are independent from the chosen angle $\varphi$. The proofs of Theorems 5.3 and 5.4 can also be found in Appendix A.

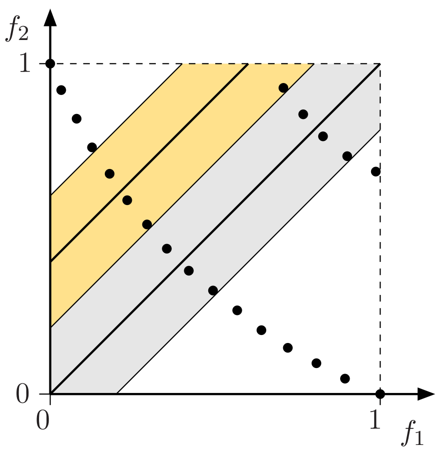
## 5.5 Alternative Objective Space Exploration

In order to visually explore different parts of the objective space, the section needs to be rotated around its origin. As we have shown in Figure 5.9, this produces indistinctness when the angle $\varphi$ is close to 0° or 90°. To overcome this issue, an alternative way of exploring the objective space can be tried: we can *slide* the section at a fixed angle instead of rotating it. Figure 5.10 shows the differences between the two variants.

When the section is slid, the angle $\varphi$ remains constant and only the origin $\boldsymbol{a}$ changes. This means that using the angle $\varphi=45°$, the indistinctness is minimized as its width is always equal to $2d$. Moreover, as sliding can always be set so that there is no overlapping of sections in the objective space, less views are needed to explore the whole objective space (using the same section width $d$) than when using rotating sections. However, section sliding has its shortcomings. First, the origin is not included in most of the slices, meaning that we lose view of what is often the ideal objective vector we wish to come near to.



(a) Rotating section



(b) Sliding section

Figure 5.10: Two variants of exploring the objective space.

Furthermore, when sliding away from the origin, the sections get smaller, covering less and less of the objective space.

The listed advantages and disadvantages are very subjective (which is more important—small indistinctness or a view of the origin?), which in turn makes the decision of which variant to use subjective, too. While we have tried to define some measures of quality for the two variants, this proved to be very difficult to do and gave no conclusive result to which variant is better. Therefore, we have (subjectively) decided upon using rotating sections to explore the objective space as they constantly provide the view of the origin—the most important reference vector available.

## 5.6 Usage Examples

Here we show how prosections can be used for visualizing solutions to problems with redundant objectives, various shapes of Pareto fronts (different from the spherical and linear used until now), the progress of a MOEA and the results of different MOEAs.

### 5.6.1 Visualization in Case of Redundant Objectives

Since our approach slices through approximation sets, we wish to explore how this affects visualization of solutions to problems with *redundant objectives.* An objective is redundant if its elimination does not affect the Pareto front of the given problem. The DTLZ5$(I, M)$ problem family (Deb & Saxena, 2006; Saxena, Duro, Tiwari, Deb, & Zhang, 2013) presents problems with redundant objectives, where $I$ denotes the dimension of the Pareto front, and $M$ is equal to the number of objectives. Here we use two 4D minimization problems, DTLZ5$(3, 4)$ and DTLZ5$(2, 4)$, which have one and two redundant objectives, respectively, and are defined with:

$$\begin{aligned}
f_1(\boldsymbol{x}) &= (1 + g(\boldsymbol{x}))\cos(\theta_1(\boldsymbol{x}))\cos(\theta_2(\boldsymbol{x}))\cos(\theta_3(\boldsymbol{x})) \\
f_2(\boldsymbol{x}) &= (1 + g(\boldsymbol{x}))\cos(\theta_1(\boldsymbol{x}))\cos(\theta_2(\boldsymbol{x}))\sin(\theta_3(\boldsymbol{x})) \\
f_3(\boldsymbol{x}) &= (1 + g(\boldsymbol{x}))\cos(\theta_1(\boldsymbol{x}))\sin(\theta_2(\boldsymbol{x})) \\
f_4(\boldsymbol{x}) &= (1 + g(\boldsymbol{x}))\sin(\theta_1(\boldsymbol{x})) \\
g(\boldsymbol{x}) &= \sum_{i=4}^{n}(x_i - 0.5)^2 \\
\theta_i(\boldsymbol{x}) &= \begin{cases} \frac{\pi}{2}x_i & \text{if } i = 1, \ldots, I-1 \\ \frac{\pi}{4(1+g(\boldsymbol{x}))}(1 + 2g(\boldsymbol{x})x_i) & \text{if } i = I, \ldots, 3 \end{cases} \\
\boldsymbol{x} &\in [0, 1]^n,
\end{aligned} \tag{5.16}$$

where $n$ is the number of decision variables and additional constraints apply.

The DTLZ5$(3, 4)$ problem has two constraints:

$$\begin{aligned}
f_4(\boldsymbol{x})^2 + f_3(\boldsymbol{x})^2 + 2f_1(\boldsymbol{x})^2 &\geq 1 \\
f_4(\boldsymbol{x})^2 + f_3(\boldsymbol{x})^2 + 2f_2(\boldsymbol{x})^2 &\geq 1,
\end{aligned} \tag{5.17}$$

while the DTLZ5$(2, 4)$ problem has three:

$$\begin{aligned}
f_4(\boldsymbol{x})^2 + 4f_1(\boldsymbol{x})^2 &\geq 1 \\
f_4(\boldsymbol{x})^2 + 4f_2(\boldsymbol{x})^2 &\geq 1 \\
f_4(\boldsymbol{x})^2 + 2f_3(\boldsymbol{x})^2 &\geq 1.
\end{aligned} \tag{5.18}$$

In the DTLZ5(3, 4) problem, the Pareto front is characterized by $2f_1^2 + f_3^2 + f_4^2 = 1$ and $f_1 = f_2$, which means that either of the first two objectives is redundant. In the DTLZ5(2, 4) problem, vectors on the Pareto front comply to $4f_1^2 + f_4^2 = 1$ and $f_1 = f_2 = \frac{\sqrt{2}}{2} f_3$, meaning that two among the first three objectives are redundant. This implies that the Pareto front is a surface in the first case and a curve in the second one.

We approximate the two Pareto fronts with two approximation sets, each consisting of 3000 vectors, and visualize these sets with a prosection matrix, which enables us to view the prosections on all prosection planes simultaneously. Figure 5.11 shows the prosection matrix using angle $\varphi = 45°$ and width $d = 0.05$ for all prosection planes. Because only slices of approximation sets are visualized, we would expect the plots to show only a small part of each approximation set at a time. They mostly do—with the exception of the first one on the prosection plane $f_1 f_2$, which coincidentally (because for all vectors $f_1 = f_2$, i.e. all vectors lie on the hyperspace that intersects $f_1 f_2$ under the angle $\varphi = 45°$) shows the whole approximation sets.

It is interesting to inquire whether the existence of redundant objectives could be inferred solely from visualization with prosections. In problems with two redundant objectives, this should be possible. If the Pareto front is a curve, no prosection will visualize it as a surface. The other case, where the problem has only one redundant objective, might not be so straightforward. As we have seen from Figure 5.11, some specific views might visualize the whole surface. However, if for a chosen prosection plane, the approximation set is visualized as a strip of a surface regardless of the angle, we could speculate on the existence of a single redundant objective.

### 5.6.2 Visualizing Mixed-Shaped Pareto Fronts

We wish to show how prosections visualize approximation sets with a shape different from the spherical and linear. The WFG test problem toolkit can be used to create scalable multiobjective test problems with different characteristics (Huband et al., 2006).

In this example we use the 4D WFG1 test problem which has an interesting mixed convex and concave shape of the Pareto front. The front is sampled using 3000 vectors. The objectives of this problem have different ranges—for the vectors on the Pareto front the following holds: $f_i \in [0, 2i], i = 1, \ldots, 4$. Therefore, we normalize all vectors in the approximation set to lie in $[0, 1]^4$ prior to visualization. The normalized approximation set is visualized with the prosection matrix in Figure 5.12.

Depending on the prosection plane, the visualization is capable of showing the mixed convex and concave shape of the Pareto front (when $f_4$ is not included in the prosection plane—the left-hand side and central plots) or not (when $f_4$ is included in the prosection plane—the right-hand side plots). This is reasonable, since in the first case we "slice through the waves", while in the second one we "slice along them".

This example shows that it is indeed important to visualize more than just a single prosection to gain a full understanding of a non-symmetric approximation set. Alternatively, additional information on the approximation set can be gained through animation of a chosen prosection plot by changing the angle $\varphi$. For example, when the prosection $4\mathrm{D}(\mathbf{0}, f_3 f_4, \varphi, 0.05)$ is animated by changing $\varphi$ from 0° to 90° using the step 5°, we can see that the approximation set "oscillates" (repeatedly comes closer to the plot origin and then draws away from it). This indicates that the approximation set is "wavy", which is a property of the set that cannot be otherwise easily seen using only the prosection plane $f_3 f_4$.

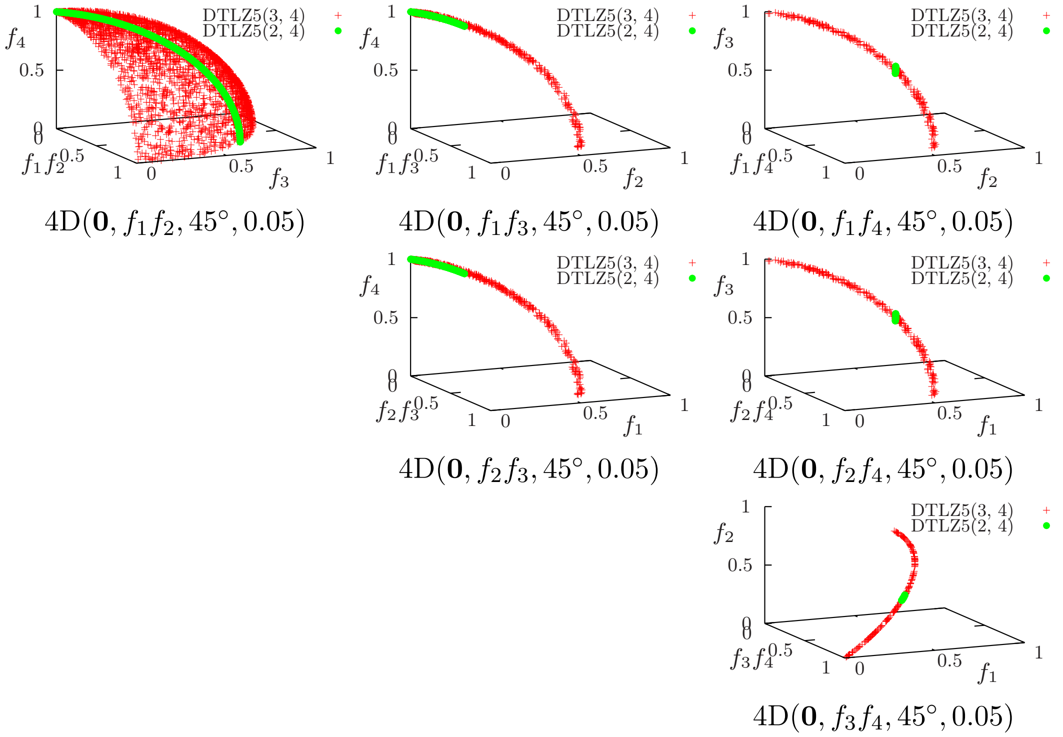



Figure 5.11: Prosection matrix for $DTLZ5(3,4)$ and $DTLZ5(2,4)$ test problems with redundant objectives.

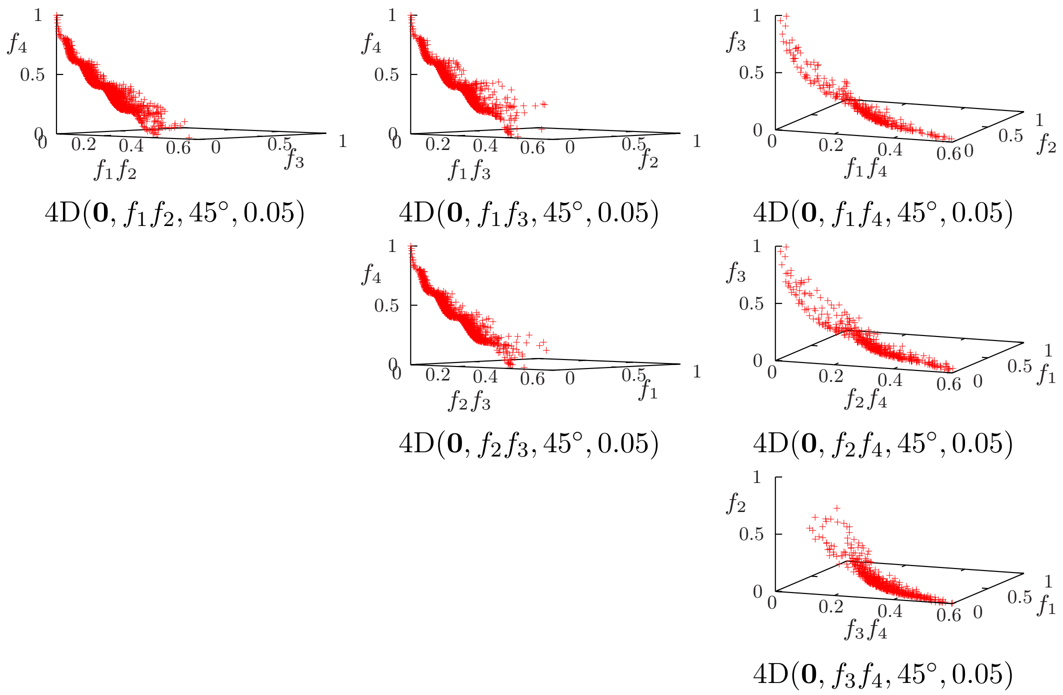



Figure 5.12: Prosection matrix for the WFG1 mixed-shaped test problem.

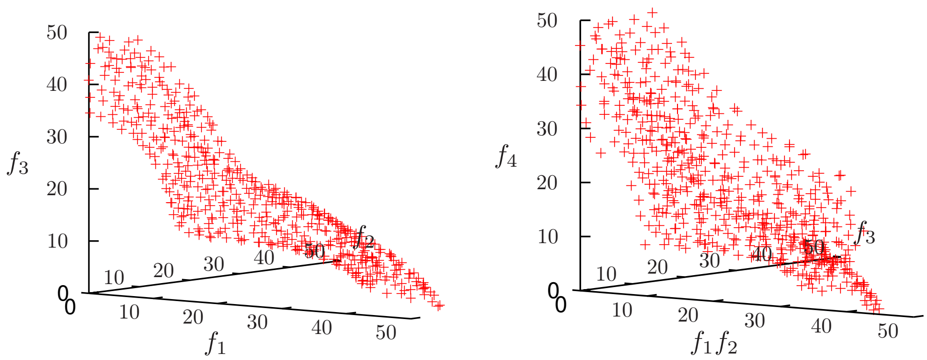
### 5.6.3 Visualizing Pareto Fronts with Knees

*Knees* are regions on the Pareto front where a small improvement in one objective leads to a large deterioration in at least one other objective. Knees are especially important for decision-making purposes as they are usually preferred to other parts of the Pareto front. It is therefore important to be able to show them when visualizing an approximation set with knees.

The first problems with knees were defined in (Branke, Deb, Dierolf, & Oswald, 2004), where the DEB2DK and DEB3DK test problems have two and three objectives, respectively. For example, the DEB3DK test problem assuming minimization in all objectives is defined as follows:

$$
\begin{aligned}
f_1(\boldsymbol{x}) &= g(\boldsymbol{x})r(\boldsymbol{x}) \sin\left(\frac{\pi}{2}x_1\right) \sin\left(\frac{\pi}{2}x_2\right) \\
f_2(\boldsymbol{x}) &= g(\boldsymbol{x})r(\boldsymbol{x}) \sin\left(\frac{\pi}{2}x_1\right) \cos\left(\frac{\pi}{2}x_2\right) \\
f_3(\boldsymbol{x}) &= g(\boldsymbol{x})r(\boldsymbol{x}) \cos\left(\frac{\pi}{2}x_1\right) \\
g(\boldsymbol{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i \\
r(\boldsymbol{x}) &= \frac{r_1(x_1) + r_2(x_2)}{2} \\
r_i(x_i) &= 5 + 10(x_i - 0.5)^2 + \frac{2\cos(2K\pi x_i)}{K} \\
\boldsymbol{x} &\in [0,1]^n.
\end{aligned}
\tag{5.19}
$$

Here, $n$ is the number of dimensions of the decision space and $K$ is a parameter that together with the number of objectives $m$ determines the number of knees in the Pareto front $K^{m-1}$. We show the approximation set consisting of 500 vectors from the Pareto front of the DEB3DK problem with $K = 1$ in Figure 5.13a. Because $K = 1$, this Pareto front has only one knee, which is clearly visible from the plot of the approximation set.




(a) 3D (b) 4D($\mathbf{0}, f_1 f_2, 45°, 2.5$)

Figure 5.13: Approximation sets of the DEB3DK and DEB4DK test problems.

Since DEB2DK and DEB3DK are based on the DTLZ problems, they are scalable to any number of objectives. However, they have not been scaled to more than 3D by their authors. Here we introduce the 4D version of this problem called DEB4DK (Tušar &

Filipič, 2012):

$$\begin{aligned}
f_1(\boldsymbol{x}) &= g(\boldsymbol{x})r(\boldsymbol{x})\sin\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right)\sin\left(\frac{\pi}{2}x_3\right)\\
f_2(\boldsymbol{x}) &= g(\boldsymbol{x})r(\boldsymbol{x})\sin\left(\frac{\pi}{2}x_1\right)\sin\left(\frac{\pi}{2}x_2\right)\cos\left(\frac{\pi}{2}x_3\right)\\
f_3(\boldsymbol{x}) &= g(\boldsymbol{x})r(\boldsymbol{x})\sin\left(\frac{\pi}{2}x_1\right)\cos\left(\frac{\pi}{2}x_2\right)\\
f_4(\boldsymbol{x}) &= g(\boldsymbol{x})r(\boldsymbol{x})\cos\left(\frac{\pi}{2}x_1\right)\\
g(\boldsymbol{x}) &= 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i\\
r(\boldsymbol{x}) &= \frac{r_1(x_1) + r_2(x_2) + r_3(x_3)}{3}\\
r_i(x_i) &= 5 + 10(x_i - 0.5)^2 + \frac{3\cos(2K\pi x_i)}{K}\\
\boldsymbol{x} &\in [0,1]^n.
\end{aligned} \tag{5.20}$$

Again, we use the DEB4DK problem with $K = 1$, which means that the Pareto front of this problem has only one knee, too. The Pareto front is again sampled with 3000 vectors, but normalization is not needed since all objectives have similar ranges[2]. Note, however, that larger objective values require a larger section width than usual (we chose $d = 0.05 \cdot 50 = 2.5$). Figure 5.13b presents the visualization of this approximation set using prosections. We show only the visualization on one prosection plane as the others produce very similar results. We can see that under the angle 45° the knee is nicely visualized.

In summary, the WFG1 and DEB4DK test problems have shown the potential of prosections for visualizing the shape of Pareto fronts.

