Author(s): Hakanen, Jussi; Miettinen, Kaisa; Matković, Kresimir

Title: Task-based visual analytics for interactive multiobjective optimization

Year: 2021

Version: Accepted version (Final draft)

Task-based visual analytics for interactive multiobjective optimization

Jussi Hakanen* and Kaisa Miettinen
University of Jyvaskyla, Faculty of Information Technology
P.O. Box 35 (Agora), FI-40014 University of Jyvaskyla, Finland
e-mail: jussi.hakanen@jyu.fi, kaisa.miettinen@jyu.fi

Krešimir Matković
VRVis Research Center
Donau-City Str. 11, A 1220 Vienna, Austria
e-mail: matkovic@vrvis.at

Abstract

We study how visual interaction techniques considered in visual analytics can be utilized when implementing interactive multiobjective optimization methods, where a decision maker iteratively participates in the solution process. We want to benefit from previous research and avoid re-inventing ideas. Our aim is to widen awareness and increase the applicability of interactive methods for solving real-world problems. As a concrete approach, we introduce seven high-level tasks that are relevant for interactive methods. These high-level tasks are based on low-level tasks proposed in the visual analytics literature. In addition, we give an example on how the high-level tasks can be implemented and demonstrate this in the context of a real-world multiobjective optimization problem related to wastewater treatment plant operation. Finally, we make recommendations for implementations of interactive methods. We conclude that task-based visual analytics can help in implementing interaction between human decision makers and interactive multiobjective optimization methods.

Keywords: Decision maker, preference information, multiple criteria optimization, visualization, task taxonomy, user interface

*Corresponding author

1 Introduction

Real-world optimization problems typically contain multiple conflicting objectives that need to be optimized simultaneously, e.g., minimize costs, maximize quality, minimize negative environmental impact and maximize reliability. Appropriate methods are needed (see e.g. Hwang and Masud (1979); Miettinen (1999)) to solve these multiobjective optimization problems. In this paper, we concentrate on so-called *interactive methods* (Miettinen, 1999; Miettinen, Ruiz, & Wierzbicki, 2008), which involve a *human decision maker* (e.g. a designer or an operator) in an iterative solution process of finding a most preferred Pareto optimal solution between the conflicting objectives. Pareto optimality means that none of the objective values can be improved without impairing some other(s). By ”most preferred”, we mean a Pareto optimal solution found during the solution process that the decision maker prefers over the other ones.

In interactive methods, a decision maker (DM) iteratively directs the solution process by providing preference information. Interactive methods differ from each other in terms of how and in which form a DM is expected to provide preference information and how this information is used in generating new Pareto optimal solutions. By employing an interactive method, a DM is able to learn about both the interdependencies between the objectives and the feasibility of one's preferences. Thanks to this learning and the insight gained, a DM can get convinced of the goodness of the most preferred solution. This makes interactive methods well-suited in solving real-world optimization problems (Miettinen, Hakanen, & Podkopaev, 2016; Miettinen et al., 2008).

Even though many interactive methods have been developed, very few implementations enabling their real-world application are available. In Kaliszewski (2004), it was observed that the number of applications of interactive methods to real-world problems was modest despite the availability of computer implementations. The complexity of the methods available for use by real DMs was assumed to explain this. In Stewart et al. (2008) as one potential reason for the limited use of interactive methods in engineering and design was assumed to be the reluctance to accept new methods since they have mainly relied on automated optimization processes without much subjectivity.

In Poles, Vassileva, and Sasaki (2008), as properties of good multiobjective optimization software were mentioned an easy-to-use graphical user interface and a good tool for visualizing results and choosing the final solution. It was also stated that multiobjective optimization had become more popular among engineering and scientific communities for solving real-world problems. The lack of available implementations can be seen as an obstacle in applying these methods in practical decision making.

The purpose of this paper is to support developers of interactive multiobjective optimization methods and software in their work to enhance interaction with the help of appropriate visualization techniques. When implementing any interactive method, a user interface facilitating the interaction between the method and the DM plays a crucial role (see e.g., Korhonen (2005)). For example, a badly implemented user interface can hinder a DM from using the method regardless of how good the method itself is. On the other hand, a well implemented user interface can make a 'not so good' method popular if it is easy and intuitive to use. Therefore, well implemented methods are needed in order to enable wider use of interactive methods in practical decision making (as emphasized e.g., in Weistroffer and Narula (1997)). Typically, developers of optimization methods are not experts in visualization (and other user interface techniques), so it is natural that they focus on what happens in the method itself. Thus, implementations of interactive methods may lack effective visualization.

In this paper, for the above-mentioned reasons, we turn to *visual analytics* to get assistance in implementing interaction in interactive multiobjective optimization methods. Visual analytics can be defined as follows (Keim et al., 2008): "Visual analytics combines automated analysis techniques with interactive visualizations for an effective understanding, reasoning and decision making on the basis of very large and complex data sets." Based on the definition, visual analytics is relevant in the visual interaction between an interactive multiobjective optimization method and a human DM. In this paper, we use the term *visual interaction* when referring to interaction between a DM and a visualization in order to distinguish this from interaction between a DM and a multiobjective optimization method.