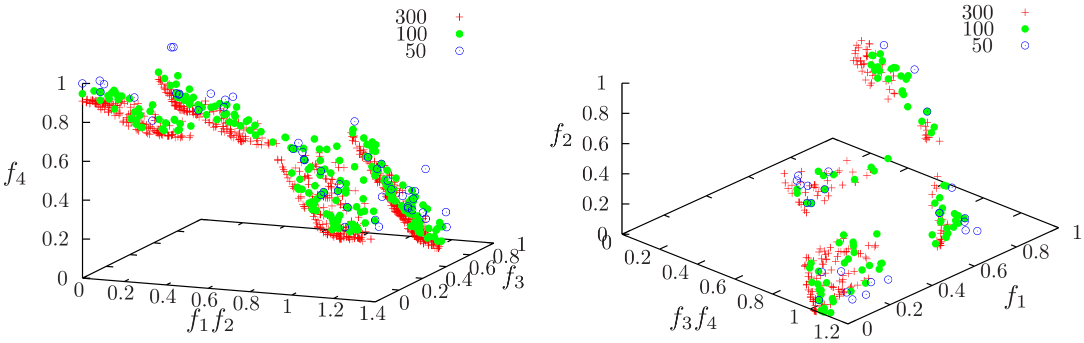
### 5.6.4 Visualizing the Progress of a MOEA

So far we have used only "artificial" approximation sets that were not achieved as a result of a MOEA. Therefore, in this usage example we wish to show how prosections can be used to visualize the progress of a true MOEA. To this end we use the Differential Evolution for Multiobjective Optimization (DEMO) algorithm (Robič & Filipič, 2005) on the DTLZ7 benchmark optimization problem with four objectives to be minimized (Deb et al., 2005). The Pareto front of this problem has eight disconnected regions and disproportionate objective values ($f_1, f_2, f_3 \in [0, 1]$ while $f_4 \in [2.9, 8]$), therefore normalization is required for the fourth objective.

The DEMO algorithm with the population of 100 vectors was run on this problem. To visualize the progress of DEMO, we plot the approximation sets achieved by the algorithm after 50, 100 and 300 generations, which have 329, 1022 and 3616 vectors, respectively (see Figure 5.14).

Again, we show only the visualizations on two prosection planes as the objectives $f_1, f_2$ and $f_3$ are symmetric and produce very similar plots. The first prosection (using $f_1 f_2$, see Figure 5.14a) shows four different regions. It is easy to see that with increasing generations the algorithm was able to converge better. Note also that the increasing number of vectors in the approximation sets is properly visualized. While the second prosection (using $f_3 f_4$, see Figure 5.14b) exhibits similar characteristics, it is interestingly able to show five regions simultaneously.

[2]The ranges of objectives of the DEB3DK and DEB4DK problems in this thesis differ from the ones presented by Branke et al. (2004). This might be due to an unwanted integer division in the original implementation of these two problems (more specifically, in the calculation of the $g(\boldsymbol{x})$ function).




(a) 4D$(\mathbf{0}, f_1 f_2, 45°, 0.05)$ (b) 4D$(\mathbf{0}, f_3 f_4, 45°, 0.05)$

Figure 5.14: Prosections visualizing the progress of a MOEA. The plots show the approximation sets at 50, 100 and 300 generations of the DEMO algorithm on the DTLZ7 problem.

### 5.6.5 Visualizing the Results of Different MOEAs

This last usage example shows how the results of different MOEAs can be visually compared with prosections and how this can provide additional insight into the behavior of an algorithm. We visualize the results of four well-known MOEAs on a polygon-based test problem from (Ishibuchi, Akedo, & Nojima, 2011).

The test problem with trapezoids is defined as follows (see also Figure 5.15):

$$\begin{aligned} f_1(\boldsymbol{x}) &= \min\{\|\boldsymbol{x} - \boldsymbol{a}^1\|, \|\boldsymbol{x} - \boldsymbol{a}^2\|\} \\ f_2(\boldsymbol{x}) &= \min\{\|\boldsymbol{x} - \boldsymbol{b}^1\|, \|\boldsymbol{x} - \boldsymbol{b}^2\|\} \\ f_3(\boldsymbol{x}) &= \min\{\|\boldsymbol{x} - \boldsymbol{c}^1\|, \|\boldsymbol{x} - \boldsymbol{c}^2\|\} \\ f_4(\boldsymbol{x}) &= \min\{\|\boldsymbol{x} - \boldsymbol{d}^1\|, \|\boldsymbol{x} - \boldsymbol{d}^2\|\} \\ \boldsymbol{x} &\in [0, 100]^2, \end{aligned} \tag{5.21}$$

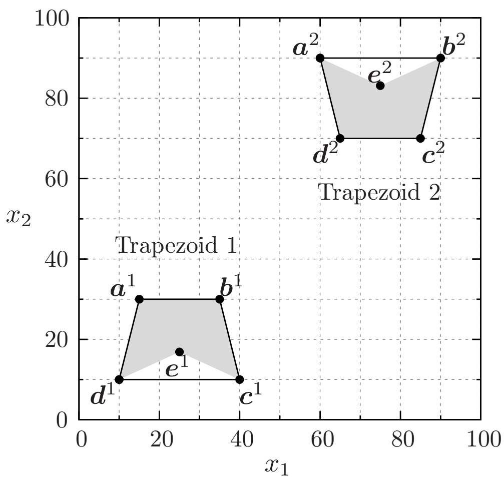



Figure 5.15: Decision space of the polygon-based test problem with trapezoids. Optimal solutions lie in the two gray polygons.

where $\boldsymbol{a}^i$, $\boldsymbol{b}^i$, $\boldsymbol{c}^i$ and $\boldsymbol{d}^i$, $i \in \{1, 2\}$, are vertices of two non-symmetric trapezoids in the decision space. Gray areas correspond to solutions minimizing all four objectives. Note that solutions $\boldsymbol{a}^2$ and $\boldsymbol{b}^2$ as well as $\boldsymbol{c}^1$ and $\boldsymbol{d}^1$ are on the border of the optimal region, but are not optimal themselves because $\boldsymbol{f}(\boldsymbol{a}^1) \prec \boldsymbol{f}(\boldsymbol{a}^2)$, $\boldsymbol{f}(\boldsymbol{b}^1) \prec \boldsymbol{f}(\boldsymbol{b}^2)$, $\boldsymbol{f}(\boldsymbol{c}^2) \prec \boldsymbol{f}(\boldsymbol{c}^1)$ and $\boldsymbol{f}(\boldsymbol{d}^2) \prec \boldsymbol{f}(\boldsymbol{d}^1)$. We denote with $\boldsymbol{e}^1$ and $\boldsymbol{e}^2$ two additional optimal solutions inside the trapezoids.

While the polygon-based test problems have been conceived to visually examine the behavior of MOEAs in the decision space, visualization with prosections enables us to extend this analysis to the objective space. The MOEAs used in the comparison are NSGA-II (Deb et al., 2002), SPEA2 (Zitzler, Laumanns, & Thiele, 2002), SMS-EMOA (Beume, Naujoks, & Emmerich, 2007) and MOEA/D (Zhang & Li, 2007). Each algorithm was run until 20 000 solutions were examined. Algorithms NSGA-II, SPEA2 and SMS-EMOA maintained a population size of 200 solutions, while the population size of MOEA/D was set to 220 (the same as the number of weight vectors used in the algorithm, which cannot be arbitrarily specified). See (Ishibuchi et al., 2011) for a complete list of the remaining parameter settings.

We visualize the approximation sets from (Ishibuchi et al., 2011) that correspond to the final populations of a single run of each algorithm. Since this is a rather easy optimization problem, for which the chosen optimization algorithms are able to find optimal solutions in all runs, this analysis concentrates on the distribution of vectors rather than the convergence to the Pareto front.

Figure 5.16 shows these approximation sets in the decision space. The solutions found by NSGA-II and MOEA/D are less evenly distributed in the decision space than those of the other two algorithms. SPEA2 and SMS-EMOA are able to find a rather even distribution of solutions, but SPEA2 (similarly to NSGA-II) finds also many solutions that are not optimal. It is also interesting to see that MOEA/D was not able to cover well Trapezoid 1, while SMS-EMOA had more difficulties than the other algorithms in finding solutions near $\boldsymbol{a}^2$, $\boldsymbol{b}^2$, $\boldsymbol{c}^1$ and $\boldsymbol{d}^1$.

The same approximation sets are visualized using prosections in Figures 5.17, 5.18 and 5.19. All prosections use section width $d = 2$, while various prosection planes and angles are used to show different parts of the objective space. Gray points denote the Pareto front, which consists of two distinct regions that intersect near the vector $\boldsymbol{f}(\boldsymbol{e}^1) = \boldsymbol{f}(\boldsymbol{e}^2)$. Depending on the chosen prosection, we can see also the images of some trapezoid vertices.

Prosections are able to show that for all algorithms, the distribution of vectors in the objective space is analogous to the distribution in the decision space. We can see how the objective vectors found by NSGA-II are "clustered" near $\boldsymbol{f}(\boldsymbol{c}^2)$ and $\boldsymbol{f}(\boldsymbol{d}^1)$, contrary to the even distribution of objective vectors by SPEA2, and how both algorithms find some objective vectors that do not lie on the Pareto front. A similar reasoning can be applied to the results of MOEA/D—its nonuniform distribution of vectors is visible also in the objective space.

Finally, visualization with prosections can provide a possible explanation of why SMS-EMOA does not find solutions near $\boldsymbol{a}^2$, $\boldsymbol{b}^2$, $\boldsymbol{c}^1$ and $\boldsymbol{d}^1$. Recall that $\boldsymbol{f}(\boldsymbol{c}^2) \prec \boldsymbol{f}(\boldsymbol{c}^1)$. When selecting solutions for the next generation, SMS-EMOA prefers those that contribute more to the overall attained objective space. Interactively rotating the SMS-EMOEA plot from Figures 5.17 shows that the objective vectors close to $\boldsymbol{f}(\boldsymbol{c}^1)$ have significantly worse values in $f_4$ and just slightly better values in the other three objectives compared to $\boldsymbol{f}(\boldsymbol{c}^2)$ and other optimal objective vectors close to $\boldsymbol{f}(\boldsymbol{c}^2)$. This means that if SMS-EMOA finds an objective vector close to $\boldsymbol{f}(\boldsymbol{c}^1)$, its contribution to the overall attained objective space is very small and such a vector gets discarded. The same holds for $\boldsymbol{f}(\boldsymbol{d}^1)$ (see Figure 5.18), as well as $\boldsymbol{f}(\boldsymbol{a}^2)$ and $\boldsymbol{f}(\boldsymbol{b}^2)$.

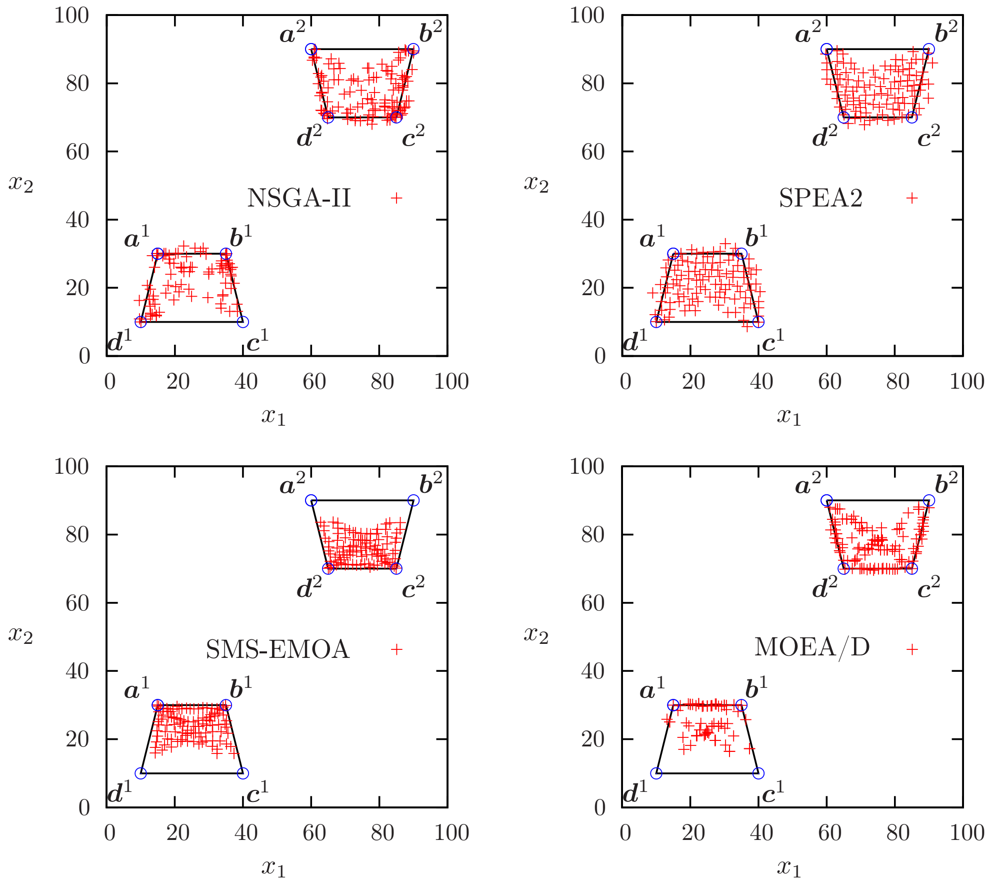



Figure 5.16: Approximation sets of four MOEAs shown in the decision space of the polygon-based test problem with trapezoids.

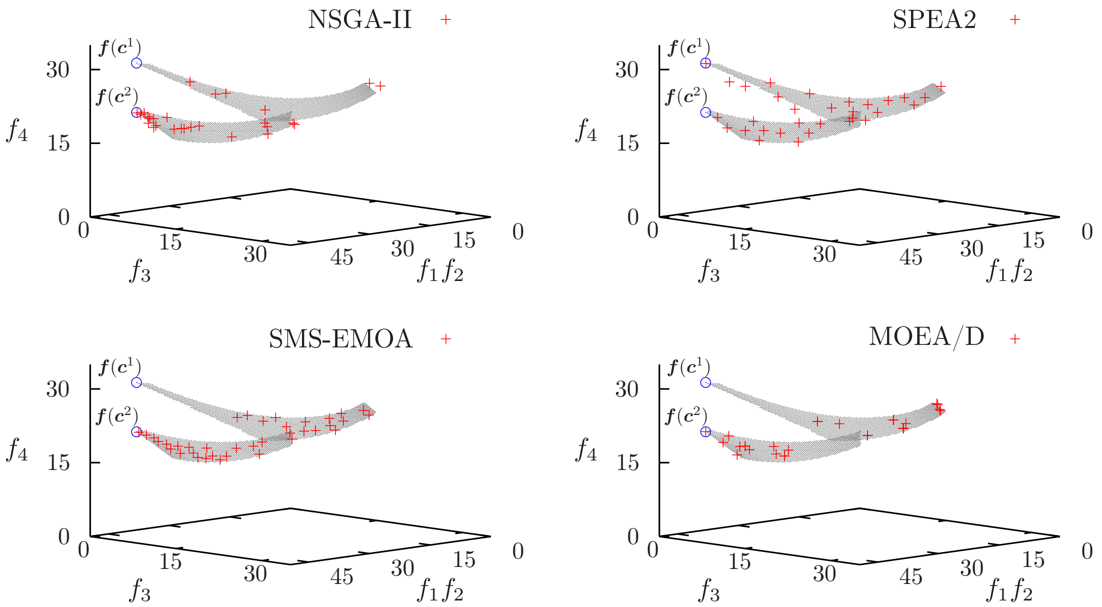



Figure 5.17: Prosection 4D$(\mathbf{0}, f_1 f_2, 35°, 2)$ of the approximation sets of four MOEAs.

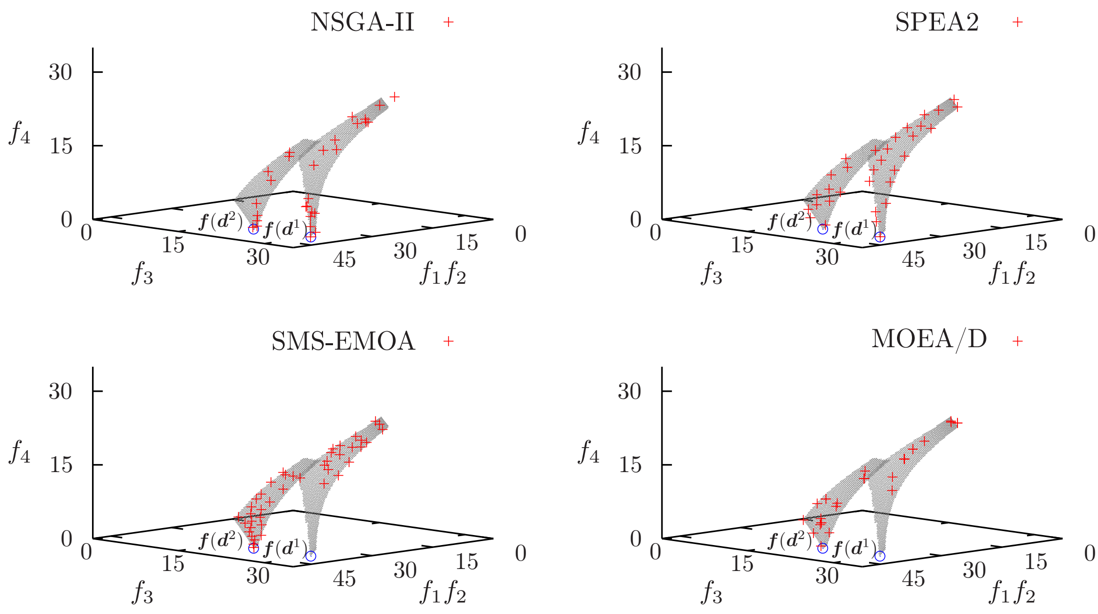



Figure 5.18: Prosection 4D$(\mathbf{0}, f_1 f_2, 55°, 2)$ of the approximation sets of four MOEAs.

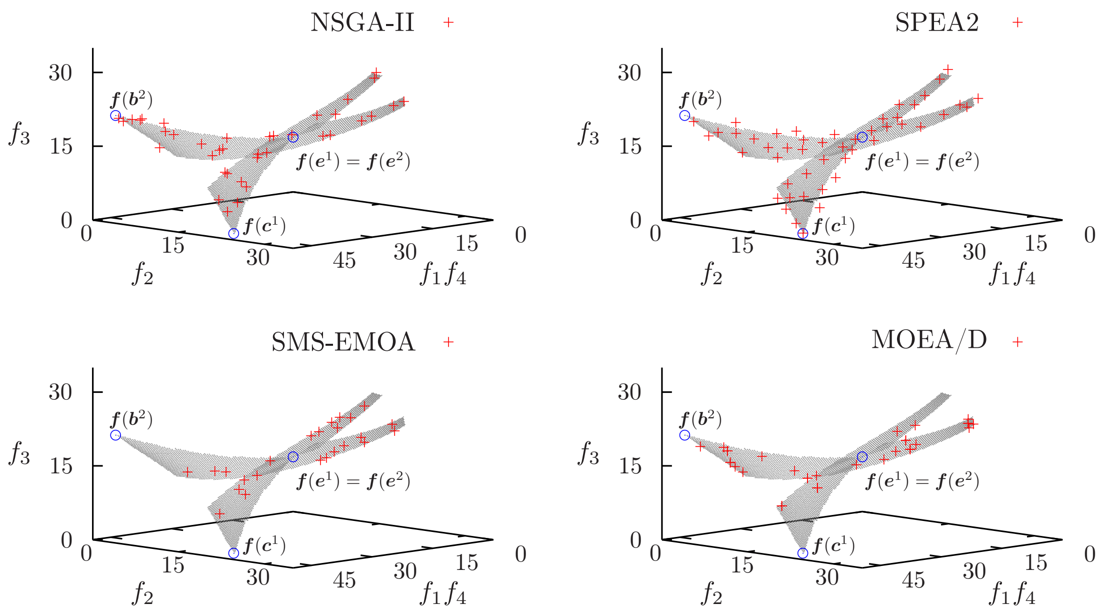



Figure 5.19: Prosection 4D$(\mathbf{0}, f_1 f_4, 45°, 2)$ of the approximation sets of four MOEAs.