We assume here that the DM is expected to compare several solutions in the solution process while using an interactive method. Further, we divide our consideration to DM providing preference information and analyzing Pareto optimal solutions. We devote our main focus to the latter as the former is more method-dependent. As far as visualization aspects are considered, the multiobjective optimization literature has mainly been interested in how to visually represent the Pareto optimal solutions to the DM, especially, when more than three objective functions are considered (Miettinen, 2014; Tusar & Filipic, 2015). Typical visualization techniques which are used include, among others, parallel coordinates plots (Wang, Purshouse, & Fleming, 2013), scatter plots, spider web charts (Trinkaus & Hanne, 2005), interactive decision maps (Lotov, Bushenkov, & Kamenev, 2004), heatmaps (Hettenhausen, Lewis, & Mostaghim, 2010), and radial visualizations (He & Yen, 2016; Ibrahim, Rahnamayan, Martin, & Deb, 2016). Most of the methods utilize individual visualization techniques (e.g., those discussed in Miettinen (2006)) but some use linked visualizations like Woodruff, Reed, and Simpson (2013). However, how the DM should analyse the generated solutions in order to gain more insight in the problem and provide updated preferences meaningfully has not been in main focus. Our mission is to support decision making further with appropriate visual interaction techniques.

We think that bringing the two research fields closer to each other can benefit both of them: visual analytics can offer support for the DM and, on the other hand, interactive multiobjective optimization can provide visual analytics with a new application area. Visual analytics is useful since it combines the strengths of human and machines by using their respective distinct capabilities for the most effective results (Keim et al., 2008). As stated in Keim et al. (2008), "the user has to be the ultimate authority in giving the direction of the analysis along his or her specific task" which is in line with the idea of interactive methods.

As far as we know, there are only two papers where visual analytics and interactive multiobjective optimization have been combined. In Tarkkanen, Miettinen, Hakanen, and Isomäki (2013), implementation challenges were considered by studying what kind of visual interaction techniques can support the decision making process. As an example, interaction design was applied to the Pareto Navigator method (Eskelinen, Miettinen, Klamroth, & Hakanen, 2010). The other example is the interactive multiobjective particle swarm optimization method introduced in Hettenhausen et al. (2010), where the user interface is based on a heatmap visualization and the design includes ideas from visual analytics. Furthermore, interactive visual analysis was applied to supporting selection of the most preferred PO solution after a set of PO solutions was generated in Berger and Piringer (2010).

In this paper, we study how and which visual interaction techniques from visual analytics can be utilized in designing and implementing interaction between an interactive multiobjective optimization method and a DM. We introduce high-level tasks that are relevant for interactive methods. These tasks are based on low-level tasks derived from the visualization literature. We also demonstrate how to implement the high-level tasks with a real-world example related to wastewater treatment plant operation. To summarize, our novel contribution is four-fold: 1) introduce seven high-level tasks, 2) give an example of how to implement these high-level tasks, 3) demonstrate the applicability with a real-world decision making process, and 4) make recommendations for implementations of interactive methods.

The structure of the paper is as follows. Section 2 contains relevant concepts used in multiobjective optimization and describes a general interactive multiobjective optimization algorithm. The proposed high-level tasks in interactive multiobjective optimization and how they are enabled by low-level tasks are introduced and discussed in Section 3. An example of how the high-level tasks can be implemented and applied is given in Section 4 based on a real-world solution process. Recommendations are given for implementing interactive methods applying lessons learned in the field of visual analytics in Section 5. Finally, the paper is concluded in Section 6.

2 Basics of interactive multiobjective optimization

Here, we briefly introduce the basic concepts and operating principles of interactive multiobjective optimization methods. Furthermore, we present a general algorithm for interactive methods to enable easier association to its phases, where visual interaction with the DM is beneficial.

2.1 Concepts

We consider multiobjective optimization problems with two or more (conflicting) nonlinear objective functions, which are optimized with respect to decision variable vectors over a feasible set. The feasible set may be formed by inequality, equality and/or box constraints. As an objective vector in the objective space we refer to a vector of objective function values calculated at any feasible decision variable vector. The objective functions can be either minimized or maximized depending on what they describe. In this paper, we demonstrate our ideas with a continuous nonlinear multiobjective optimization problem but the techniques discussed are applicable to other types of multiobjective optimization problems.

Because of the conflicting nature of the objective functions, no solution, where all the objectives can reach their individual optima, exists. Instead, we can identify a set of so-called Pareto optimal solutions where no objective function value can be improved without impairing at least one of the others. To be more specific, a feasible decision vector and the corresponding objective vector are *Pareto optimal* (PO) if there does not exist another feasible decision vector such that it gives a better or equal value for all the objectives and the value is strictly better for at least one of them. Typically, multiobjective optimization problems have many PO solutions and they form a Pareto optimal set. The number of PO solutions can be even infinite.