## 5.7 Summary

We can look at visualization with prosections in view of the desired properties for a visualization method from Section 1.1.1 (see also Table 4.1). As shown in Section 5.4 (and proven in Appendix A), visualization with prosections is able to preserve the Pareto dominance relation and relative closeness to reference vectors for some vectors, which is crucial for the correct interpretation of the visualized results in the decision making process following optimization. It could also be used to analyze the many-objective algorithms relying on reference vectors to enhance convergence (Deb & H. Jain, 2013). In addition, all the visualized approximation sets have demonstrated that this method is also good at maintaining their shape characteristics (for example, knees) and distribution of vectors. However, because the prosection is done under an angle, the ranges of the two objectives included in the prosection lose some of their meaning (see Section 5.4).

Further, prosections are as robust as scatter plots—addition or removal of a vector does not importantly change the visualization. Because only a part of the approximation set is visualized at a time, prosections are able to visualize large approximation sets. Moreover, they are computationally inexpensive as they require only a simple mapping to be performed on the chosen two objectives.

One of the best properties of prosections is that they are able to visualize two (or more) approximation sets simultaneously, therefore allowing for direct comparison between different approximation sets. This means that they can be used to study the quality of convergence to the Pareto front (if known), to compare different MOEAs and visualize the progress of a single MOEA.

On the other hand, their biggest disadvantage in their present form is that they are limited to 4D approximation sets. Scalability to five or more dimensions is challenging for two reasons. First, applying prosection multiple times might result in a loss of ability to preserve the Pareto dominance relation and maintain the shape, range and distribution of vectors, which is essential for producing meaningful visualizations. Second, because only a small part of the objective space is visualized at a time, this would scale too, meaning that it would become impractical to show enough views to gain a good understanding of the approximation set. Handling these challenges is a task for future work.

Finally, a note on whether prosections are simple to understand and use. The idea behind prosections is very simple—they slice through the 4D approximation set and visualize this slice in 3D. However, in practice some parameters need to be set (see Section 5.3) and several prosection planes and angles should be explored to gain a full understanding of an approximation set. This makes them less simple to use. To enhance their usability, we suggest to adhere to the following procedure for visualization of a 4D approximation set, where all four objectives need to be minimized[3]:

1. If the objectives are disproportionate, normalize the approximation sets to the interval $[0, 1]^4$ (for the sake of brevity we assume the approximation sets are normalized from this point on).

2. Set the vector $\mathbf{0}$ as the origin and choose $d$ depending on the size of the approximation sets (for example, $d \in [0.02, 0.05]$ for approximation sets with 3000 vectors and can be smaller for larger sets and larger for smaller sets).

3. Look at the prosection matrix at different angles (for example $\varphi = 0°, 10°, \ldots, 90°$) either with separate plots or animation.

4. Choose the prosection plane and angle $\varphi$ that give most information and visualize and analyze only this one.

5. Repeat the previous step if needed.

Using this procedure the user gets a small number of meaningful visualizations that can help better understand the 4D approximation sets.

[3] A few `gnuplot` scripts to support this visualization procedure can be found at http://dis.ijs.si/tea/prosections.htm

# Part II

# Visualization of Empirical Attainment Functions

# Chapter 6

# Related Work

To the best of our knowledge, there have been no attempts to visualize 3D EAF values and differences before. Therefore this chapter presents related work regarding visualization of 3D attainment surfaces and 2D EAF values and differences (Section 6.1). We also introduce the Slicing, MIP and DVR methods that can be used for visualizing spatial data (see Sections 6.2, 6.3 and 6.4, respectively).

## 6.1 Visualization of EAFs

2D EAFs are most often visualized through best, median and worst summary attainment surfaces corresponding to 1/r %, $\sim$ 50% and 100%-attainment surfaces, respectively. The first (and only) method for visualizing 3D summary attainment surfaces was documented by Knowles (2005). At the time, an efficient algorithm for computing 3D attainment anchors was not yet available, so separate summary attainment surfaces were approximated by discretizing the objective space using a grid. While the procedure could be used for more than three objectives, we will only describe the 3D case here. Assume the $t/r$-attainment surface needs to be visualized. The algorithm examines the objective space for each objective separately. First, it finds the intersections between all approximation sets and the lines stemming from the 2D grid of the remaining two objectives. Then, the intersections on each of these lines are counted and if they amount to $t$, a marker is drawn at the corresponding height. Combining these markers by considering all objectives yields an informative visualization of the chosen 3D summary attainment surface. Years later, an efficient algorithm for computing attainment anchors and their EAF values for the 3D case was finally provided by Fonseca et al. (2011).

Recently, the idea of visualizing EAF differences in addition to EAF values was introduced by López-Ibáñez et al. (2010). When visually comparing algorithms $A$ and $B$, four plots can be produced—each corresponding to one of the values of $\alpha^{\mathrm{A}}$, $\alpha^{\mathrm{B}}$, $\delta^{\mathrm{A-B}}$ and $\delta^{\mathrm{B-A}}$. All attainment anchors are plotted using gray-shaded dots, where the shade corresponds to the value of the chosen function ($\alpha^{\mathrm{A}}$, $\alpha^{\mathrm{B}}$, $\delta^{\mathrm{A-B}}$ or $\delta^{\mathrm{B-A}}$). On top of this, the best, median and worst overall summary attainment surfaces are drawn. Examples from (López-Ibáñez et al., 2010; López-Ibáñez & Stützle, 2012) show that this kind of visualization can be very useful in exploratory analysis.

## 6.2 Slicing

This is a simple method where spatial data is sliced using a plane. Then, the intersection of the plane with the data is visualized in 2D. In its most general form, Slicing can be applied to any kind of spatial data, i.e. the data does not have to be represented by voxels.

While the plane used in Slicing is most often axis-aligned, the nature of multiobjective optimization problems yields the need for a plane that always contains one axis and the origin of the objective space (which usually corresponds to the ideal objective vector) and intersects all summary attainment surfaces. Therefore (similarly to the *prosection plane* introduced in Section 5.1), the slicing plane cuts the objective space at an angle $\varphi \in (0°, 90°)$.

While not used for visualizing attainment surfaces, the Interactive Decision Maps by Lotov and Miettinen (2008), see also Section 4.2.2, are related to Slicing using axis-aligned planes. They visualize a number of axis-aligned sampling surfaces of the EPH and use different colors to represent each slice. As the surface of $Z_{\text{EPH}}$ is exactly the attainment surface of $Z$, it should be rather straightforward to apply this method for visualizing attainment surfaces.

## 6.3 Maximum Intensity Projection

Maximum Intensity Projection (MIP), originally called Maximum Activity Projection (Wallis, Miller, Lerner, & Kleerup, 1989), is a volume rendering method for spatial data represented by voxels. 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. MIP 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 (see (Heidrich, McCool, & Stevens, 1995; Díaz & Alcocer, 2010) for attempts to remedy this issue). Moreover, the viewer cannot distinguish between left and right and front and back. 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.

## 6.4 Direct Volume Rendering

Direct Volume Rendering (DVR) is another volume rendering method based on the voxel representation of data that is often used for visualization in medicine and science (Engel, Hadwiger, Kniss, Rezk-Salama, & Weiskopf, 2006). It can generate very appealing results by employing advanced shading techniques and can achieve real-time interactive rendering. DVR is able to produce images of volumetric data without the need to explicitly extract geometry or surfaces (hence the name *direct*).

First, each voxel value is assigned a color and opacity (the RGBA value) by means of a chosen *transfer function*. The transfer function can be a simple ramp, a piecewise linear function or an arbitrary table (the problem of specifying a good transfer function is a research area on its own (Botha & Post, 2002; Šereda, Vilanova, & Gerritsen, 2006)). Then, one of the following techniques is used to project the RGBA values to the screen: ray casting[1], splatting, shear-warp factorization or texture-based volume rendering. In ray casting (Levoy, 1988), viewing rays are traced from the viewpoint through the volume, accumulating color and opacity values at each sample position along the ray. The final

[1] As only ray casting is used in this thesis, from now on any mention of DVR can be understood to refer to ray-casting DVR.

RGB values shown in the image are computed from the accumulated RGBA values using the volume rendering integral (Max, 1995).

In addition to presenting existing efforts in visualization of EAFs, this chapter described three methods for visualizing spatial data: Slicing, MIP and DVR. The next chapter will show how they can be employed for visualization of 3D EAF values and differences.

# Chapter 7

# Proposed Visualization Methods

This chapter presents three visualization methods that can be used for visualizing exact and approximated 3D EAFs and shows their outcome on two sets of BASes. First, Section 7.1 introduces the chosen BASes that are used in the visualizations. Next, visualization of exact and approximated EAFs is presented in Sections 7.2 and 7.3, respectively. Finally, the proposed methods are summarized in Section 7.4.

## 7.1 BASes

In order to show the outcome of different visualization methods, we reuse the BASes from Chapter 3. However, this time we need a set of linear and a set of spherical BASes instead of single instances.

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

$$\sum_{i=1}^{3} z_i = c_{\mathrm{L}}. \tag{7.1}$$

In order to simulate the behavior of a stochastic algorithm that is not always able to achieve the Pareto front, the values $c_{\mathrm{L}}$ are not equal for all sets. We set them to follow the normal distribution $N(1, 0.05)$. The second set contains five spherical BASes with a nonuniform distribution of vectors. The vectors from the spherical BASes satisfy the constraint

$$\sum_{i=1}^{3} z_i^2 = c_{\mathrm{S}}^2, \tag{7.2}$$

where values $c_{\mathrm{S}}$ follow the normal distribution $N(0.75, 0.05)$. Each individual approximation set contains 100 objective vectors. All ten BASes are presented in Figure 7.1.

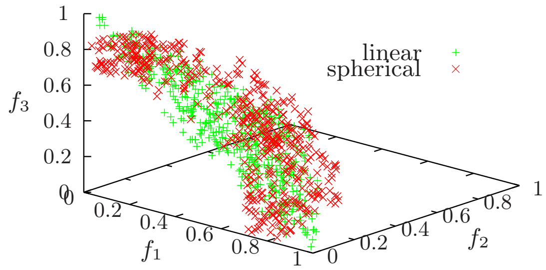



Figure 7.1: The chosen linear and spherical BASes.

The values of $c_\mathrm{L}$ and $c_\mathrm{S}$ were chosen so that the sets are intertwined—in one region, the linear sets dominate the spherical ones, while in others, the spherical sets dominate the linear ones. This assures that there will always be visible EAF differences between the two sets of approximation sets.

We will use $\alpha_5^\mathrm{L}$ and $\alpha_5^\mathrm{S}$ to denote EAF values of the linear and spherical BASes, respectively, and $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ to denote differences is those values.

## 7.2 Visualizing Exact 3D EAFs

Exact 3D EAFs can be represented as unions of cuboids with values corresponding to either EAF values of one algorithm or EAF differences between two algorithms (Tušar & Filipič, 2013, 2014c). This section first describes the procedure for computing the cuboids from the given attainment surfaces (Section 7.2.1). Then, visualization of the cuboids using Slicing and MIP is shown in Sections 7.2.2 and 7.2.3, respectively.

### 7.2.1 Computing the Cuboids

The algorithm for computing EAFs (Fonseca et al., 2011) takes as input a series of approximation sets $Z_1, \ldots, Z_r$ and returns as output a series of sets containing attainment anchors $A_1, \ldots, A_r$ that define the corresponding summary attainment surfaces and for which the following holds: $A_i \preceq A_{i+1}$ for $i \in \{1, 2, \ldots, r-1\}$. From these sets of attainment anchors and a reference vector $\boldsymbol{r}^2$ that limits the observed objective space, the cuboids can be computed. We propose a simple algorithm for this purpose that uses the first set of attainment anchors and the opposite of the second set of attainment anchors to compute the overlapping cuboids (see Algorithm 7.1).

First, a set containing only the given reference vector $\boldsymbol{r}^2$ is added at the end of the input set of attainment anchors, so that any cuboids ranging from the last set of attainment anchors to $\boldsymbol{r}^2$ can also be computed. Then, the reference vector $\boldsymbol{r}^1$ needed for computation of the opposites is set to be equal to the ideal objective vector. Next, the main loop iterates through all adjacent pairs of sets $A_i$ and $A_{i+1}$ and computes the opposite of $A_{i+1}$ using Algorithm 7.2. For each attainment anchor $\boldsymbol{z} \in A_i$ all dominated vectors from the opposite are stored in $O'$. Each pair $(\boldsymbol{z}, \boldsymbol{o})$, where $\boldsymbol{o} \in O'$, represents the two vertices required to

**Algorithm 7.1:** Computing the cuboids. The opposite$(A_{i+1}, \boldsymbol{r}^1, \boldsymbol{r}^2)$ function is detailed in Algorithm 7.2. The cuboid$(\boldsymbol{z}, \boldsymbol{o}$, value$(\boldsymbol{z}))$ function constructs a cuboid with vertices $\boldsymbol{z}$ and $\boldsymbol{o}$ and the value of $(\boldsymbol{z})$.

**Input**: Sets of attainment anchors $A_1, \ldots, A_r$ and a reference vector $\boldsymbol{r}^2$, for which $A_1 \preceq \cdots \preceq A_r$ and $\boldsymbol{z} \prec\!\!\prec \boldsymbol{r}^2$ for all $\boldsymbol{z} \in A_i, i \in \{1, \ldots, r\}$
**Output**: The set of cuboids $C$

$A_{r+1} \leftarrow \{\boldsymbol{r}^2\}$;
$\boldsymbol{r}^1 \leftarrow \boldsymbol{z}^*$;
**foreach** adjacent pair of sets $A_i$ and $A_{i+1}$, $i \in \{1, \ldots, r\}$ **do**
  $O \leftarrow$ opposite$(A_{i+1}, \boldsymbol{r}^1, \boldsymbol{r}^2)$;
  **foreach** $\boldsymbol{z} \in A_i$ **do**
    $O' \leftarrow \{\boldsymbol{o} \in O; \boldsymbol{z} \prec\!\!\prec \boldsymbol{o}\}$;
    **foreach** $\boldsymbol{o} \in O'$ **do**
      $C \leftarrow C \cup$ cuboid$(\boldsymbol{z}, \boldsymbol{o}$, value$(\boldsymbol{z}))$;
    **end**
  **end**
**end**

**Algorithm 7.2:** Computing the opposite of a 3D approximation set. The redundant($\boldsymbol{o}^{n_i}, \boldsymbol{r}^1, O$) function returns true if $\boldsymbol{o}^{n_i}$ does not contribute to the opposite.

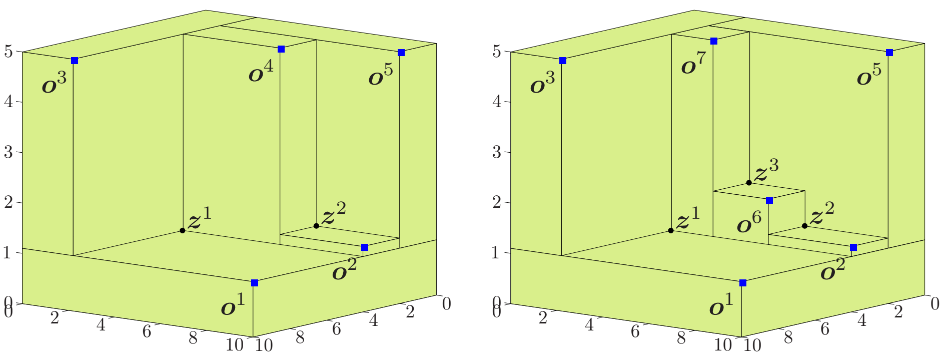
**Input**: Approximation set $A$ and reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$, for which $\boldsymbol{r}^1 \preceq \boldsymbol{z} \prec\!\!\prec \boldsymbol{r}^2$ for all $\boldsymbol{z} \in A_i, i \in \{1, \dots, r\}$
**Output**: The opposite $O$

$O \leftarrow \{\boldsymbol{r}^2\}$;
**foreach** $\boldsymbol{z} \in A$ **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** $\neg$ redundant($\boldsymbol{o}^{n_i}, \boldsymbol{r}^1, O$) **then**
        $O \leftarrow O \cup \{\boldsymbol{o}^{n_i}\}$;
      **end**
    **end**
  **end**
**end**

define the cuboid. In addition, the cuboid is given a value—either the EAF value or the EAF difference in $\boldsymbol{z}$.

Note that there might be multiple cuboids stemming from the same attainment anchor (see the analog example of multiple rectangles in Figure 2.4)—we call this a *union of cuboids*. This is not a problem since all such overlapping cuboids have the same value (if they did not have the same value, another attainment anchor would have already split the cuboid). The result of Algorithm 7.1 is a set of cuboids (or unions of cuboids).

Next, we show with the help of Figure 7.2 how to find the opposite of a 3D approximation set. Imagine a single cuboid defined by the reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$. The opposite is initialized to $\{\boldsymbol{r}^2\}$. Every time a vector from $Z$ is "cut into" the existing cuboid, the vectors strictly dominated by it are removed from the opposite and new vectors are added to the opposite. Assume we have already performed this step for vectors $\boldsymbol{z}^1$ and $\boldsymbol{z}^2$ (see Figure 7.2a) and vector $\boldsymbol{z}^3$ is next in line (see Figure 7.2b). First, we delete from the




(a) Opposite of $\{\boldsymbol{z}^1, \boldsymbol{z}^2\}$ (b) Opposite of $\{\boldsymbol{z}^1, \boldsymbol{z}^2, \boldsymbol{z}^3\}$

Figure 7.2: Constructing the opposite of a 3D approximation set. The plots show two consecutive steps for approximation set $Z$.

opposite all vectors that are strictly dominated by the current vector $\boldsymbol{z}$ (this means that in our 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:

$$\begin{aligned} \boldsymbol{o}^{n_1} &= (z_1, o_2, o_3) \\ \boldsymbol{o}^{n_2} &= (o_1, z_2, o_3) \\ \boldsymbol{o}^{n_3} &= (o_1, o_2, z_3) \end{aligned} \tag{7.3}$$

Finally, each of these vectors is added to the opposite if it is not redundant. This means that it has to contribute to the opposite (it cannot be coplanar with $\boldsymbol{r}^1$ nor collinear with any of the vectors from the existing 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 $\boldsymbol{o}^5$ and thus redundant. When these steps have been taken for every vector from $Z$, the resulting set $O$ represents the opposite of $Z$. See Algorithm 7.2 for the algorithmic notation of the described procedure.

As we cannot visualize the entire (infinite) objective space, Algorithms 7.1 and 7.2 use two reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$ to limit it. In order to preserve all information, they need to be set in such a way that $\boldsymbol{r}^1 \preceq \boldsymbol{z} \prec\!\!\prec \boldsymbol{r}^2$ for all attainment anchors $\boldsymbol{z}$. While the ideal point is the most sensible choice for $\boldsymbol{r}^1$, we can choose any objective vector strictly dominated by all attainment anchors for $\boldsymbol{r}^2$.

The presented approach for computing the cuboids is simple and easy to implement, but comes at a high computational cost. Computing the cuboids between two sets of $n$ attainment anchors has a worst-case computational complexity of $O(n^3)$, which means that all cuboids between $r$ pairs of attainment surfaces can be computed in $O(rn^3)$ time. However, in practice, the loop **foreach** $\boldsymbol{o} \in O'$ **do** in Algorithm 7.2 (lines 5 to 14) is executed only a few times for each $\boldsymbol{z} \in A$, meaning that in practice the computational complexity is quadratic in the size of the attainment anchors set (the check for redundancy still takes linear time).

Another possible approach would be to deal only with non-overlapping cuboids, which is somewhat similar to the problem of computing the hypervolume indicator in 3D. In future work we wish to investigate if a more efficient algorithm for our problem could be found by, for example, modifying the space-sweep algorithm by Beume, Fonseca, López-Ibáñez, Paquete, and Vahrenhold (2009), which computes the hypervolume indicator in the 3D case with a computational complexity of only $O(n \log n)$. Should this be possible, the total cost of computing the cuboids would be $O(rn \log n)$. However, time complexity is not the only important aspect in the computation of cuboids. It would be interesting to study which method would yield a lower number of cuboids as this considerably affects the tractability of their visualization.

### 7.2.2 Visualizing Cuboids Using Slicing

The slicing plane is always aligned with one of the three axes (assume for now that this is $f_3$), includes the origin $\boldsymbol{r}^1$ and cuts through the underlying plane ($f_1 f_2$) at some angle $\varphi \in (0°, 90°)$. In this way, each slice always captures all attainment surfaces.

Slicing through the objective space containing a large number of cuboids means that only those cuboids that intersect the slicing plane are visualized. When a cuboid is sliced, the result is a rectangle in 3D (see the example in Figure 7.3). Two steps are needed to compute this rectangle in 2D. First, we need to calculate the projected cuboid vertices $\boldsymbol{z}'$ and $\boldsymbol{o}'$ yielding the rectangle in 3D, and second, we need to rotate the vertices by angle

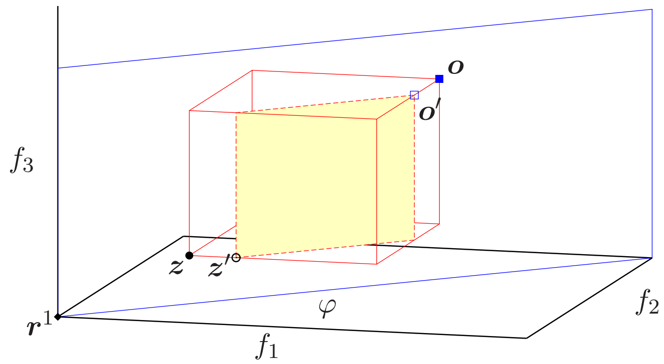



Figure 7.3: Slicing a cuboid. The slicing plane is aligned with $f_3$ that slices the $f_1 f_2$ plane under angle $\varphi$. The cuboid vertices $\boldsymbol{z}$ and $\boldsymbol{o}$ are projected to 3D rectangle vertices $\boldsymbol{z}'$ and $\boldsymbol{o}'$, respectively.

$-\varphi$ to get $\boldsymbol{z}''$ and $\boldsymbol{o}''$. Depending on the angles

$$\varphi_{\boldsymbol{z}} = \arctan\left(\frac{z_2 - r_2^1}{z_1 - r_1^1}\right) \text{ and } \varphi_{\boldsymbol{o}} = \arctan\left(\frac{o_2 - r_2^1}{o_1 - r_1^1}\right), \tag{7.4}$$

this is done in the following way:

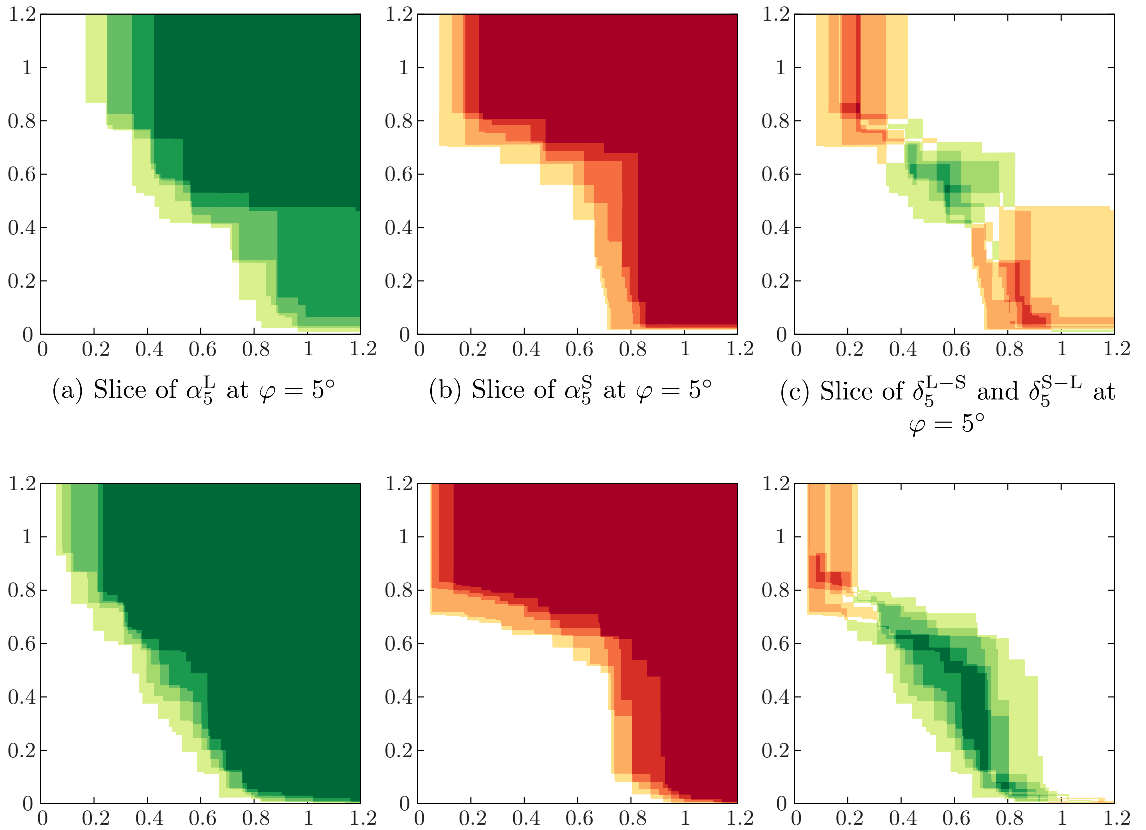
$$\begin{aligned}
\boldsymbol{z}' &= \begin{cases} (r_1^1 + \frac{z_2 - r_2^1}{\tan\varphi}, z_2, z_3) & \text{if } \varphi_{\boldsymbol{z}} \geq \varphi \\ (z_1, r_2^1 + (z_1 - r_1^1)\tan\varphi, z_3) & \text{if } \varphi_{\boldsymbol{z}} < \varphi \end{cases} \\
\boldsymbol{o}' &= \begin{cases} (o_1, r_2^1 + (o_1 - r_1^1)\tan\varphi, o_3) & \text{if } \varphi_{\boldsymbol{o}} \geq \varphi \\ (r_1^1 + \frac{o_2 - r_2^1}{\tan\varphi}, o_2, o_3) & \text{if } \varphi_{\boldsymbol{o}} < \varphi \end{cases} \\
\boldsymbol{z}'' &= \begin{cases} (r_1^1 + \frac{z_2 - r_2^1}{\sin\varphi}, z_3) & \text{if } \varphi_{\boldsymbol{z}} \geq \varphi \\ (r_1^1 + \frac{z_1 - r_1^1}{\cos\varphi}, z_3) & \text{if } \varphi_{\boldsymbol{z}} < \varphi \end{cases} \\
\boldsymbol{o}'' &= \begin{cases} (r_1^1 + \frac{o_1 - r_1^1}{\cos\varphi}, o_3) & \text{if } \varphi_{\boldsymbol{o}} \geq \varphi \\ (r_1^1 + \frac{o_2 - r_2^1}{\sin\varphi}, o_3) & \text{if } \varphi_{\boldsymbol{o}} < \varphi \end{cases}.
\end{aligned} \tag{7.5}$$

When slicing through a union of cuboids stemming from the same vertex $\boldsymbol{z}$, some of the projected vertices $\boldsymbol{o}''$ can be redundant. In order to decrease the number of total rectangles to plot, it is therefore sensible to keep only non-redundant vertices $\boldsymbol{o}''$ (those that are nondominated with regard to the inverted weak Pareto dominance relation $\succeq$).

The cuboids can be visualized by slicing them at different angles, thus showing different parts of the objective space. We present visualization using Slicing of our BASes at angles $\varphi = 5°$ and $\varphi = 45°$. The EAF values $\alpha_5^{\mathrm{L}}$ and $\alpha_5^{\mathrm{S}}$ are shown separately, while the EAF differences $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ are presented on the same plot (see Figure 7.4).

From the plots of EAF values it is easy to distinguish linear sets (green hues) from the spherical ones (red hues) as the shape of the sets is readily visible. We can also see that the two best spherical BASes are better than the remaining three by a solid margin. While these plots provide a lot of information by themselves, the comparison between the sets is best visualized using EAF differences. They nicely show regions in the objective space where linear sets outperform the spherical ones and vice versa.

Note that in order to fully explore the entire objective space, slicing at several different angles should be performed. Also, one might want to consider slicing planes aligned to the




(a) Slice of $\alpha_5^{\mathrm{L}}$ at $\varphi = 5°$ (b) Slice of $\alpha_5^{\mathrm{S}}$ at $\varphi = 5°$ (c) Slice of $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ at $\varphi = 5°$

(d) Slice of $\alpha_5^{\mathrm{L}}$ at $\varphi = 45°$ (e) Slice of $\alpha_5^{\mathrm{S}}$ at $\varphi = 45°$ (f) Slice of $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ at $\varphi = 45°$

Figure 7.4: Slices of exact 3D EAF values and differences for the BASes under two angles. Darker colors represent higher EAF values/differences.

axis $f_1$ or $f_2$, too. However, as our BASes are rather symmetrical, there is no need to do this here.

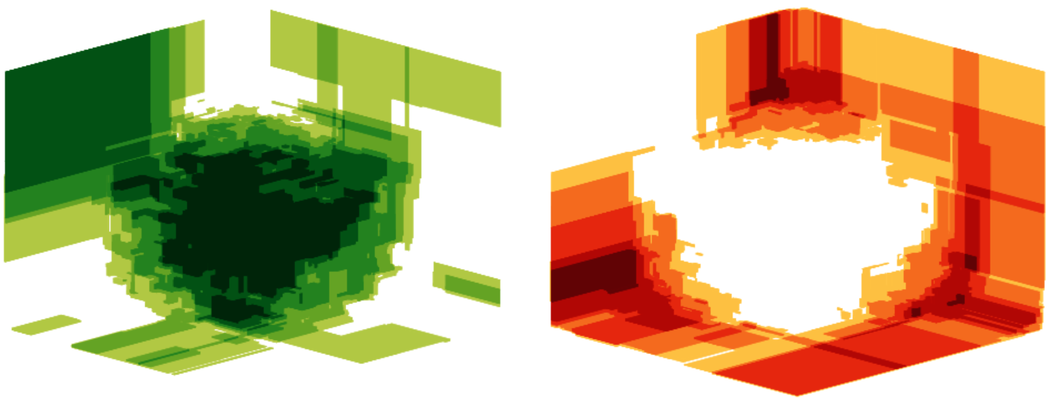
### 7.2.3 Visualizing Cuboids Using MIP

As explained in Section 6.3, MIP shows only the maximum value encountered on rays from the viewpoint to the projection plane. This means that it is not sensible to use this method for visualizing EAF values as most of the objective space has the maximum value and the resulting visualization would not be very informative. However, MIP seems a good choice for visualizing EAF differences where the highest values are of most interest as they represent the largest differences between the algorithms.

MIP could be used to visualize cuboids of different values by sorting the cuboids so that those with higher values would be put on top of those with lower values. However in general, the 3D plotting tools capable of visualizing cuboids render them in a sequence that tries to maintain some notion of depth (cuboids near the viewpoint are shown in front of the cuboids further away) and do not allow for custom sorting of the cuboids according to their values. Therefore, this sorting can only be done after the viewpoint has been set and the cuboids are already projected to 2D. At that time we can choose to visualize them in the order of ascending values, which (although a bit cumbersome) effectively achieves the MIP visualization of the cuboids.

Another difficulty in using MIP for visualizing exact EAF differences is that we need to visualize a large number of cuboids, which can be challenging to do. While many of them are completely covered by cuboids with larger values and could therefore be spared without altering the final image, removal of such cuboids is not trivial as it depends also on the position of the viewpoint.

Nevertheless, we present visualization of exact EAF differences using MIP in Figure 7.5. The differences $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ need to be visualized separately in order to avoid overlapping of cuboids. The MIP visualizations are very informative—we can easily see which parts of the objective space are better attained by the linear and which by the spherical BASes. As it is typical with MIP, while we are able to "see through the cuboids", we lose the sense of depth. Although 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. Together they give a good idea of the 3D "cloud" of cuboids.



(a) MIP of $\delta_5^{\mathrm{L-S}}$ (b) MIP of $\delta_5^{\mathrm{S-L}}$

Figure 7.5: MIP of exact 3D EAF differences between the BASes.

## 7.3 Visualizing Approximated 3D EAFs

If we wish to visually compare the outcome of two algorithms and are not interested in the details of such a visualization, we can use approximated instead of exact EAFs. Approximation means that the objective space is discretized into a grid of voxels (similarly to what was proposed by Knowles (2005)). This section first presents how such a discretization is performed and then illustrates visualization of approximated EAFs using Slicing, MIP and DVR.

### 7.3.1 Discretization into Voxels

Let us consider a regularly spaced 3D grid of $v_1 \cdot v_2 \cdot v_3$ voxels. Recall that based on the reference vectors $\boldsymbol{r}^1$ and $\boldsymbol{r}^2$, the 3D observed objective space $R$ is defined as

$$R = \{\boldsymbol{z} \in F; z_i \in [r_i^1, r_i^2] \text{ for all } i \in \{1, 2, 3\}\}. \tag{7.6}$$

If $R$ is a cube, $v_1 = v_2 = v_3$, otherwise some care must be taken to ensure that the grid of voxels is truly regular. Either $R$ must be normalized prior to approximation, or the number of voxels in each dimension $v_i$ must be set to be proportional to $r_i^2 - r_i^1$ for all

$i \in \{1, 2, 3\}$. Note that a voxel represents only a single point, not a volume. How this missing information is reconstructed depends on the visualization method.

From the observed objective space $R$ the grid of $v_1 \cdot v_2 \cdot v_3$ voxels is constructed so that the voxel at grid position $(k_1, k_2, k_3)$ has the following coordinates:

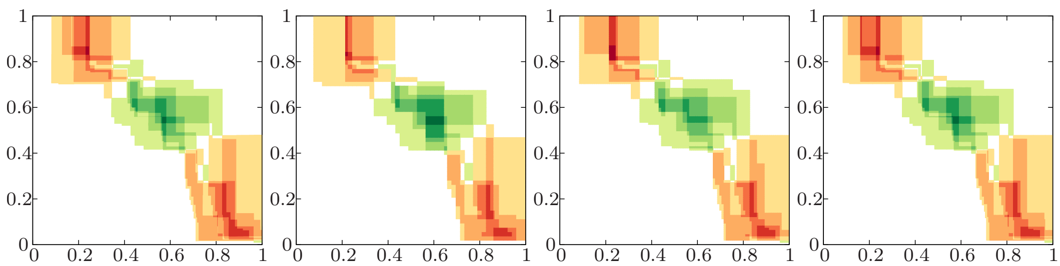
$$\left( \frac{(r_1^2 - r_1^1)(2k_1 - 1)}{2v_1}, \frac{(r_2^2 - r_2^1)(2k_2 - 1)}{2v_2}, \frac{(r_3^2 - r_3^1)(2k_3 - 1)}{2v_3} \right), \tag{7.7}$$

where $k_i \in \{1, \ldots, v_i\}$ for $i \in \{1, 2, 3\}$.

Computing voxel values if the cuboids have already been computed is rather straightforward. Iterating over all cuboids, the voxels that are "covered" by the cuboid adopt its value. If we are not interested in visualizing the exact EAFs (and therefore have not computed the cuboids), we can compute the voxel values directly from the set of approximation sets. The value of each voxel is set to either EAF value or EAF difference by counting how many approximation sets weakly dominate it.

### 7.3.2 Visualizing Voxels Using Slicing

For visualization using Slicing we assume that the whole volume of the voxel has the same value as its center. Therefore, slicing of voxels is done in the same way as slicing of cuboids (see Section 7.2.2). In order to avoid showing plots that are very similar to those presented in Figure 7.4, Figure 7.6 shows only the slices of $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ at angle $\varphi = 5°$ produced using different discretizations.




(a) Using cuboids (b) Using $64^3$ voxels (c) Using $128^3$ voxels (d) Using $256^3$ voxels

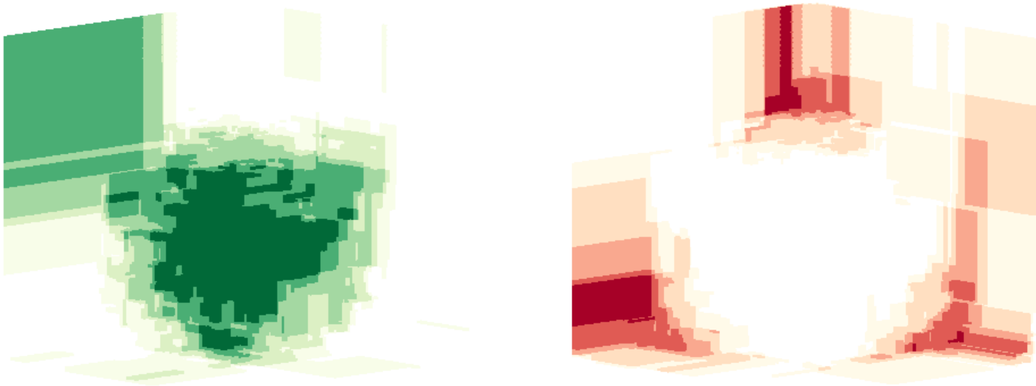
Figure 7.6: Effects of different discretizations. The plots show slices of exact and approximated EAF differences $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ at angle $\varphi = 5°$.

We can see that by refining the discretization we get results that are increasingly similar to the exact EAFs. As we believe that for the purpose of this study the discretization into $128^3$ voxels suffices, we will use this discretization in the remainder of the thesis.

### 7.3.3 Visualizing Voxels Using MIP

Visualizing voxels using MIP is trivial with a tool supporting such visualization, as there are no additional parameters to set. We use a volume rendering engine called Voreen (Meyer-Spradow, Ropinski, Mensmann, & Hinrichs, 2009; "Voreen, Volume rendering engine," 2014) to produce the MIP images from Figure 7.7 and all DVR images.

Figure 7.7 shows the MIP for $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$ and we can see that although these are approximations of the images presented in Figure 7.5, the results are quite similar.



(a) MIP of $\delta_5^{\mathrm{L-S}}$ (b) MIP of $\delta_5^{\mathrm{S-L}}$

Figure 7.7: MIP of approximated 3D EAF differences between the BASes.

### 7.3.4 Visualizing Voxels Using DVR

Visualization using DVR is a bit trickier as we need to define the transfer function that assigns color and opacity to each voxel value. In our case, voxel values are discrete as they equal either the EAF values or the EAF differences, which makes this task easier.

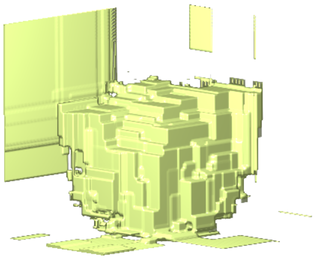
Figures 7.8 and 7.9 show visualization using DVR of the EAF differences between the two sets of BASes $\delta_5^{\mathrm{L-S}}$ and $\delta_5^{\mathrm{S-L}}$, respectively. The first five plots in both figures show the voxels with a single value from $\{1/5, \dots, 5/5\}$. The transfer functions used to obtain these plots are simple piecewise linear functions that set the opacity of voxels of the desired value to 1 and the opacity of all other voxels to 0. In this way, we are able to visualize each of the values separately, which is reflected in the nice and informative plots in Figures 7.8 and 7.9.

The biggest advantage of DVR is that by setting the transfer function "the right way", it is possible to see inside the visualized volume. This was attempted on the plots shown in Figures 7.8f and 7.9f, which show completely opaque voxels with value 5/5 and almost transparent voxels with value 1/5 (both plots are rotated to give a nicer view of inner voxels).

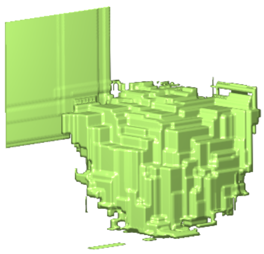
In contrast to MIP, DVR can be used for visualizing EAF values, too. Figure 7.10 shows an example of such a visualization for the spherical BASes, where only the values $\delta_5^{\mathrm{S}} \in \{1/5, 4/5\}$ are shown (opacity is set to 0.3 for these two values and to 0 for all other values).