By solving a multiobjective optimization problem one can mean identifying the most preferred PO solution according to the preferences of a human DM or finding a representation of the PO set. Because real problems usually need one solution to be implemented, the final solution must be selected in one way or the other. As PO objective vectors cannot be ordered completely without additional information, the role of the DM is important because she can express preference information. Further, a DM can judge *trade-offs* between the conflicting objectives which roughly speaking means the ratio of change related to how much one objective improves when some other objective impairs (for a pair of objectives), see e.g. Miettinen (1999).

Multiobjective optimization methods can be classified according to the role of the DM (see e.g. Miettinen (1999)) in four classes. In no-preference methods, no preference information is available and some neutral compromise (Wierzbicki, 1999) has to be identified. In a priori methods, the DM first specifies some hopes or expectations and the solution best corresponding to them is found, whereas in a posteriori methods a representation of the PO set is generated and shown to the DM, who then must select the most preferred one. Examples of visualizations to support a DM in the selection can be found in He and Yen (2016); Ibrahim et al. (2016); Lotov and Miettinen (2008); Tusar and Filipic (2015). The fourth class, namely interactive methods, involves a solution pattern which is iteratively repeated: first the DM specifies preferences and, then, PO solution(s) are generated that obey the preferences as well as possible. They are shown to the DM, and during this interactive solution process, the DM can provide and/or correct one's preferences and gradually get convinced of which solution meets her preferences and needs best. Here, the amount of solutions generated and information the DM must handle at a time remain limited. Importantly, the DM can learn about the interdependencies among the objectives and the feasibility of one's preferences. For these desirable reasons, we mostly discuss interactive methods in this paper.

In many methods, ranges of the objective function values in the PO set provide valuable understanding about what kind of solutions are available. The best individual optima of the objective functions form an ideal objective vector in the objective space. The worst values in the PO set cannot be calculated easily as the entire PO set is unknown but they can be estimated to get a nadir objective vector in the objective space (see e.g. Miettinen (1999) and references therein).

2.2 General interactive multiobjective optimization algorithm

In this section, we present a general algorithm for interactive multiobjective optimization methods, which shows essential steps of interaction with the DM. Besides individual interactive methods, there have also been attempts to gather different types of interactive methods under a joint framework, which enables using different ways of expressing preferences during the interactive solution process, if necessary (Gardiner & Steuer, 1994a, 1994b; Luque, Ruiz, & Miettinen, 2011; Ruiz, Luque, & Miettinen, 2012).

Next, we present main steps of a general interactive multiobjective optimization algorithm based on Ojalehto, Miettinen, and Laukkanen (2014). The general algorithm involves two active parties: the DM and the algorithm. Here, the DM is asked to provide preference information and to select the solution that best corresponds to the preferences. On the other hand, the algorithm generates for the DM the initial solution and new PO solutions based on the preference information.

The steps of the general algorithm are the following:

1. Initialize the solution process, e.g., calculate the ideal and estimate the nadir objective vector.

2. By solving a method-specific subproblem, generate an initial PO solution to be used as a current solution.

3. Show the current solution to the DM.

4. Ask the DM to provide preference information related to the current solution.

5. Generate new solution(s) based on the preference information by solving appropriate subproblem(s).

6. Show the solutions generated in step 5 to the DM. Ask her to select the best one and denote it as the current solution.

7. If the selected current solution is satisfactory for the DM, stop. Otherwise, continue from step 4.

The interactive solution process can often be divided in two phases, namely, the *learning phase* and the *decision phase* (see e.g. Miettinen et al. (2008)). In the learning phase, the DM can specify different preferences in order to search for promising regions in different parts of the PO set and learn about problem characteristics. On the other hand, the preferences used in the decision phase are not necessarily that different from each other and the idea is to converge towards the most preferred PO solution in the region of interest identified in the learning phase. Note that it is not always easy to see where the learning phase ends and the decision phase starts. We will refer to these phases in Section 4, when we describe an example of a real interactive solution process.

The general interactive algorithm presented above does not directly apply to interactive evolutionary multiobjective optimization methods where the DM guides a population of solutions towards interesting parts of the PO set with her preferences (see, e.g., Jaszkiewicz and Branke (2008); Purshouse, Deb, Mansor, Mostaghim, and Wang (2014)). The reason for the difference is that such algorithms operate on a population of solutions at a time instead of formulating and solving single objective subproblems. However, the visual interaction techniques presented later in this paper can also be applied to support the implementation of interactive evolutionary multiobjective optimization methods.