## 7.4 Summary

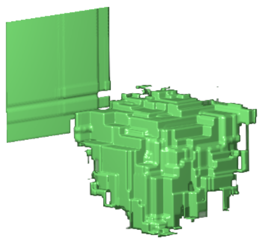
Let us summarize the properties of all presented visualization methods (see also Table 7.1). Slicing is a rather simple method that enables visualization of exact EAF values and differences. Its biggest advantage is its accuracy—it enables a detailed analysis of the approximation sets in the same way that can be done for 2D EAFs. Its biggest drawback is that it is able to visualize only one slice at a time. Generally, the approximation sets are not as symmetrical as in our benchmark case, therefore multiple slices are needed to sufficiently inspect the EAFs. Also, note that the meaning of the angle $\varphi$ changes if the observed objective space has different ranges (if $R$ is not a cube, is not cut exactly in half by the plane at angle $\varphi = 45°$). While Slicing can be used also for visualizing approximated EAFs, there is no need to do so as it works well for exact EAFs.



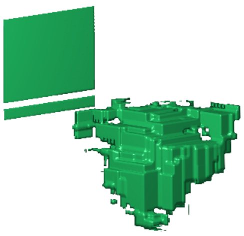
(a) DVR of $\delta_5^{\mathrm{L-S}} = 1/5$



(b) DVR of $\delta_5^{\mathrm{L-S}} = 2/5$

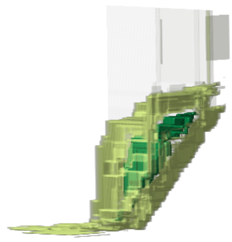


(c) DVR of $\delta_5^{\mathrm{L-S}} = 3/5$



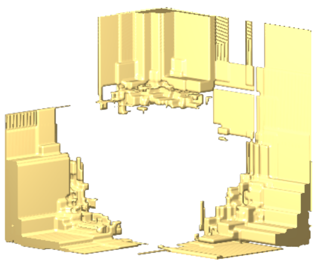
(d) DVR of $\delta_5^{\mathrm{L-S}} = 4/5$

(e) DVR of $\delta_5^{\mathrm{L-S}} = 5/5$

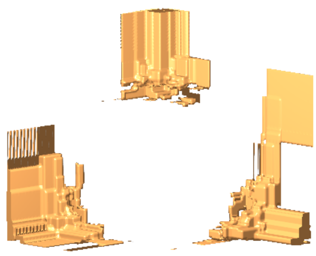


(f) DVR of $\delta_5^{\mathrm{L-S}} \in \{1/5, 5/5\}$

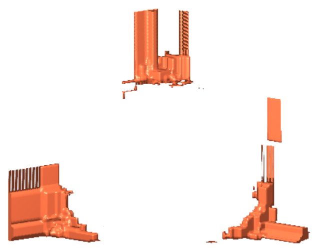
Figure 7.8: DVR of approximated 3D EAF differences between the linear and spherical BASes $\delta_5^{\mathrm{L-S}}$.



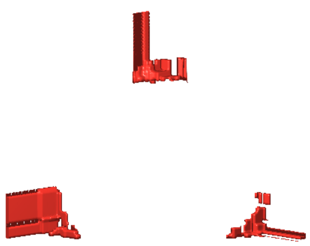
(a) DVR of $\delta_5^{\mathrm{S-L}} = 1/5$



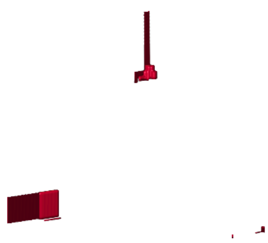
(b) DVR of $\delta_5^{\mathrm{S-L}} = 2/5$



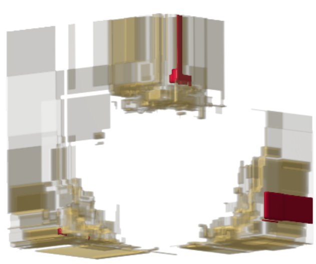
(c) DVR of $\delta_5^{\mathrm{S-L}} = 3/5$



(d) DVR of $\delta_5^{\mathrm{S-L}} = 4/5$



(e) DVR of $\delta_5^{\mathrm{S-L}} = 5/5$



(f) DVR of $\delta_5^{\mathrm{S-L}} \in \{1/5, 5/5\}$

Figure 7.9: DVR of approximated 3D EAF differences between the spherical and linear BASes $\delta_5^{\mathrm{S-L}}$.

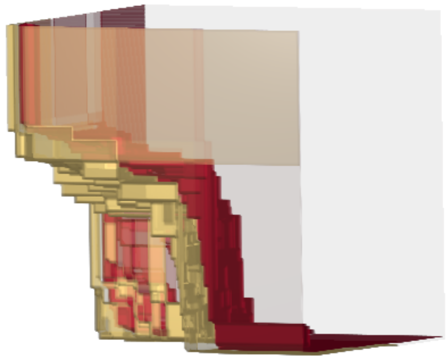


Figure 7.10: DVR of approximated 3D EAF values of the spherical BASes for $\alpha_5^S \in \{1/5, 4/5\}$.

Table 7.1: Summary analysis of methods for visualizing EAFs.

| | Slicing | MIP | DVR |
|---|---|---|---|
| Exact EAF values | + Enables detailed analysis of the approximation sets<br>– Visualizes only one slice at a time | – Not sensible to use since usually a large portion of the objective space has the maximum EAF value | – Not possible to use |
| Exact EAF differences | + Enables detailed analysis of the approximation sets<br>+ Capable of simultaneously visualizing positive and negative differences without overlapping<br>– Visualizes only one slice at a time | + No parameters to set<br>+ Shows all values in a single image<br>– No sense of depth<br>– Feasible only for a limited number of cuboids (impractical) | – Not possible to use |
| Approximated EAF values | – No need to use it on the approximation as it works well for exact EAFs | – Not sensible to use since usually a large portion of the objective space has the maximum EAF value | + Nice visualizations<br>+ Enables looking through the volume and preserves the sense of depth<br>– Requires definition of the transfer function |
| Approximated EAF differences | – No need to use it on the approximation as it works well for exact EAFs | + No parameters to set<br>+ Shows all values in a single image<br>– No sense of depth | + Nice visualizations<br>+ Enables looking through the volume and preserves the sense of depth<br>– Requires definition of the transfer function |

It is not reasonable to use MIP for visualizing EAF values since usually a large portion of the objective space has the maximum EAF value and such plots are not very informative. While MIP can be used for visualizing exact EAF differences it works much better on the approximated case, for which it was conceived. Visualizing 3D cuboids is rather impractical (especially for a very large number of cuboids) and requires sorting of cuboids with regard to their value in order to produce a MIP image. However, MIP can easily be used to visualize approximated EAF differences as it has no parameters to set. Its biggest advantage is that it combines all values in a single image giving precedence to largest differences (which are the most important when visualizing EAF differences), while its biggest disadvantage is the lost sense of depth.

Finally, DVR can be used to visualize approximated EAFs. It produces nice and informative visualizations of both the EAF values and differences. With an insightful setting of the transfer function it is even possible to "look through" the cloud of cuboids. The need to define the transfer function is at the same time the biggest disadvantage of DVR as it might be demanding for a user not familiar with the method. Nevertheless, it is much easier to find the right transfer function for EAFs than, for example, for some medical data, as EAFs take only discrete values.

# Chapter 8

# Steel-Casting Use Case

This chapter shows how the proposed visualization methods can be used on a real-world optimization problem with three objectives from a previous study (Mlakar, Tušar, & Filipič, 2012).

## 8.1 The Steel-Casting Problem

Continuous casting of steel is a complex metallurgical process where molten steel is cooled and shaped into semi-manufactures of desired dimensions. Cooling is done using water in the primary and secondary cooling subsystems. Primary cooling is performed in the mold, while secondary cooling comprehends wreath cooling when the steel exits the mold and spray cooling of the strand before it is cut into billets.

The goal is to set the parameters of this process (casting speed, mold outlet coolant temperature and wreath and spray coolant flows, see Table 8.1) in such a way that the quality of the cast steel is as high as possible. The quality of steel is defined by the following three objectives: distance from desired *metallurgical length* (the length of the liquid core in the strand), distance from desired *shell thickness* (thickness of the solid shell at mold exit) and distance from desired *surface temperature* at a predefined point in the strand. Table 8.2 details the variables defining these optimization objectives. Their bounds and desired values were determined by domain experts.

If a parameter setting yields values outside the boundary constraints presented in Table 8.2, it is deemed infeasible. The objectives of the feasible solutions are computed by

$$f_k = |y_k - y_k^*| \text{ for all } i \in 1, 2, 3, \tag{8.1}$$

where $y_k$ is the value of the observed variable and $y_k^*$ its desired value.

As real-world experimentation with parameter settings is expensive, time-consuming and could also be dangerous, we have simulated it using a numerical model of steel casting

Table 8.1: Steel-casting process parameters.

| Parameter | Lower bound | Upper bound | Discretization step |
|---|---|---|---|
| Casting speed [m/min] | 1.5 | 2.0 | 0.01 |
| Mold outlet coolant temp. [°C] | 33 | 35 | 1 |
| Wreath coolant flow [$m^3$/h] | 10 | 40 | 5 |
| Spray coolant flow [$m^3$/h] | 25 | 65 | 5 |

Table 8.2: Variables defining the optimization objectives.

| Variable | Lower bound | Upper bound | Desired value |
|---|---|---|---|
| Metallurgical length [m] | 10 | 11 | 10 |
| Shell thickness [mm] | 11 | 15 | 13 |
| Surface temperature [°C] | 1115 | 1130 | 1122.5 |

based on a meshless technique for diffusive heat transport (Vertnik & Šarler, 2009). One simulation of the steel-casting process takes approximately 2 minutes on a standard desktop computer.

Two instances of this optimization problem were studied: *discrete* and *continuous*. The discrete problem instance was set by domain experts as presented in Table 8.1 is such a way that all possible solutions (9639 in total) could be explored. While the discrete problem instance enables a rough exploration of the objective space, the continuous problem instance where any value within the variable bounds could be chosen is the one we wish to solve.

## 8.2 Results of Optimization Algorithms

The discrete problem instance was solved using the Exhaustive Search (ES) algorithm. These solutions form the Pareto front of the discrete problem instance and we wanted to see whether an algorithm tackling the continuous problem instance could come close to this Pareto front.

We chose the algorithm DEMO (Differential Evolution for Multiobjective Optimization) (Robič & Filipič, 2005), a MOEA algorithm that uses Differential Evolution (Storn & Price, 1997), to explore the decision space for the continuous problem instance. DEMO was run five times, each time exploring 3200 solutions. More detailed information on the employed experimental setup can be found in (Mlakar et al., 2012).

The majority of the explored solutions were infeasible. For example, out of 9639 solutions found by ES only 1242 were feasible, and of those only 72 mutually nondominated. DEMO was solving an extended problem and therefore found more nondominated solutions—on average, 645 per run (although their number varied significantly over different runs).

First, we visualize all final (feasible and nondominated) solutions found by both algorithms in Figure 8.1. We can see that ES found two distinct subsets of solutions. A more detailed analysis of results revealed that they correspond to two of the three mold outlet coolant temperatures (the remaining one always produces infeasible solutions). Since DEMO was not bound by this discretization, it was able to find solutions also around these subsets. Both algorithms found a few solutions with low distances from desired shell thickness that are rather “detached” from the rest. They actually lie on a larger disconnected region of the Pareto front of which just this minor part is feasible.

This visualization is able to show that DEMO is able to reach the Pareto front of the discrete problem and is therefore a good choice for solving this problem. However, there are two aspects we are interested in that are hard to infer from this visualization alone (they could, of course, be computed from the solutions). First, we wish to see whether different runs of DEMO produce similar results. Visualizing all five approximation sets together with different markers or visualizing one approximation set at a time does not provide a good means for comparison. Second, although the approximation sets found by DEMO

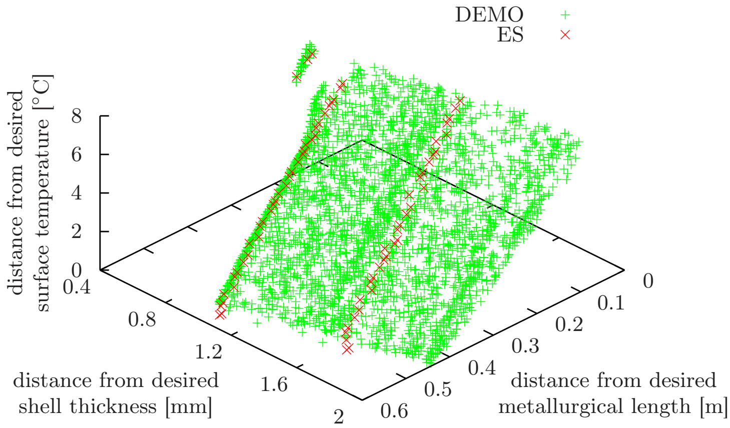


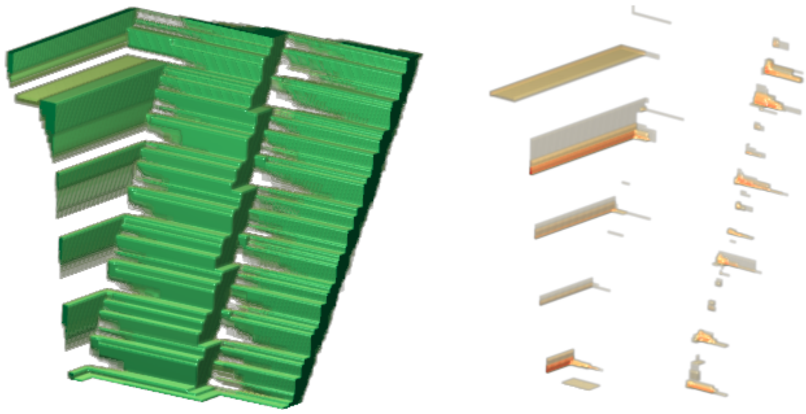

Figure 8.1: The best solutions of the steel-casting problem. The plot presents solutions found by ES on the discrete problem instance and by DEMO on the continuous problem instance (all five runs shown).

look linear, they are in fact slightly convex making it very hard to see (even by rotating the objective space) whether results by ES are in fact dominated by those by DEMO. We will try to find the answers to these questions using the proposed visualization methods.

In order to use the proposed visualization methods to visualize the results of ES and DEMO, we need to address two issues. The first are uneven ranges of the attainment surfaces, which are important when computing voxels. As we can infer from Table 8.2, the observed objective space is equal to $[0, 1] \times [0, 2] \times [0, 7.5]$. Moreover, the feasible results found by the two algorithms actually lie in an even smaller cuboid contained in $[0, 0.6] \times [0.5, 2] \times [0, 7.5]$. If the discretization to voxels was based on these objective ranges, the first two objectives would be discretized too roughly. Therefore, we choose to normalize the objective vectors to $[0, 1]^3$ before performing discretization to $128^3$ voxels. The second issue is the uneven number of runs of both algorithms (ES was run once). To leave the meaning of the EAFs differences intact, we copy the results by ES to get five runs in total. This seems a reasonable approach as the deterministic ES would actually produce five equal results if it was run five times.

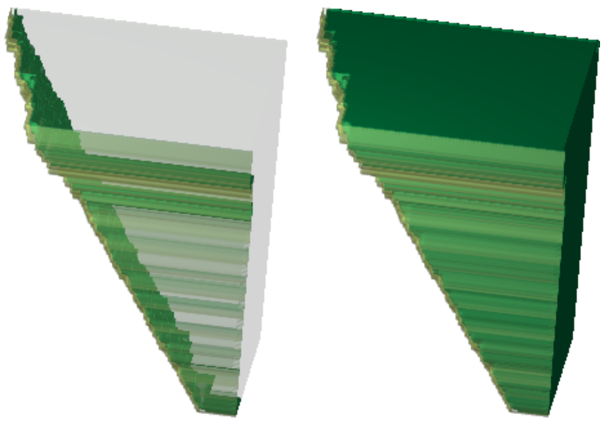
Now, visualization methods as described in Sections 7.2 and 7.3 can be used on these results. Figure 8.2 presents the DVR of approximated EAF differences between the two algorithms, $\delta_5^{\mathrm{DEMO-ES}}$ and $\delta_5^{\mathrm{ES-DEMO}}$, which gives us a general idea of their performance. However, this does not answer our two questions. To check the repeatability of DEMO, we need to inspect the difference between the best and worst summary attainment surfaces. The narrow layer between $\alpha_5^{\mathrm{DEMO}} = 1/5$ and $\alpha_5^{\mathrm{DEMO}} = 4/5$ as well as $\alpha_5^{\mathrm{DEMO}} = 1/5$ and $\alpha_5^{\mathrm{DEMO}} = 5/5$ (see Figure 8.3) suggests that although DEMO's approximation sets were of considerably different cardinality, they attain a similar portion of the objective space. This is further confirmed by a more detailed inspection using Slicing at angles $\varphi = 25°$ and $45°$ (see Figure 8.4), which shows that only a small border of the attained objective space is attained less than five times (denoted by light green hues).

Finally, MIP can be used to check whether DEMO ever finds better solutions than those found by ES. If this was not the case, each solution found by ES would have exactly one union of cuboids (albeit small) for which $\delta_5^{\mathrm{ES-DEMO}} = 5/5$. As accuracy is important in this case (small cuboids can be omitted if the discretization into voxels is not fine enough), we use MIP on exact EAF differences. Figure 8.5 clearly shows that although $\delta_5^{\mathrm{ES-DEMO}} = 5/5$ for some solutions, this does not hold for all of them, meaning that



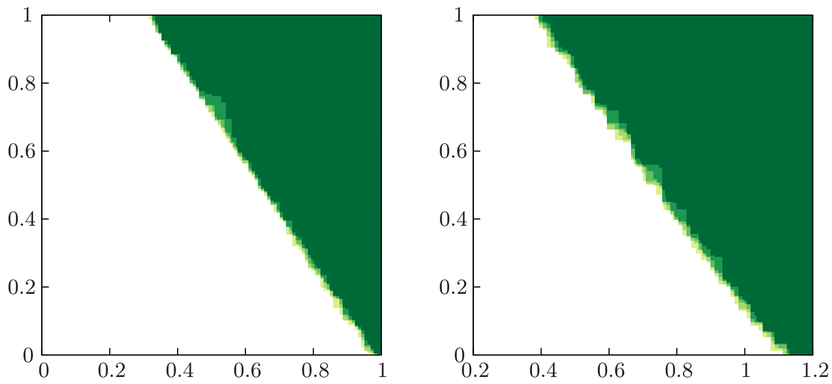
(a) DVR of $\delta_5^{\mathrm{DEMO-ES}}$ (b) DVR of $\delta_5^{\mathrm{ES-DEMO}}$

Figure 8.2: DVR of approximated 3D EAF differences between the two algorithms. Using transparencies, all values in $\{1/5, \ldots, 5/5\}$ are shown.



(a) DVR of $\alpha_5^{\mathrm{DEMO}} \in \{1/5, 4/5\}$ (b) DVR of $\alpha_5^{\mathrm{DEMO}} \in \{1/5, 5/5\}$

Figure 8.3: DVR of approximated 3D EAF values of DEMO.




(a) Slice of $\alpha_5^{\mathrm{DEMO}}$ at $\varphi = 25°$ (b) Slice of $\alpha_5^{\mathrm{DEMO}}$ at $\varphi = 45°$

Figure 8.4: Slices of exact 3D EAF values of DEMO at two angles.

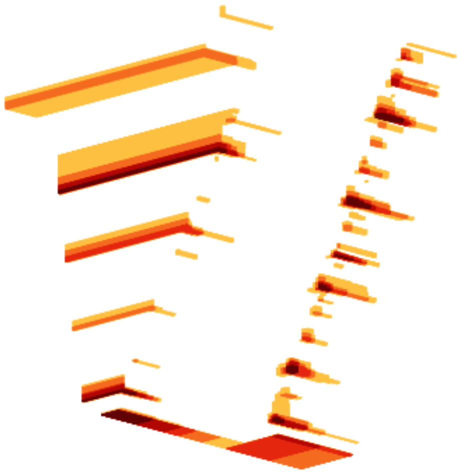


Figure 8.5: MIP of exact 3D EAF differences between ES and DEMO.

DEMO was actually able to find solutions that dominate those by ES. This was of course possible only because DEMO was solving an extended instance of the problem.

# Chapter 9

# Conclusions

This final chapter presents the concluding remarks. A summary of the thesis is given in Section 9.1, while Section 9.2 recalls its original contributions. The chapter ends with future work ideas in Section 9.3.

## 9.1 Summary

### 9.1.1 Visualization of Approximation Sets

Visualization of Pareto front approximations has different requirements than visualization of other multidimensional data. We are interested not only in the distribution of vectors in the objective space, but also in the dominance relations between them (to be able to compare different approximation sets) and in the shape of approximation sets (we wish to see their knees, discontinuities, etc.). Moreover, visualization methods need to handle large approximations sets as the sets found by MOEAs are usually large. To inspect how visualization methods comply with these requirements, we have introduced two novel 4D benchmark approximation sets, one linear with a uniform distribution of vectors and the other spherical with a nonuniform distribution of vectors. They are close together in the objective space, but have a different shape and distribution of vectors. Therefore, a good visualization method should be able to recognize their features and differentiate well between them.

We have shown the visualizations of the existing methods on the two benchmark approximation sets and summarized their properties in a table. Most of the presented methods are scalable to any number of objectives, but fail to correctly show the dominance relations between vectors, the approximation set shape or the distribution of vectors. Moreover, some are not suitable for comparing two or more approximation sets and face difficulties when visualizing large sets.

The presented visualization with prosections, which performs dimension reduction for all vectors within the chosen section, has just the opposite properties: it is able to correctly show the dominance relations between many vectors, the approximation set shape and the distribution of vectors, but is not easily scalable to more than 4D. In addition, it can handle multiple and large approximation sets while being robust and computationally inexpensive. Because of this, prosections are best used to study the quality of convergence to the Pareto front (if known), to compare different MOEAs and visualize the progress of a single MOEA. This was demonstrated on some well-known multiobjective optimization problems as well a problem with knees and problems with redundant objectives, while the partial preservation of the dominance relation and the relative closeness to reference vectors was formally proven. This means that a thorough investigation of the first hypothesis from

Section 1.2 gave no evidence against it.

This thesis tackled visualization of approximation sets in a somehow limited scope as only four objectives were considered. However, in our opinion it is important to be able to first understand and really *see* the 4D approximation sets before moving on to more than four dimensions. Also, while the step from 3D to 4D might seem small, it is significant. Most standard MOEAs that work well in 2D and 3D fail to reach good results in 4D. Being able to visualize their outcome on 4D problems gives researchers a powerful tool for finding pitfalls and improving the performance of these algorithms on 4D optimization problems.

### 9.1.2 Visualization of Empirical Attainment Functions

The EAF can be used to describe how well one or more algorithms attain the objective space with their multiple approximation sets. While some aspects of visualization of EAFs are similar to those of approximation sets (for example, we are interested to see the shapes of approximation sets, we must be able to visualize large sets, etc.), the two tasks are very different. Visualization of EAFs means visualization of volumetric data, not just sets of vectors, and is therefore even more demanding.

While visualization of EAFs is rather straightforward in 2D, it presents a challenge in 3D as multiple cuboids need to be visualized. We have presented how these cuboids can be computed and visualized using Slicing and MIP. Slicing visualizes rectangles obtained by "cutting" through the cuboids at a chosen angle. This gives an accurate, but limited visualization of the cuboids. On the other hand, MIP performs a projection of the cuboids that gives precedence to those with higher values. While the result provides an overview of the most "important" cuboids in a single image, all sense of depth is lost. If accuracy of visualization is not crucial, the EAF values and differences can be approximated by discretizing the objective space into a grid of voxels. In this way, DVR can be used for visualization of EAFs in addition to Slicing and MIP. DVR uses a transfer function to assign color and opacity to voxel values. With the correct setting of the transfer function we can "look through" the cloud of cuboids. We have shown how Slicing, MIP and DVR perform on two sets of benchmark approximation sets and discussed their advantages and disadvantages.

In addition, we demonstrated the use of all three methods on a real-world optimization problem solved by exhaustive search and a MOEA algorithm. We have shown that these powerful visualization methods are able to give a new insight regarding the performance of the algorithms that cannot be otherwise seen using solely "standard" visualization of approximation sets. In summary, no evidence against the second hypothesis from Section 1.2 was found.

## 9.2 Original Contributions

Let us recall the original scientific contributions of this thesis:

- The design of benchmark approximation sets, not yet existing in the literature, which can provide a basis for establishing an evaluation and comparison methodology for visualization methods. Also, the use of these sets in a critical review of existing methods for visualizing approximation sets.

- A novel visualization method capable of simultaneously visualizing several 4D approximation sets while preserving the Pareto dominance relation between numerous vectors. Such a visualization method enables analysis and straightforward comparison of different 4D MOEAs, which has not been satisfactorily handled so far. In addition, some of the properties of this visualization method (partial preservation of the dominance relation and relative closeness to reference vectors) can be formally proven.

- The first attempt at visualizing 3D EAF values and differences, which covers the exact as well as the approximated case.

In both tasks (visualization of approximation sets and visualization of EAFs) we have made a step forward from the current state of the art. In visualization of approximation sets, we have provided a method that favors accuracy over scalability and can truthfully visualize 4D approximation sets in 3D in an intuitive way. In visualization of EAFs, we made the transition from visualizing EAF values and differences from 2D to 3D.

## 9.3 Future work

The most important future work direction is to supply a software that implements the presented visualization methods. While we have provided `gnuplot` scripts that can be used to visualize prosections, a dedicated software that would allow visualization of either single prosections or prosection matrices together with interactive setting of their parameters would be more attractive for potential users. Moreover, this software should be capable of visualizing exact EAFs using Slicing and MIP. While it could also perform the discretization to voxels, dedicated volume rendering software should be then used to visualize approximated EAFs.

Regarding further research ideas, we are interested in exploring how prosections can be extended beyond 4D. While at the first glance this seems straightforward—just apply prosection twice—we wish to find a way that will retain all the good properties of 4D prosections while at the same time not introduce new parameters to keep the method as simple and manageable as possible. One way to do this would be to construct some kind of a "recommendation function" that would provide a ranking of views with regard to their importance to the user. This ranking could be based on the properties of the approximation sets to be visualized. In this way, the user would not have to set any parameters of the method or look at the prosection matrix, but could simply visualize the top recommended views.

Finally, we wish to find a more efficient way of computing the cuboids either by improving the provided algorithms or by adjusting efficient algorithms for hypervolume calculation to suit our needs.

# Appendix A

# Proofs of Theorems

In the proofs we denote with $ijk_1 \dots k_{m-2}$ the permutation of indices $1, \dots, m$ so that $k_1 < \dots < k_{m-2}$ and use the following abbreviations:

$$\begin{aligned} s^{\mathrm{A}} &= s_{\varphi,d,\boldsymbol{a}}(z_i^{\mathrm{A}}, z_j^{\mathrm{A}}) = (z_i^{\mathrm{A}} - a_i)\cos\varphi + (z_j^{\mathrm{A}} - a_j)\sin\varphi \\ s^{\mathrm{B}} &= s_{\varphi,d,\boldsymbol{a}}(z_i^{\mathrm{B}}, z_j^{\mathrm{B}}) = (z_i^{\mathrm{B}} - a_i)\cos\varphi + (z_j^{\mathrm{B}} - a_j)\sin\varphi \end{aligned} \tag{A.1}$$

**Theorem 5.1.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \geq 2$ and $\varphi \in (0°, 90°)$. Then for any two objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \dots, z_m^{\mathrm{A}})$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \dots, z_m^{\mathrm{B}})$ inside the section the following holds:

$$\boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}} \Rightarrow \sigma(\boldsymbol{z}^{\mathrm{A}}) \prec \sigma(\boldsymbol{z}^{\mathrm{B}}). \tag{5.9}$$

*Proof.* First, because $\boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}}$, it follows that $z_i^{\mathrm{A}} \leq z_i^{\mathrm{B}}$, $z_j^{\mathrm{A}} \leq z_j^{\mathrm{B}}$ and $z_{k_l}^{\mathrm{A}} \leq z_{k_l}^{\mathrm{B}}$ for $l \in \{1, \dots, m-2\}$. Also, because $\varphi \in (0°, 90°)$, $\sin\varphi > 0$ and $\cos\varphi > 0$. Therefore

$$s^{\mathrm{B}} - s^{\mathrm{A}} = \underbrace{(z_i^{\mathrm{B}} - z_i^{\mathrm{A}})}_{\geq 0}\underbrace{\cos\varphi}_{>0} + \underbrace{(z_j^{\mathrm{B}} - z_j^{\mathrm{A}})}_{\geq 0}\underbrace{\sin\varphi}_{>0} \geq 0. \tag{A.2}$$

This means that $\sigma(\boldsymbol{z}^{\mathrm{A}}) \preceq \sigma(\boldsymbol{z}^{\mathrm{B}})$. Now we only have to prove that $\sigma(\boldsymbol{z}^{\mathrm{A}}) \neq \sigma(\boldsymbol{z}^{\mathrm{B}})$. If there exists an index $l$ so that $z_{k_l}^{\mathrm{A}} < z_{k_l}^{\mathrm{B}}$, then $\sigma(\boldsymbol{z}^{\mathrm{A}}) \neq \sigma(\boldsymbol{z}^{\mathrm{B}})$. Otherwise, if $z_{k_l}^{\mathrm{A}} = z_{k_l}^{\mathrm{B}}$ for all $l \in \{1, \dots, m-2\}$, then either $z_i^{\mathrm{A}} < z_i^{\mathrm{B}}$ or $z_j^{\mathrm{A}} < z_j^{\mathrm{B}}$. In either case this means that $s^{\mathrm{B}} - s^{\mathrm{A}} > 0$, which proves the theorem. □

**Theorem 5.2.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \geq 2$ and $\varphi \in (0°, 90°)$. Then for any two objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \dots, z_m^{\mathrm{A}})$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \dots, z_m^{\mathrm{B}})$ inside the section the following holds:

$$\sigma(\boldsymbol{z}^{\mathrm{A}}) \prec \sigma(\boldsymbol{z}^{\mathrm{B}}) \wedge s_{\varphi,d,\boldsymbol{a}}(z_i^{\mathrm{B}}, z_j^{\mathrm{B}}) - s_{\varphi,d,\boldsymbol{a}}(z_i^{\mathrm{A}}, z_j^{\mathrm{A}}) \geq 2d \max\left\{\tan\varphi, \tan^{-1}\varphi\right\} \Rightarrow \boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}}. \tag{5.10}$$

*Proof.* First, let us show that under the assumptions of the theorem $\boldsymbol{z}^{\mathrm{A}} \preceq \boldsymbol{z}^{\mathrm{B}}$. Because $\sigma(\boldsymbol{z}^{\mathrm{A}}) \prec \sigma(\boldsymbol{z}^{\mathrm{B}})$, it follows that $z_{k_l}^{\mathrm{A}} \leq z_{k_l}^{\mathrm{B}}$ for $l \in \{1, \dots, m-2\}$. We need to show that $z_i^{\mathrm{A}} \leq z_i^{\mathrm{B}}$ and $z_j^{\mathrm{A}} \leq z_j^{\mathrm{B}}$.

If $d > 0$, $(z_i^{\mathrm{A}}, z_j^{\mathrm{A}})$ is not the only vector to be projected into the value $s^A$. In fact, the whole line segment

$$z_j = a_j - \frac{z_i - a_i}{\tan\varphi} + \frac{s^A}{\sin\varphi}, \tag{A.3}$$

where

$$\begin{aligned} z_i &\in [a_i + s^A\cos\varphi - d\sin\varphi, a_i + s^A\cos\varphi + d\sin\varphi], \\ z_j &\in [a_j + s^A\sin\varphi - d\cos\varphi, a_j + s^A\sin\varphi + d\cos\varphi], \end{aligned} \tag{A.4}$$

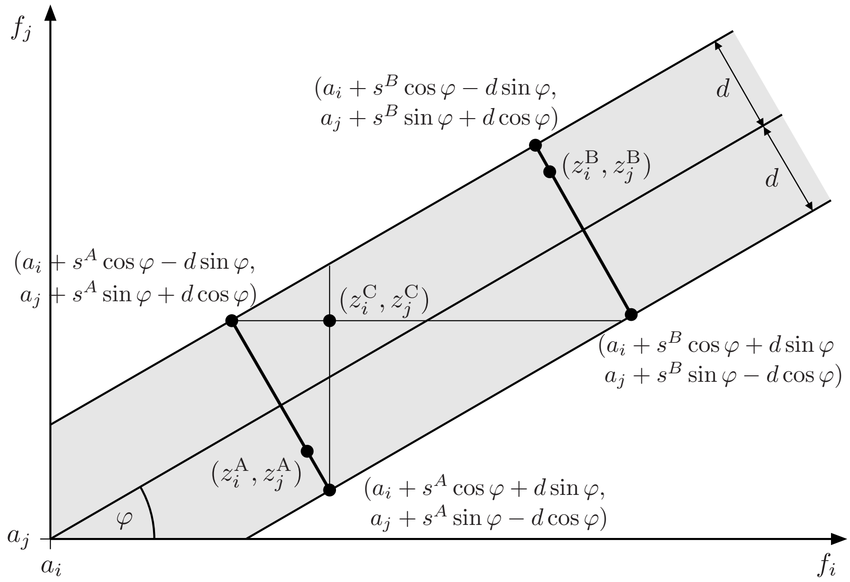



Figure A.1: Line segments $A$ and $B$.

is projected into the same value $s^A$ (see Figure A.1). We will denote this as line segment $A$. Analogously, the whole line segment

$$z_j = a_j - \frac{z_i - a_i}{\tan\varphi} + \frac{s^B}{\sin\varphi}, \tag{A.5}$$

where

$$\begin{aligned} z_i &\in [a_i + s^B \cos\varphi - d\sin\varphi, a_i + s^B \cos\varphi + d\sin\varphi], \\ z_j &\in [a_j + s^B \sin\varphi - d\cos\varphi, a_j + s^B \sin\varphi + d\cos\varphi], \end{aligned} \tag{A.6}$$

is projected into the value $s^B$. This is line segment $B$. Note that if $d = 0$, the line segments $A$ and $B$ consist of only one vector each, which are equal to $(a_i + s^A \cos\varphi, a_j + s^A \sin\varphi)$ and $(a_i + s^B \cos\varphi, a_j + s^B \sin\varphi)$, respectively.

Showing that all vectors from line segment $A$ weakly dominate the whole line segment $B$ proves that $z_i^{\mathrm{A}} \leq z_i^{\mathrm{B}}$ and $z_j^{\mathrm{A}} \leq z_j^{\mathrm{B}}$. Since the weak dominance relation is transitive, this can be done in two steps:

1. Line segment $A$ weakly dominates a vector $(z_i^{\mathrm{C}}, z_j^{\mathrm{C}})$.

2. The vector $(z_i^{\mathrm{C}}, z_j^{\mathrm{C}})$ weakly dominates line segment $B$.

We can show that this holds for the vector

$$(z_i^{\mathrm{C}}, z_j^{\mathrm{C}}) = (a_i + s^A \cos\varphi + d\sin\varphi, a_j + s^A \sin\varphi + d\cos\varphi). \tag{A.7}$$

*Proof of step 1:* It is trivial to see that all vectors from the line segment $A$ weakly dominate the vector $(z_i^{\mathrm{C}}, z_j^{\mathrm{C}})$.

*Proof of step 2:* The vector $(z_i^{\mathrm{C}}, z_j^{\mathrm{C}})$ weakly dominates the line segment $B$ when the following two inequalities hold:

$$\begin{aligned} z_i^{\mathrm{C}} &\leq a_i + s^B \cos\varphi - d\sin\varphi \\ z_j^{\mathrm{C}} &\leq a_j + s^B \sin\varphi - d\cos\varphi \end{aligned} \tag{A.8}$$

$$\begin{aligned} z_i^{\mathrm{C}} &\le a_i + s^B\cos\varphi - d\sin\varphi \\ a_i + s^A\cos\varphi + d\sin\varphi &\le a_i + s^B\cos\varphi - d\sin\varphi \\ (s^B - s^A)\cos\varphi &\ge 2d\sin\varphi \\ s^B - s^A &\ge 2d\tan\varphi \end{aligned} \tag{A.9}$$

$$\begin{aligned} z_j^{\mathrm{C}} &\le a_j + s^B\sin\varphi - d\cos\varphi \\ a_j + s^A\sin\varphi + d\cos\varphi &\le a_j + s^B\sin\varphi - d\cos\varphi \\ (s^B - s^A)\sin\varphi &\ge 2d\cos\varphi \\ s^B - s^A &\ge 2d\tan^{-1}\varphi \end{aligned} \tag{A.10}$$

Both inequalities hold because of the condition from the theorem:

$$s^B - s^A \ge 2d\max\left\{\tan\varphi, \tan^{-1}\varphi\right\}. \tag{A.11}$$

Now we only need to show that $\boldsymbol{z}^{\mathrm{A}} \ne \boldsymbol{z}^{\mathrm{B}}$. If there exists an index $l$ so that $z_{k_l}^{\mathrm{A}} < z_{k_l}^{\mathrm{B}}$, then $\boldsymbol{z}^{\mathrm{A}} \ne \boldsymbol{z}^{\mathrm{B}}$. Otherwise, if $z_{k_l}^{\mathrm{A}} = z_{k_l}^{\mathrm{B}}$ for all $l \in \{1, \ldots, m-2\}$, then $s^{\mathrm{A}} < s^{\mathrm{B}}$. Since $\varphi \in (0°, 90°)$, $\sin\varphi > 0$ and $\cos\varphi > 0$. Therefore,

$$s^{\mathrm{B}} - s^{\mathrm{A}} = \underbrace{(z_i^{\mathrm{B}} - z_i^{\mathrm{A}})}_{\ge 0}\underbrace{\cos\varphi}_{>0} + \underbrace{(z_j^{\mathrm{B}} - z_j^{\mathrm{A}})}_{\ge 0}\underbrace{\sin\varphi}_{>0} > 0 \tag{A.12}$$

Because of this, $(z_i^{\mathrm{B}} - z_i^{\mathrm{A}})$ and $(z_j^{\mathrm{B}} - z_j^{\mathrm{A}})$ cannot be 0 at the same time, which means that $\boldsymbol{z}^{\mathrm{A}} \prec \boldsymbol{z}^{\mathrm{B}}$. □

**Theorem 5.3.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \ge 2$. Then for any two objective vectors $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}})$ and $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}})$ inside the section the following holds:

$$\left\|\sigma(\boldsymbol{z}^{\mathrm{A}}) - \sigma(\boldsymbol{z}^{\mathrm{B}})\right\| \le \left\|\boldsymbol{z}^{\mathrm{A}} - \boldsymbol{z}^{\mathrm{B}}\right\|. \tag{5.12}$$

The equality holds iff

$$\frac{z_j^{\mathrm{A}} - z_j^{\mathrm{B}}}{z_i^{\mathrm{A}} - z_i^{\mathrm{B}}} = \tan\varphi. \tag{5.13}$$

*Proof.* Let us first provide the proof for (5.12).

$$\begin{aligned} &\left\|\boldsymbol{z}^{\mathrm{A}} - \boldsymbol{z}^{\mathrm{B}}\right\|^2 - \left\|\sigma(\boldsymbol{z}^{\mathrm{A}}) - \sigma(\boldsymbol{z}^{\mathrm{B}})\right\|^2 = \\ &= (z_1^{\mathrm{A}} - z_1^{\mathrm{B}})^2 + \cdots + (z_m^{\mathrm{A}} - z_m^{\mathrm{B}})^2 - \\ &- ((s^{\mathrm{A}} - s^{\mathrm{B}})^2 + (z_{k_1}^{\mathrm{A}} - z_{k_1}^{\mathrm{B}})^2 + \cdots + (z_{k_{m-2}}^{\mathrm{A}} - z_{k_{m-2}}^{\mathrm{B}})^2) = \\ &= (z_i^{\mathrm{A}} - z_i^{\mathrm{B}})^2 + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})^2 - (s^{\mathrm{A}} - s^{\mathrm{B}})^2 = \\ &= (z_i^{\mathrm{A}} - z_i^{\mathrm{B}})^2 + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})^2 - \left((z_i^{\mathrm{A}} - z_i^{\mathrm{B}})\cos\varphi + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})\sin\varphi\right)^2 = \\ &= ((z_i^{\mathrm{A}} - z_i^{\mathrm{B}})^2 + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})^2)(\sin^2\varphi + \cos^2\varphi) - \left((z_i^{\mathrm{A}} - z_i^{\mathrm{B}})\cos\varphi + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})\sin\varphi\right)^2 = \\ &= (z_i^{\mathrm{A}} - z_i^{\mathrm{B}})^2\sin^2\varphi + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})^2\sin^2\varphi + (z_i^{\mathrm{A}} - z_i^{\mathrm{B}})^2\cos^2\varphi + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})^2\cos^2\varphi - \\ &- (z_i^{\mathrm{A}} - z_i^{\mathrm{B}})^2\cos^2\varphi - 2(z_i^{\mathrm{A}} - z_i^{\mathrm{B}})(z_j^{\mathrm{A}} - z_j^{\mathrm{B}})\sin\varphi\cos\varphi - (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})^2\sin^2\varphi = \\ &= (z_i^{\mathrm{A}} - z_i^{\mathrm{B}})^2\sin^2\varphi + (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})^2\cos^2\varphi - 2(z_i^{\mathrm{A}} - z_i^{\mathrm{B}})(z_j^{\mathrm{A}} - z_j^{\mathrm{B}})\sin\varphi\cos\varphi = \\ &= ((z_i^{\mathrm{A}} - z_i^{\mathrm{B}})\sin\varphi - (z_j^{\mathrm{A}} - z_j^{\mathrm{B}})\cos\varphi)^2 \ge 0 \end{aligned} \tag{A.13}$$

This means that $\left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{z}^{\mathrm{B}}\right\|^2 \geq \left\|\sigma(\boldsymbol{z}^{\mathrm{A}})-\sigma(\boldsymbol{z}^{\mathrm{B}})\right\|^2$, from which it follows that $\left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{z}^{\mathrm{B}}\right\| \geq \left\|\sigma(\boldsymbol{z}^{\mathrm{A}})-\sigma(\boldsymbol{z}^{\mathrm{B}})\right\|$.