An implementation of an interactive method can be characterized with two distinct parts: the user interface (steps 3, 4 and 6) and the algorithm (steps 1, 2 and 5). The preferences of the DM can be asked in different ways in step 4. For example, in Luque et al. (2011); Ruiz et al. (2012), four ways of expressing preferences are considered: a DM can 1) give a *reference point* consisting of *aspiration levels*, that is, desired values for each objective function (Jaszkiewicz & Slowinski, 1999; Wierzbicki, 1982; Wierzbicki, Makowski, & Wessels, 2000), 2) *classify* the objective functions in a current PO solution to reflect how function values should be changed to get a more preferred solution (Miettinen & Mäkelä, 2006; Nakayama & Furukawa, 1985), 3) select a preferred solution from a small set of sample solutions provided (Steuer & Choo, 1983) or 4) give desirable values for trade-offs involving different objectives (Luque, Yang, & Wong, 2009) (called marginal rates of substitution (Miettinen, 1999)).

Furthermore, one can consider navigation based methods that dynamically generate and visualize multiple PO solutions to the DM in a given direction in the objective space. In these methods (Eskelinen et al., 2010; Hartikainen, Miettinen, & Klamroth, 2019), the idea is to create an illusion of a continuous movement and the DM controls both the direction and the speed of movement when navigating among PO solutions.

Not all interactive multiobjective optimization methods contain an explicit possibility to return to previously generated solutions in their algorithms. However, it is often possible to enable this, when actually implementing the algorithm. We think that an ability to return to previously generated solutions and reconsider them will offer more support and learning possibilities for a DM. Therefore, it is advisable to have an archive to store all the generated solutions. In that case, appropriate visual interaction techniques can support their efficient analysis. More information on interactive multiobjective optimization methods can be found e.g. in Miettinen (1999); Miettinen et al. (2016, 2008).

3 High-level tasks in interactive multiobjective optimization

As already mentioned in the introduction, we want to study how techniques from visual analytics can be used to aid implementation of interactive multiobjective optimization methods from the interaction point of view. In the visual analytics field, task-based visual interaction techniques (along with task abstraction) have been proposed (see e.g. Amar and Stasko (2004); Lee, Plaisant, Parr, Fekete, and Henry (2006); Munzner (2014); Valiati, Pimenta, and Freitas (2006) together with ideas of how they could be implemented efficiently). These techniques enable users to directly or indirectly manipulate and interpret visual representations of solutions considered.

Static images or autonomously animated images do not have associated visual interaction techniques (Yi, Kang, Stasko, & Jacko, 2007), that is, a user can not interact with them. Static images are useful for many cases, but their usefulness is limited when considering large data sets consisting of high-dimensional data (Yi et al., 2007). This is the case in multiobjective optimization, since there are usually more than three objectives and several objective vectors to be analyzed and compared. In addition to the objectives, there can be a large number of decision variables which describe how each solution is implemented in practice and, thus, may contain very important information for DMs. However, visualizing the decision variables is typically application dependent and, therefore, not considered in this paper.

In interactive multiobjective optimization methods, the DM needs to constantly interact with the algorithm. While doing so, she faces tasks such as 1) compare PO solutions, 2) specify preferences, 3) check feasibility of preferences, 4) determine a most preferred solution in a subset of alternatives, 5) learn about problem characteristics, 6) detect correlations, and 7) post-process the most preferred solution, e.g., in terms of sensitivity, robustness or uncertainty. These tasks are here called *high-level tasks*. To enable achieving those tasks, visual interaction techniques can be implemented. The high-level tasks can be facilitated by combining simpler low-level tasks as shown in Figure 1. Different low-level tasks are considered e.g. in Heer and Schneiderman (2012); Kerren and

Schreiber (2012); Yi et al. (2007). Among those, we have selected those that are related to the high-level tasks mentioned above. They are connect, filter, explore, encode, select, organize, abstract/elaborate, derive/adjust and reconfigure and they will be introduced in the following section.