The equality (5.13) holds when

$$(z_i^{\mathrm{A}}-z_i^{\mathrm{B}})\sin\varphi-(z_j^{\mathrm{A}}-z_j^{\mathrm{B}})\cos\varphi=0, \tag{A.14}$$

which is equivalent to

$$\frac{z_j^{\mathrm{A}}-z_j^{\mathrm{B}}}{z_i^{\mathrm{A}}-z_i^{\mathrm{B}}}=\tan\varphi. \tag{A.15}$$

□

**Theorem 5.4.** Suppose the $\sigma = m\mathrm{D}(\boldsymbol{a}, f_i f_j, \varphi, d)$ prosection is performed, where $m \geq 2$. Let $\boldsymbol{z}^{\mathrm{A}} = (z_1^{\mathrm{A}}, \ldots, z_m^{\mathrm{A}})$, $\boldsymbol{z}^{\mathrm{B}} = (z_1^{\mathrm{B}}, \ldots, z_m^{\mathrm{B}})$ and $\boldsymbol{r} = (r_1, \ldots, r_m)$ be three vectors inside the section and let us assume that $\left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\| < \left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|$. Then

$$\left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|^2-\left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|^2>4d^2 \Rightarrow \left\|\sigma(\boldsymbol{z}^{\mathrm{A}})-\sigma(\boldsymbol{r})\right\|<\left\|\sigma(\boldsymbol{z}^{\mathrm{B}})-\sigma(\boldsymbol{r})\right\|. \tag{5.14}$$

On the other hand,

$$\left\|\sigma(\boldsymbol{z}^{\mathrm{B}})-\sigma(\boldsymbol{r})\right\|^2-\left\|\sigma(\boldsymbol{z}^{\mathrm{A}})-\sigma(\boldsymbol{r})\right\|^2>4d^2 \Rightarrow \left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|<\left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|. \tag{5.15}$$

*Proof.* First, note that for any vector $\boldsymbol{z}$ in the section the following holds:

$$\|\sigma(\boldsymbol{z})-\sigma(\boldsymbol{r})\|^2+4d^2 \geq \|\boldsymbol{z}-\boldsymbol{r}\|^2 \geq \|\sigma(\boldsymbol{z})-\sigma(\boldsymbol{r})\|^2. \tag{A.16}$$

Let us provide the proof for (5.14). From (A.16) and the assumption in the theorem it follows that

$$\left\|\sigma(\boldsymbol{z}^{\mathrm{B}})-\sigma(\boldsymbol{r})\right\|^2+4d^2 \geq \left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|^2 > \left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|^2+4d^2. \tag{A.17}$$

Therefore $\left\|\sigma(\boldsymbol{z}^{\mathrm{B}})-\sigma(\boldsymbol{r})\right\| > \left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|$.

The second affirmation (5.15) can be proven in a similar way using (A.16) and the assumption in the theorem:

$$\left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\|^2 \geq \left\|\sigma(\boldsymbol{z}^{\mathrm{B}})-\sigma(\boldsymbol{r})\right\|^2 > \left\|\sigma(\boldsymbol{z}^{\mathrm{A}})-\sigma(\boldsymbol{r})\right\|^2+4d^2 \geq \left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|^2. \tag{A.18}$$

This means that $\left\|\boldsymbol{z}^{\mathrm{B}}-\boldsymbol{r}\right\| > \left\|\boldsymbol{z}^{\mathrm{A}}-\boldsymbol{r}\right\|$. □

# References

Agrawal, G., Bloebaum, C. L., & Lewis, K. (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, Austin, TX, USA* (1813:1–1813:14). American Institute of Aeronautics and Astronautics.

Agrawal, G., Lewis, K., Chugh, K., Huang, C.-H., Parashar, S., & Bloebaum, C. L. (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, Albany, NY, USA* (4434:1–4434:11). American Institute of Aeronautics and Astronautics.

Ang, K. H., Chong, G., & Li, Y. (2002). Visualization technique for analyzing non-dominated set comparison. In L. Wang (Ed.), *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning, SEAL'02, Singapore* (pp. 36–40). Singapore: Nanyang Technological University.

Ashby, M. F. (2000). Multi-objective optimization in material design and selection. *Acta Materialia*, *48*(1), 359–369.

Beume, N., Fonseca, C. M., López-Ibáñez, M., Paquete, L., & Vahrenhold, J. (2009). On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation*, *13*(5), 1075–1081.

Beume, N., Naujoks, B., & Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, *181*(3), 1653–1669.

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

Borg, I. & Groenen, P. J. F. (2005). *Modern multidimensional scaling: Theory and applications*. New York, NY, USA: Springer.

Botha, C. P. & Post, F. H. (2002). New technique for transfer function specification in direct volume rendering using real-time visual feedback. In S. K. Mun (Ed.), *Proceedings of the SPIE international symposium on medical imaging, San Diego, CA, USA* (Vol. 4681: Visualization, Image-Guided Procedures and Display, pp. 349–356). Society of Photo-Optical Instrumentation Engineers.

Branke, J., Deb, K., Dierolf, H., & Oswald, M. (2004). Finding knees in multi-objective optimization. In X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. M. Guervós, J. A. Bullinaria, … H.-P. Schwefel (Eds.), *Proceedings of the 8th international conference on parallel problem solving from nature, PPSN VIII, Birmingham, UK* (Vol. 3242, pp. 722–731). Lecture Notes in Computer Science. Springer.

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

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms.* Chichester, UK: John Wiley & Sons.

Deb, K. (2012). Advances in evolutionary multi-objective optimization. In G. Fraser & J. Teixeira de Souza (Eds.), *Search based software engineering* (Vol. 7515, pp. 1–26). Lecture Notes in Computer Science. Springer.

Deb, K. & Jain, H. (2013). An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints. *IEEE Transactions on Evolutionary Computation.* doi:10.1109/TEVC.2013.2281535

Deb, K. & Jain, S. (2002). Running performance metrics for evolutionary multi-objective optimization. In L. Wang (Ed.), *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning, SEAL'02, Singapore* (pp. 13–20). Singapore: Nanyang Technological University.

Deb, K., Pratap, A., Agrawal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197.

Deb, K. & Saxena, D. K. (2006). Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the IEEE congress on evolutionary computation, CEC 2006, Vancouver, BC, Canada* (pp. 3352–3360). IEEE.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable test problems for evolutionary multi-objective optimization. In A. Abraham, R. Jain, & R. Goldberg (Eds.), *Evolutionary multiobjective optimization: Theoretical advances and applications* (pp. 105–145). Advanced Information and Knowledge Processing. Springer.

Díaz, J. & Alcocer, P. P. V. (2010). Depth-enhanced maximum intensity projection. In R. Westermann & G. Kindlmann (Eds.), *Proceedings of the 8th IEEE/EG international symposium on volume graphics, VG2010, Norrköping, Sweden* (pp. 93–100). IEEE.

dos Santos, S. R. (2004). *A framework for the visualization of multidimensional and multivariate data* (Doctoral dissertation, University of Leeds, UK).

Engel, K., Hadwiger, M., Kniss, J. M., Rezk-Salama, C., & Weiskopf, D. (2006). *Real-time volume graphics.* Natick, MA, USA: A. K. Peters.

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

Fonseca, C. M., Guerreiro, A. P., López-Ibáñez, M., & Paquete, L. (2011). On the computation of the empirical attainment function. In R. H. C. Takahashi, K. Deb, E. F. Wanner, & S. Greco (Eds.), *Proceedings of the 6th international conference on evolutionary multi-criterion optimization, EMO 2011, Ouro Preto, Brazil* (Vol. 6576, pp. 106–120). Lecture Notes in Computer Science. Springer.

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

Grunert da Fonseca, V. D., Fonseca, C. M., & Hall, A. O. (2001). Inferential performance assessment of stochastic optimisers and the attainment function. In E. Zitzler, K. Deb, L. Thiele, C. A. Coello Coello, & D. Corne (Eds.), *Proceedings of the first international conference on evolutionary multi-criterion optimization, EMO 2001, Zurich, Switzerland* (Vol. 1993, pp. 213–225). Lecture Notes in Computer Science. Springer.

Heidrich, W., McCool, M., & Stevens, J. (1995). Interactive maximum projection volume rendering. In *Proceedings of the IEEE conference on visualization* (pp. 11–18). Washington, DC, USA: IEEE.

Hoffman, P. E., Grinstein, G. G., Marx, K., Grosse, I., & Stanley, E. (1997). DNA visual and analytic data mining. In *Proceedings of the IEEE conference on visualization* (pp. 437–441). Los Alamitos, CA, USA: IEEE.

Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, *10*(5), 477–506.

Inselberg, A. (2009). *Parallel coordinates: Visual multidimensional geometry and its applications*. New York, NY, USA: Springer.

Ishibuchi, H., Akedo, N., & Nojima, Y. (2011). A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In N. Krasnogor & P. L. Lanzi (Eds.), *Proceedings of the genetic and evolutionary computation conference, GECCO '11, Dublin, Ireland* (pp. 649–656). ACM.

Khare, V. R., Yao, X., & Deb, K. (2003). Performance scaling of multi-objective evolutionary algorithms. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, & L. Thiele (Eds.), *Proceedings of the 2nd international conference on evolutionary multi-criterion optimization, EMO 2003, Faro, Portugal* (Vol. 2632, pp. 376–390). Lecture Notes in Computer Science. Springer.

Knowles, J. (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* (pp. 552–557). Washington, DC, USA: IEEE.

Knowles, J., Corne, D., & Deb, K. (2008). Introduction: Problem solving, EC and EMO. In J. Knowles, D. Corne, & K. Deb (Eds.), *Multiobjective problem solving from nature: From concepts to applications* (pp. 1–28). Natural Computing Series. Springer.

Kohonen, T. (2001). *Self-organizing maps*. Springer Series in Information Sciences. Berlin, Germany: Springer-Verlag.

Köppen, M. & Yoshida, K. (2007). Visualization of Pareto-sets in evolutionary multi-objective optimization. In *Proceedings of the 7th international conference on hybrid intelligent systems, HIS 2007* (pp. 156–161). IEEE.

Kudo, F. & Yoshikawa, T. (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, Brisbane, Australia* (pp. 1–6). IEEE.

Levoy, M. (1988). Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, *8*(3), 29–37.

Liebscher, M., Witowski, K., & Goel, T. (2009). Decision making in multi-objective optimization for industrial applications – Data mining and visualization of Pareto data. In *Proceedings of the 8th world congress on structural and multi-disciplinary optimization, Lisbon, Portugal*.

López-Ibáñez, M., Paquete, L., & Stützle, T. (2010). Exploratory analysis of stochastic local search algorithms in biobjective optimization. In T. Bartz-Beielstein, M. Chiarandini, L. Paquete, & M. Preuss (Eds.), *Experimental methods for the analysis of optimization algorithms* (pp. 209–222). Springer.

López-Ibáñez, M. & Stützle, T. (2012). An experimental analysis of design choices of multi-objective ant colony optimization algorithms. *Swarm Intelligence*, *6*(3), 207–232.

Lotov, A. V., Bushenkov, V. A., & Kamenev, G. K. (2004). *Interactive decision maps: Approximation and visualization of Pareto frontier*. Boston, MA, USA: Kluwer Academic Publishers.

Lotov, A. V. & Miettinen, K. (2008). Visualizing the Pareto frontier. In J. Branke, K. Deb, K. Miettinen, & R. Slowinski (Eds.), *Multiobjective optimization* (Vol. 5252, pp. 213–243). Lecture Notes in Computer Science. Springer.

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

Martin, A. R. & Ward, M. O. (1995). High dimensional brushing for interactive exploration of multivariate data. In *Proceedings of the 6th IEEE visualization conference, Atlanta, GA, USA* (pp. 271–278). IEEE.

Max, N. (1995). Optical models for direct volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, *1*(2), 99–108.

Metzger, W. (2006). *Laws of seeing*. Cambridge, MA, USA: The MIT Press.

Meyer-Spradow, J., Ropinski, T., Mensmann, J., & Hinrichs, K. H. (2009). Voreen: A rapid-prototyping environment for ray-casting-based volume visualizations. *IEEE Computer Graphics and Applications*, *29*(6), 6–13.

Miettinen, K. (2014). Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum*, *36*(1), 3–37.

Mlakar, M., Tušar, T., & Filipič, B. (2012). Discrete vs. continuous multiobjective optimization of continuous casting of steel. In T. Soule & J. H. Moore (Eds.), *Companion material proceedings of the genetic and evolutionary computation conference, GECCO '12, Philadelphia, PA, USA* (pp. 587–590). ACM.

Obayashi, S. & Sasaki, D. (2003). Visualization and data mining of Pareto solutions using self-organizing map. In C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, & L. Thiele (Eds.), *Proceedings of the 2nd international conference on evolutionary multi-criterion optimization, EMO 2003, Faro, Portugal* (Vol. 2632, pp. 796–809). Lecture Notes in Computer Science. Springer.

Poles, S., Geremia, P., Campos, F., Weston, S., & Islam, M. (2007). MOGA-II for an automotive cooling duct optimization on distributed resources. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.), *Proceedings of the 4th international conference on evolutionary multi-criterion optimization, EMO 2007, Matsushima, Japan* (Vol. 4403, pp. 633–644). Lecture Notes in Computer Science. Springer.

Pryke, A., Mostaghim, S., & Nazemi, A. (2007). Heatmap visualisation of population based multi objective algorithms. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.), *Proceedings of the 4th international conference on evolutionary multi-criterion optimization, EMO 2007, Matsushima, Japan* (Vol. 4403, pp. 361–375). Lecture Notes in Computer Science. Springer.

Purshouse, R. C. & Fleming, P. J. (2003). Evolutionary multi-objective optimisation: An exploratory analysis. In *Proceedings of the IEEE congress on evolutionary computation, CEC 2003, Canberra, Australia* (pp. 2066–2073). IEEE.

Robič, T. & Filipič, B. (2005). DEMO: Differential evolution for multiobjective optimization. In C. A. Coello Coello, A. Hernández Aguirre, & E. Zitzler (Eds.), *Proceedings of the third international conference on evolutionary multi-criterion optimization, EMO 2005, Guanajuato, Mexico* (Vol. 3410, pp. 520–533). Lecture Notes in Computer Science. Springer.

Rubinstein, R. Y. & Kroese, D. P. (2008). *Simulation and the Monte Carlo method*. Hoboken, NJ, USA: Wiley.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, *C-18*(5), 401–409.

Saxena, D. K., Duro, J. A., Tiwari, A., Deb, K., & Zhang, Q. (2013). Objective reduction in many-objective optimization: Linear and nonlinear algorithms. *IEEE Transactions on Evolutionary Computation*, *17*(1), 77–99.

Šereda, P., Vilanova, A., & Gerritsen, F. A. (2006). Automating transfer function design for volume rendering using hierarchical clustering of material boundaries. In *Proceedings of the eighth joint eurographics / IEEE VGTC conference on visualization, EUROVIS'06, Aire-la-Ville, Switzerland* (pp. 243–250). Eurographics Association.

Storn, R. & Price, K. V. (1997). Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

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

Tušar, T. & Filipič, B. (2011). Visualizing 4D approximation sets of multiobjective optimizers with prosections. In N. Krasnogor & P. L. Lanzi (Eds.), *Proceedings of the genetic and evolutionary computation conference, GECCO '11, Dublin, Ireland* (pp. 737–744). ACM.

Tušar, T. & Filipič, B. (2012). Scaling and visualizing multiobjective optimization test problems with knees. In M. Bohanec, M. Gams, D. Mladenić, M. Grobelnik, M. Heričko, U. Kordeš, ... M. Bernik (Eds.), *Proceedings of the 15th international multiconference information society, IS 2012, Ljubljana, Slovenia* (Vol. A, pp. 155–158). Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T. & Filipič, B. (2013). An approach to visualizing the 3D empirical attainment function. In C. Blum & E. Alba (Eds.), *Companion material proceedings of the genetic and evolutionary computation conference, GECCO '13, Amsterdam, The Netherlands* (pp. 1367–1372). ACM.

Tušar, T. & Filipič, B. (2014a). Initial experiments in visualization of empirical attainment function differences using maximum intensity projection. In C. Igel & D. V. Arnold (Eds.), *Companion material proceedings of the genetic and evolutionary computation conference, GECCO '14, Vancouver, BC, Canada* (pp. 1099–1106). ACM.

Tušar, T. & Filipič, B. (2014b). Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation.* In press. doi:10.1109/TEVC.2014.2313407

Tušar, T. & Filipič, B. (2014c). Visualizing exact and approximated 3D empirical attainment functions. *Mathematical Problems in Engineering.* In press.

Tweedie, L., Spence, R., Dawkes, H., & Su, H. (1996). Externalising abstract mathematical models. In *Proceedings of the SIGCHI conference on human factors in computing systems, CHI '96, Vancouver, BC, Canada* (pp. 406–412). ACM.

Valdés, J. & Barton, A. (2007). Visualizing high dimensional objective spaces for multiobjective optimization: A virtual reality approach. In *Proceedings of the IEEE congress on evolutionary computation, CEC 2007, Singapore* (pp. 4199–4206). IEEE.