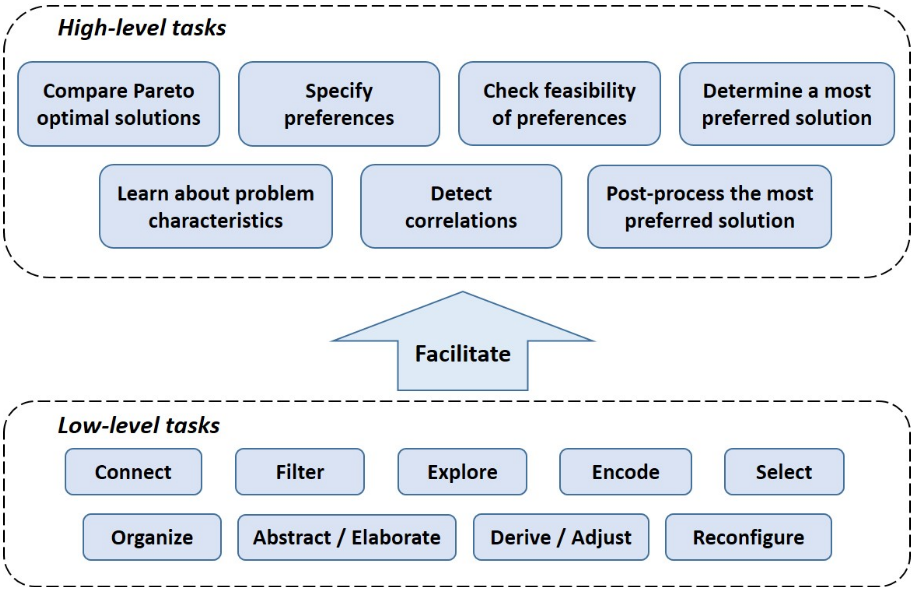

High-level tasks
Compare Pareto optimal solutions
Specify preferences
Check feasibility of preferences
Determine a most preferred solution
Learn about problem characteristics
Detect correlations
Post-process the most preferred solution
Facilitate
Low-level tasks
Connect
Filter
Explore
Encode
Select
Organize
Abstract / Elaborate
Derive / Adjust
Reconfigure

Figure 1: Low-level tasks facilitating high-level tasks in interactive multiobjective optimization.

3.1 Basic terminology on visualization

Visual analytics is defined as *"the science of analytical reasoning facilitated by interactive visual interfaces"* (Thomas & Cook, 2005). It is a combination of interactive exploration by means of interactive visualization and computational analysis methods. When considering the quickly growing number of success stories in academia and in many applications, interactive visualization and visual analysis can certainly be seen as an already well established and useful complement to computational data analysis, when it comes to the study of data with challenging content or structure.

In case of complex problems, a straight-forward use of automated methods is often not possible. An example of a complex problem is multiobjective optimization. The visual part is essential to integrate the user in the loop, to support her, and to exploit her imagination and intuition in order to solve complex problems. Instead of being automated, the whole process of insight gaining is rather iterative loop of refining parameters and analyzing results.

One of the most often deployed methods in visual analysis are coordinated multiple views (Gresh, Rogowitz, Winslow, Scollan, & Yung, 2000; North & Shneiderman, 2000; Roberts, 2007). The main idea of the coordinated multiple views is to depict different attributes of data items in at least two views. Now, the user can interactively select a subset of data points in one view and the same subset of data points is highlighted in all other views. The user selection is called *brushing*. The consistent emphasis of the same data points across all views is called *linking* and emphasizing certain data points in a visualization, for example by a consistent coloring scheme, is called *focus+context* visualization (Hauser, 2006).

Figures 2, 3, and 4 illustrate *linking and brushing* using two views. In what follows, we use a set of PO solutions for the DTLZ7 problem (Deb, Thiele, Laumanns, & Zitzler, 2001) having eight objectives as an example. The scatter plot on the left in Figure 2 shows the objective values for f_1 and f_2 while the parallel coordinates plot on the right shows the values for $f_3 - f_8$. Each PO solution or one record in the data set is represented by a point in the scatter plot and by a poly-line in the parallel coordinates plot.

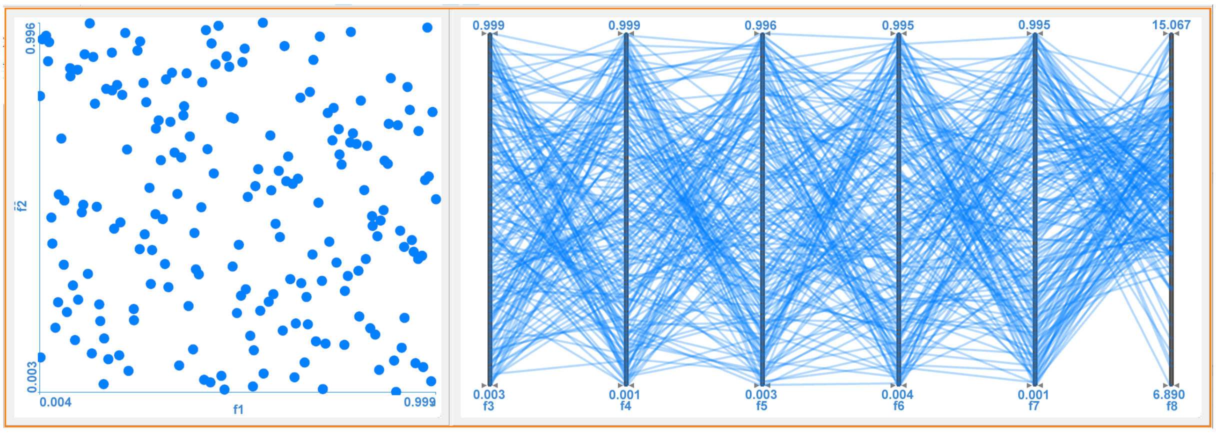


Figure 2: *Linking and brushing.* Two views show eight attributes (objective function values) of the data set.

What we cannot see is how the data items are connected, i.e. which points in the left scatter plot correspond to which lines in the right view. Linking and brushing solves this problem. Let us assume that a user is interested in the points with low values of f_1 and f_2. The user draws a rectangle selection in the left scatter plot and the brushed points are highlighted (Figure 3, top, left). At the same time the data items that correspond to the same records are highlighted in the linked view on the right (Figure 3, top, right). Now the user can move the brush in the left view, and observe what is happening in the right view (Figure 3, bottom). Note, for example changes on the right most axis in the right view. The low values of f_1 and f_2 have high values of f_8. If we brush high f_1 and keep the f_2 low, then f_8 decreases. In this way, the user gets insight into data and understands much better what is going on.

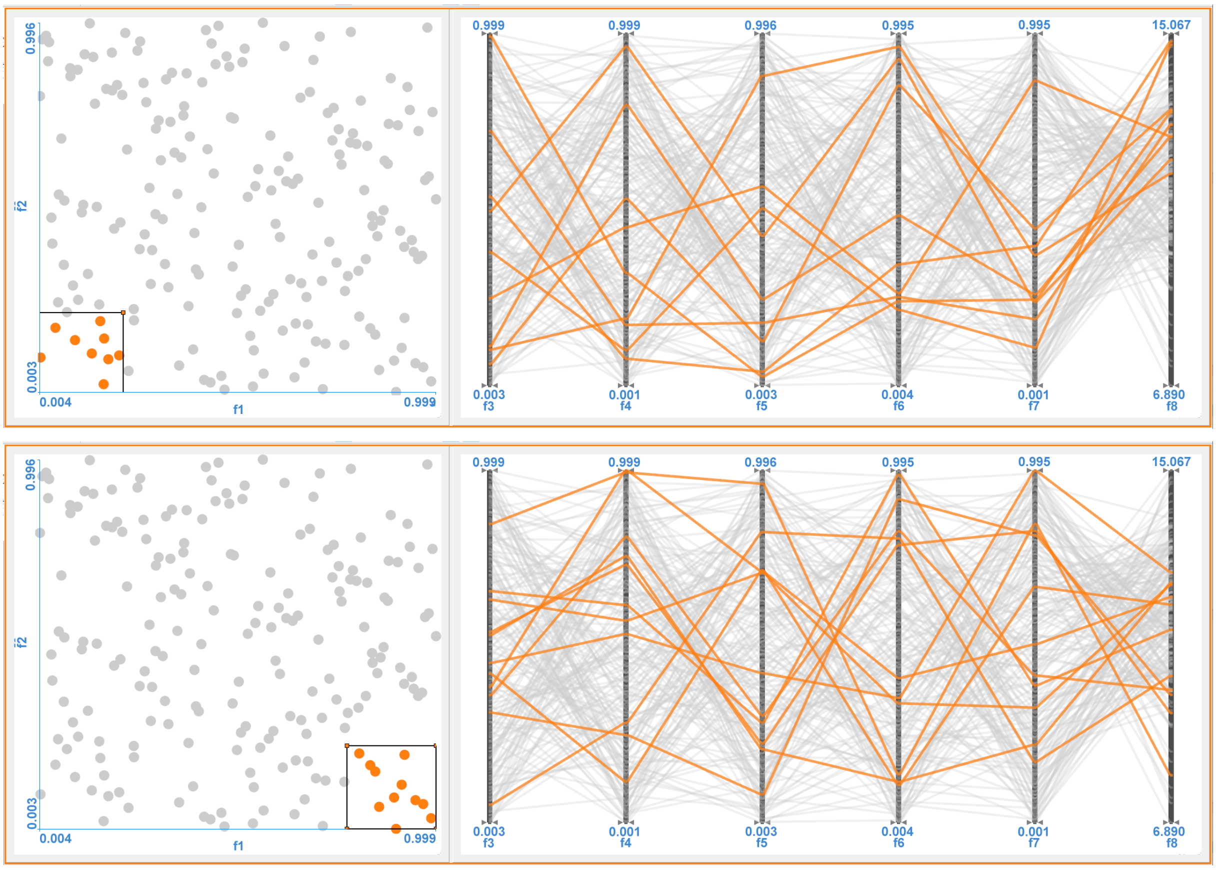


Figure 3: *Linking and brushing.* **Top:** The interactive selection of a data subset in one view (called *brushing*) leads to a consistent highlighting of this subset in all views (called *linking*). **Bottom:** The user can move the brush and observe changes in all views.

It is easy to refine selection by composite brushing. Brushes are combined using Boolean operations. In Figure 4 top, the user refines previous selection by brushing low values on the f_7 axis in the parallel coordinates plot. Note that there are some gray points in the brush area in the scatter plot and on the parallel coordinate axis. Only data items that are in both ranges at the same time are highlighted. Brushes from different views or from one view can be combined in a composite brush.

In the example which is depicted in Figure 4 bottom, the user combined three brushes in the parallel coordinates view.