Vertnik, R. & Šarler, B. (2009). Simulation of continuous casting of steel by a meshless technique. *International Journal of Cast Metals Research*, *22*(1–4), 311–313.

Voreen, Volume rendering engine. (2014). Retrieved June 1, 2014, from http://voreen.uni-muenster.de

Walker, D. J., Everson, R. M., & Fieldsend, J. E. (2010). Visualisation and ordering of many-objective populations. In *Proceedings of the IEEE congress on evolutionary computation, CEC 2010, Barcelona, Spain* (pp. 1–8). IEEE.

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

Walker, D. J., Fieldsend, J. E., & Everson, R. M. (2012). Visualising many-objective populations. In T. Soule & J. H. Moore (Eds.), *Companion material proceedings of the* *genetic and evolutionary computation conference, GECCO '12, Philadelphia, PA, USA* (pp. 451–458). ACM.

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

Yamamoto, M., Yoshikawa, T., & Furuhashi, T. (2010). Study on effect of MOGA with interactive island model using visualization. In *Proceedings of the IEEE congress on evolutionary computation, CEC 2010, Barcelona, Spain* (pp. 1–6). IEEE.

Zhang, Q. & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, *11*(6), 712–731.

Zitzler, E., Brockhoff, D., & Thiele, L. (2001). The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.), *Proceedings of the 4th international conference on evolutionary multi-criterion optimization, EMO 2007, Matsushima, Japan* (Vol. 4403, pp. 862–876). Lecture Notes in Computer Science. Springer.

Zitzler, E., Laumanns, M., & Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm. In K. C. Giannakoglou (Ed.), *Proceedings of evolutionary methods for design, optimization and control with applications to industrial problems, EUROGEN 2001, Athens, Greece* (pp. 95–100). International Center for Numerical Methods in Engineering (CIMNE).

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & Grunert da Fonseca, V. D. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, *7*(2), 117–132.

# Bibliography

## Publications Related to the Thesis

### Journal Articles

Tušar, T. & Filipič, B. (2014b). Visualization of Pareto front approximations in evolutionary multiobjective optimization: A critical review and the prosection method. *IEEE Transactions on Evolutionary Computation*. In press. doi:10.1109/TEVC.2014.2313407

Tušar, T. & Filipič, B. (2014c). Visualizing exact and approximated 3D empirical attainment functions. *Mathematical Problems in Engineering*. In press.

### Conference Papers

Tušar, T. & Filipič, B. (2011). Visualizing 4D approximation sets of multiobjective optimizers with prosections. In N. Krasnogor & P. L. Lanzi (Eds.), *Proceedings of the genetic and evolutionary computation conference, GECCO '11, Dublin, Ireland* (pp. 737–744). ACM.

Tušar, T. & Filipič, B. (2012). Scaling and visualizing multiobjective optimization test problems with knees. In M. Bohanec, M. Gams, D. Mladenić, M. Grobelnik, M. Heričko, U. Kordeš, . . . M. Bernik (Eds.), *Proceedings of the 15th international multiconference information society, IS 2012, Ljubljana, Slovenia* (Vol. A, pp. 155–158). Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T. & Filipič, B. (2013). An approach to visualizing the 3D empirical attainment function. In C. Blum & E. Alba (Eds.), *Companion material proceedings of the genetic and evolutionary computation conference, GECCO '13, Amsterdam, The Netherlands* (pp. 1367–1372). ACM.

Tušar, T. & Filipič, B. (2014a). Initial experiments in visualization of empirical attainment function differences using maximum intensity projection. In C. Igel & D. V. Arnold (Eds.), *Companion material proceedings of the genetic and evolutionary computation conference, GECCO '14, Vancouver, BC, Canada* (pp. 1099–1106). ACM.

## Other Publications

### Journal Articles

Dovgan, E., Javorski, M., Tušar, T., Gams, M., & Filipič, B. (2013). Comparing a multiobjective optimization algorithm for discovering driving strategies with humans. *Expert Systems with Applications*, *40*(7), 2687–2695.

Dovgan, E., Javorski, M., Tušar, T., Gams, M., & Filipič, B. (2014). Discovering driving strategies with a multiobjective optimization algorithm. *Applied Soft Computing*, *16*, 50–62.

Dovgan, E., Kaluža, B., Tušar, T., & Gams, M. (2010). Improving user verification by implementing an agent-based security system. *Journal of Ambient Intelligence and Smart Environments*, *1*(2), 21–30.

Dovgan, E., Tušar, T., Javorski, M., & Filipič, B. (2012). Discovering comfortable driving strategies using simulation-based multiobjective optimization. *Informatica*, *3*(36), 319–326.

Filipič, B., Tušar, T., & Laitinen, E. (2007). Preliminary numerical experiments in multiobjective optimization of a metallurgical production process. *Informatica*, *2*(31), 233–240.

Gams, M. & Tušar, T. (2007b). Intelligent high-security access control. *Informatica*, *4*(31), 469–477.

Kaluža, B., Dovgan, E., Tušar, T., Tambe, M., & Gams, M. (2011). A probabilistic risk analysis for multimodal entry control. *Expert Systems with Applications*, *6*(38), 6696–6704.

Marinčič, D., Tušar, T., Gams, M., & Šef, T. (2009). Analysis of automatic stress assignment in Slovene. *Informatica (Vilnius)*, *1*(20), 35–35.

Tušar, T., Korošec, P., Papa, G., Šilc, J., & Filipič, B. (2007). A comparative study of stochastic optimization methods in electric motor design. *Applied Intelligence*, *2*(27), 101–111.

### Conference Papers

Böhm, M., Dannecker, L., Doms, A., Dovgan, E., Filipič, B., Fischer, U., . . . Tušar, T. (2012). Data management in the MIRABEL smart grid system. In D. Srivastava & I. Ari (Eds.), *Proceedings of the 2012 joint EDBT-ICDT workshop* (pp. 95–102). New York, NY, USA: ACM.

Čepin, G., Tušar, T., & Filipič, B. (2013). Iskanje podobnih enobarvnih fragmentov pri množičnem sestavljanju stenskih poslikav. In B. Zajc & A. Trost (Eds.), *Proceedings of the twenty-second international electrotechnical and computer science conference, ERK 2013, Portorož, Slovenia* (Vol. B, pp. 73–76). In Slovene. IEEE Region 8.

Depolli, M., Tušar, T., & Filipič, B. (2006). Uglaševanje parametrov evolucijskega algoritma za večkriterijsko optimiranje na industrijskem problemu. In B. Zajc & A. Trost (Eds.), *Proceedings of the fifteenth international electrotechnical and computer science conference, ERK 2006, Portorož, Slovenia* (Vol. B, pp. 95–98). In Slovene. IEEE Region 8.

Dovgan, E., Javorski, M., Tušar, T., & Filipič, B. (2012). Dealing with comfort as an objective in multiobjective optimization of driving strategies. In M. Bohanec, M. Gams, D. Mladenić, M. Grobelnik, M. Heričko, U. Kordeš, . . . M. Bernik (Eds.), *Proceedings of the 15th international multiconference information society, IS 2012, Ljubljana, Slovenia* (Vol. A, pp. 103–106). Ljubljana, Slovenia: Jožef Stefan Institute.

Dovgan, E., Kaluža, B., Tušar, T., & Gams, M. (2009). Agent-based security system for user verification. In *Proceedings of the IEEE/WIC/ACM international joint conferences on web intelligence and intelligent agent technology, WI-IAT '09, Milan, Italy* (pp. 331–334). IEEE.

Dovgan, E., Tušar, T., & Filipič, B. (2010). Parameter tuning in an evolutionary algorithm for commodity transportation optimization. In *Proceedings of the IEEE congress on evolutionary computation, CEC 2010, Barcelona, Spain* (pp. 3704–3711). IEEE.

Filipič, B., Mlakar, M., Dovgan, E., & Tušar, T. (2011). Razvoj sistema za računalniško podprto evidentiranje in sestavljanje fragmentov stenskih poslikav. In M. Bohanec, M. Gams, D. Mladenić, M. Grobelnik, M. Heričko, U. Kordeš, . . . M. Bernik (Eds.), *Proceedings of the 14th international multiconference information society, IS 2011,* *Ljubljana, Slovenia* (Vol. A, pp. 45–48). In Slovene. Ljubljana, Slovenia: Jožef Stefan Institute.

Filipič, B. & Robič, T. (2004). A comparative study of coolant flow optimization on a steel casting machine. In *Proceedings of the IEEE congress on evolutionary computation, CEC 2004, Portland, OR, USA* (pp. 569–573). IEEE.

Filipič, B. & Tušar, T. (2013). Challenges of applying optimization methodology in industry. In C. Blum & E. Alba (Eds.), *Companion material proceedings of the genetic and evolutionary computation conference, GECCO '13, Amsterdam, The Netherlands* (pp. 1103–1104). ACM.

Filipič, B., Tušar, T., & Laitinen, E. (2006). Computer-assisted analysis of a metallurgical production process in view of multiple objectives. In B. Filipič & J. Šilc (Eds.), *Proceedings of the second international conference on bioinspired optimization methods and their applications, BIOMA 2006, Ljubljana, Slovenia* (pp. 167–176). Ljubljana, Slovenia: Jožef Stefan Institute.

Filipič, B., Tušar, T., & Laitinen, E. (2008). Analyzing production processes performance through evolutionary multiobjective optimization. In *Proceedings of the 3rd European symposium on nature-inspired smart information systems, NiSIS 2007, St. Julians, Malta* (pp. 87–92).

Gams, M., Krivec, J., Ožek, M., & Tušar, T. (2008). Implementacija procesno analitske tehnologije (PAT) s pomočjo tehnik strojnega učenja. In B. Zajc & A. Trost (Eds.), *Proceedings of the seventeenth international electrotechnical and computer science conference, ERK 2008, Portorož, Slovenia* (Vol. B, pp. 153–156). In Slovene. IEEE Region 8.

Gams, M. & Tušar, T. (2007a). Inteligentna biometrija za nadzor vstopa. In M. Bohanec, M. Gams, V. Rajkovič, T. Urbančič, M. Bernik, D. Mladenić, ... O. Markič (Eds.), *Proceedings of the 10th international multiconference information society, IS 2007, Ljubljana, Slovenia* (pp. 76–79). In Slovene. Ljubljana, Slovenia: Jožef Stefan Institute.

Kaluža, B., Dovgan, E., Tušar, T., & Gams, M. (2009). Intelligent risk analysis in access control. In J. Marecki, P. Paruchuri, C. Kiekintveld, & K. Sycara (Eds.), *Proceedings of the international workshop on quantitative risk analysis for security applications, QRASA, Pasadena, CA, USA* (pp. 9–16).

Mlakar, M., Tušar, T., & Filipič, B. (2012b). Discrete vs. continuous multiobjective optimization of continuous casting of steel. In T. Soule & J. H. Moore (Eds.), *Companion material proceedings of the genetic and evolutionary computation conference, GECCO '12, Philadelphia, PA, USA* (pp. 587–590). ACM.

Mlakar, M., Tušar, T., & Filipič, B. (2013). Primerjava rešitev ob negotovosti v večkriterijski optimizaciji. In M. Gams, R. Piltaver, D. Mladenić, M. Grobelnik, F. Novak, B. Blažica, ... A. Brodnik (Eds.), *Proceedings of the 16th international multiconference information society, IS 2013, Ljubljana, Slovenia* (Vol. A, pp. 86–89). In Slovene. Ljubljana, Slovenia: Jožef Stefan Institute.

Ožek, M., Gams, M., Krivec, J., & Tušar, T. (2008). Klasifikacija in vizualizacija procesov z metodami strojnega učenja. In M. Bohanec, M. Gams, V. Rajkovič, T. Urbančič, M. Bernik, D. Mladenić, ... O. Markič (Eds.), *Proceedings of the 11th international multiconference information society, IS 2008, Ljubljana, Slovenia* (pp. 34–36). In Slovene. Ljubljana, Slovenia: Jožef Stefan Institute.

Poles, S., Rigoni, E., & Robič, T. (2004). MOGA-II performance on noisy optimization problems. In B. Filipič & J. Šilc (Eds.), *Proceedings of the international conference on bioinspired optimization methods and their applications, BIOMA 2004, Ljubljana, Slovenia* (pp. 51–62). Ljubljana, Slovenia: Jožef Stefan Institute.

Robič, T. (2005). Performance of DEMO on new test problems: A comparison study. In B. Zajc & A. Trost (Eds.), *Proceedings of the fourteenth international electrotechnical and computer science conference, ERK 2005, Portorož, Slovenia* (Vol. B, pp. 121–124). IEEE Region 8.

Robič, T. & Filipič, B. (2004a). In search for an efficient parameter tuning method for steel casting. In B. Filipič & J. Šilc (Eds.), *Proceedings of the international conference on bioinspired optimization methods and their applications, BIOMA 2004, Ljubljana, Slovenia* (pp. 83–94). Ljubljana, Slovenia: Jožef Stefan Institute.

Robič, T. & Filipič, B. (2004b). Večkriterijska optimizacija z evolucijskimi algoritmi. In B. Zajc (Ed.), *Proceedings of the thirteenth international electrotechnical and computer science conference, ERK 2004, Portorož, Slovenia* (Vol. B, pp. 149–152). In Slovene. IEEE Region 8.

Robič, T. & Filipič, B. (2005). DEMO: Differential evolution for multiobjective optimization. In C. A. Coello Coello, A. Hernández Aguirre, & E. Zitzler (Eds.), *Proceedings of the third international conference on evolutionary multi-criterion optimization, EMO 2005, Guanajuato, Mexico* (Vol. 3410, pp. 520–533). Lecture Notes in Computer Science. Springer.

Robič, T. & Šef, T. (2004a). Uporaba oblikoslovnih oznak pri naglaševanju slovenskih besed. In B. Zajc (Ed.), *Proceedings of the thirteenth international electrotechnical and computer science conference, ERK 2004, Portorož, Slovenia* (Vol. B, pp. 153–156). In Slovene. IEEE Region 8.

Robič, T. & Šef, T. (2004b). Vpliv izbire atributov na točnost naglaševanja nepoznanih slovenskih besed. In T. Erjavec & J. Ž. Gros (Eds.), *Proceedings of the 7th international multiconference information society, IS 2004, Ljubljana, Slovenia* (Vol. B, pp. 115–118). Ljubljana, Slovenia: Jožef Stefan Institute.

Šef, T., Piltaver, R., & Tušar, T. (2013). Projekt OpUS: "Optimizacija upravljanja energetsko učinkovitih pametnih stavb". In M. Gams, R. Piltaver, D. Mladenić, M. Grobelnik, F. Novak, B. Blažica, ... A. Brodnik (Eds.), *Proceedings of the 16th international multiconference information society, IS 2013, Ljubljana, Slovenia* (Vol. A, pp. 110–113). In Slovene. Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T. (2007). Optimizing accuracy and size of decision trees. In B. Zajc & A. Trost (Eds.), *Proceedings of the sixteenth international electrotechnical and computer science conference, ERK 2007, Portorož, Slovenia* (Vol. B, pp. 81–84). IEEE Region 8.

Tušar, T., Bratko, A., Gams, M., & Šef, T. (2005). Comparison between humans and machines on the task of accentuation of Slovene words. In O. Markič, M. Gams, U. Kordeš, M. Heričko, D. Mladenić, M. Grobelnik, ... M. Bohanec (Eds.), *Proceedings of the 8th international multiconference information society, IS 2005, Ljubljana, Slovenia* (pp. 353–356). Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T., Dovgan, E., & Filipič, B. (2011). Scheduling of flexible electricity production and consumption in a future energy data management system: Problem formulation. In *Proceedings of the 14th international multiconference information society, IS 2011, Ljubljana, Slovenia* (Vol. A, pp. 96–99). Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T., Dovgan, E., & Filipič, B. (2012). Evolutionary scheduling of flexible offers for balancing electricity supply and demand. In *Proceedings of IEEE congress on evolutionary computation, CEC 2012, Brisbane, Australia* (pp. 1212–1219). IEEE.

Tušar, T. & Filipič, B. (2007). Differential evolution versus genetic algorithms in multiobjective optimization. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, & T. Murata (Eds.), *Proceedings of the 4th international conference on evolutionary multi-criterion* *optimization, EMO 2007, Matsushima, Japan* (Vol. 4403, pp. 257–271). Lecture Notes in Computer Science. Springer.

Tušar, T., Filipič, B., Dovgan, E., Mahnič, B., Čepin, G., Kuret, J., ... Vladušič, D. (2013). Aplikacija e-Pedius za podporo množičnemu sestavljanju fragmentov stenskih poslikav. In M. Gams, R. Piltaver, D. Mladenić, M. Grobelnik, F. Novak, B. Blažica, ... A. Brodnik (Eds.), *Proceedings of the 16th international multiconference information society, IS 2013, Ljubljana, Slovenia* (Vol. A, pp. 122–125). In Slovene. Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T. & Gams, M. (2006). Odkrivanje izjem na primeru inteligentnega sistema za kontrolo pristopa. In M. Bohanec, M. Gams, V. Rajkovič, T. Urbančič, M. Bernik, D. Mladenić, ... B. Novak Škarja (Eds.), *Proceedings of the 9th international multiconference information society, IS 2006, Ljubljana, Slovenia* (Vol. B, pp. 136–139). Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T. & Gams, M. (2007). Iskanje dobrih nastavitev modula v inteligentnem sistemu za nadzor pristopa. In M. Bohanec, M. Gams, V. Rajkovič, T. Urbančič, M. Bernik, D. Mladenić, ... O. Markič (Eds.), *Proceedings of the 10th international multiconference information society, IS 2007, Ljubljana, Slovenia* (pp. 103–106). In Slovene. Ljubljana, Slovenia: Jožef Stefan Institute.

Tušar, T., Šikšnys, L., Pedersen, T. B., Dovgan, E., & Filipič, B. (2012). Using aggregation to improve the scheduling of flexible energy offers. In B. Filipič & J. Šilc (Eds.), *Proceedings of the fifth international conference on bioinspired optimization methods and their applications, BIOMA 2012, Bohinj, Slovenia* (pp. 347–358). Ljubljana, Slovenia: Jožef Stefan Institute.

# Biography

Tea Tušar, born Robič, grew up in Nova Gorica. After secondary school, where she was awarded the Zois scholarship for gifted students, she moved to Ljubljana to attend the Faculty of Mathematics and Physics at the University of Ljubljana. In 2002 she received the B.Sc. degree in applied mathematics with the thesis entitled *A Genetic Algorithm for the School Timetabling Problem*. She continued her studies at the Faculty of Computer and Information Science at the same university.

In 2004 she became a research assistant at the Department of Intelligent Systems of the Jožef Stefan Institute. At first her research interests were diverse, spanning from evolutionary algorithms to outlier detection in access control systems and machine learning methods for knowledge discovery and text processing. She eventually focused on evolutionary algorithms for single- and multiobjective optimization with applications in optimization of production processes and engineering design.

In 2007 she received her M.Sc. degree in computer and information science with the thesis entitled *Design of an Algorithm for Multiobjective Optimization with Differential Evolution*. In 2009 she became a teaching assistant at the Jožef Stefan International Postgraduate School. Her academic career was interrupted by two year-long maternity leaves in 2008 and 2010. In 2011 she enrolled in the Jožef Stefan International Postgraduate School, where she is currently pursuing a Ph.D. in information and communication technologies. In accordance with her dissertation topic her research interests now include visualization techniques used in multiobjective optimization.

Since the start of her career she has been involved in the organization of different conferences and workshops: the biennial International Conference on Bioinspired Optimization Methods and their Applications (BIOMA 2004–2012), the International Multiconference Information Society (IS 2004–2006), and more recently the Student Workshop at the Genetic and Evolutionary Computation Conference (SWS@GECCO 2013–2014) and the conference Parallel Problem Solving from Nature (PPSN 2014). She has been a member of the Slovenian Artificial Intelligence Society (SLAIS) since 2004 and a student member of IEEE since 2012.