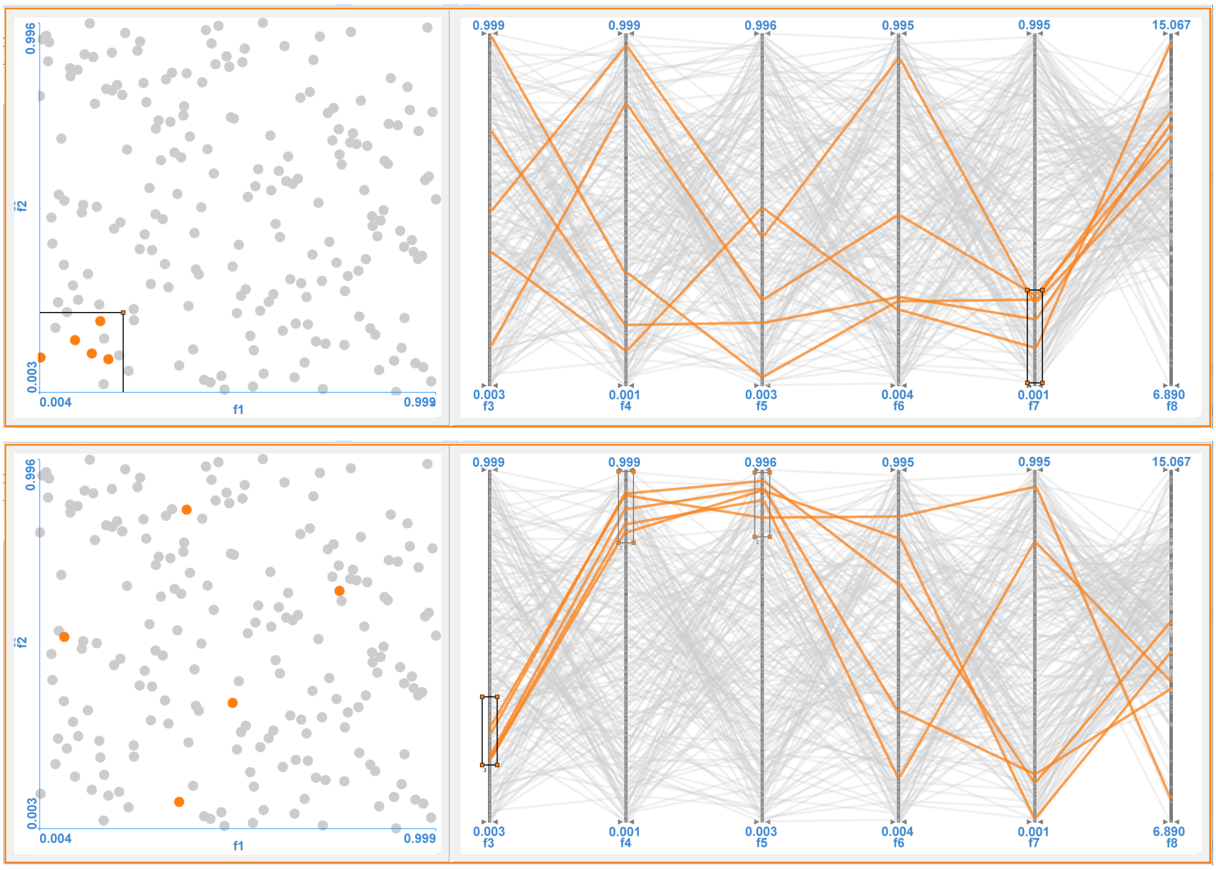


Figure 4: *Composite brushing.* It is possible to combine brushes using Boolean operations. The AND operation is used here. Individual brushes can be from any view.

In this simple illustrative example, only two views are used; usually, a more advanced analysis will depend on several (linked) views. Already a simple solution (like two views with linking and brushing) represents a significant improvement over static plots or the manual inspection of data. More views and advanced visual interaction solutions including multiple brushes, make the exploration even more powerful.

In case of multiple brushes, the user can combine several brushes using Boolean operations in an iterative interactive drill-down process. The user can easily refine selection using the AND operation and, if needed, can broaden the selection using the OR operation.

The qualitative nature of interactive visualization is its biggest advantage and a drawback at the same time. Radoš et al. (2016) propose several ways how to structure brushing in order to make it reproducible, and how to enhance linked views in order to make them more quantitative. Konyha, Lež, Matković, Jelović, and Hauser (2012) introduce three levels of interactive visual analysis. On the first level a single brush is available, the second level supports multiple brushes, and the third level, in addition, supports complex brushes and on-the fly data derivation and aggregation. A detailed overview of available linking and brushing research is beyond the scope of this paper.

Let us summarize the basic concepts used in visualization when coordinated multiple views are deployed.

Linking means that a selected data item(s) can be simultaneously identified in two or more views representing the same data. Linking enables studying the same data in different representations at the same time and helps in getting more insight when compared to only using a single representation.

Brushing refers to highlighting some data items among the data visualized. Typical to brushing is that the data items not highlighted are still shown in a view but differently e.g. in a different color. Brushing is closely related to *filtering* where some elements of the data are filtered out of the consideration and they are typically not shown. However, the definition of filtering in Yi et al. (2007) says that *”data items outside of the range or not satisfying the condition are hidden from the display or shown differently”* and *”the hidden or differently shown data items can be recovered”*. Since we use Yi et al. (2007) as the basis of our low-level tasks introduced in the next section, we will follow this definition and use filtering and brushing here as synonyms.

Select is usually defined depending on application scenario. In case of multiobjective optimization, we define it as selecting few solutions. It is usually done at the end of the drill-down process when some interesting cases are identified. It is not the end of the analysis process, but it is used to mark some of the solutions as potentially interesting.

3.2 Low-level tasks

Often in visual analytics, all pieces of the data are assumed to be readily available (e.g. in some database) and different parts of it can be explored. On the contrary in interactive multiobjective optimization, it is typical that not all PO solutions are known but they are continuously generated during the interactive solution process. In other words, actually, no complete set of solutions is ever available. Furthermore, visual interaction techniques in visual analytics do not typically make any changes to the data but only show subsets of it to the user.

In the visual analytics literature, research has been done in order to categorize visual interaction techniques. In Yi et al. (2007), seven categories of visual interaction techniques were proposed based on a thorough review of the existing literature and commercial systems of information visualization (in total, 59 papers and 51 systems were considered). The driving force in defining these categories was *user intent*, that is, "What a user wants to achieve". The categories were given the following names (a short name and a descriptive phrase, respectively (Yi et al., 2007)): 1) Select: mark something as interesting, 2) Explore: show me something else, 3) Reconfigure: show me a different arrangement, 4) Encode: show me a different representation, 5) Abstract/Elaborate: show me more or less detail, 6) Filter: show me something conditionally, and 7) Connect: show me related items. A more detailed description of these categories can be found in Yi et al. (2007).

More recently, Heer and Schneiderman (2012) and Kerren and Schreiber (2012) presented a taxonomy of interactive dynamics to assist designers in creating visualization tools. Within the taxonomy, visual interaction techniques were divided into three main categories: 1) data and view specification (encode/visualize, reconfigure, filter, sort, derive, adjust), 2) view manipulation (select, navigate/explore, coordinate/connect, organize), and 3) process and provenance (record, annotate, share, guide). The first category deals with initial exploration of the data and how it is visually represented, the second category considers interaction with the data through usage of different ways to use visual representations while the third category is mainly related to collaboration and information sharing.

The visual interaction techniques presented in Yi et al. (2007) and the ones in the three categories of Heer and Schneiderman (2012); Kerren and Schreiber (2012) implement *low-level tasks* and they are basic actions for the user to interact with data. Further, they can be used as building blocks for enabling *high-level tasks*. The three categories in Heer and Schneiderman (2012); Kerren and Schreiber (2012) describe general high-level goals. A more concrete example related to multiobjective optimization is that in order to be able to compare objective vectors (high-level task), one often has to first manipulate how they are visualized, e.g., filter or sort them or visualize them in some other way (low-level tasks). Examples of high-level tasks related to interactive multiobjective optimization are given in the next section.

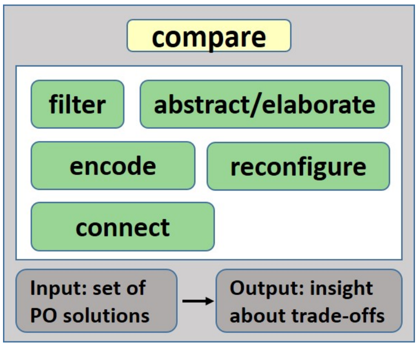
We think that augmenting the seven user intent based visual interaction techniques from Yi et al. (2007) with *derive/adjust* and *organize* interactions from Heer and Schneiderman (2012); Kerren and Schreiber (2012) provide a meaningful approach to study visual interaction techniques related to implementing interaction in interactive multiobjective optimization. Although collaboration and information sharing is important especially when dealing with multidisciplinary optimization, we will not consider visual interaction techniques related to them here.

Next, we briefly describe some of the selected ones (whose meaning is not obvious). In the *explore* interaction, the aim of the user is to see some other data items instead of the ones that are currently shown. The aim of the *reconfigure* interaction is to provide different ways of representing a given dataset. As opposed to the reconfigure interaction that considers spatial arrangement of the data, the *encode* interaction can be used to change how the data is represented visually. Encode is related, for example, to the visual appearance of each data item including colour, size and shape. The *connect* interaction connects related data items if multiple representations of the same data items are shown simultaneously. *Derive/adjust* in our context means that some of the parameters of the computations can be changed from the visual interface by the user (e.g. indirectly changing the parameters of scalarizing functions by specifying preferences). Finally, *organize* refers to how the different views are arranged in a user interface (e.g. have a two dimensional scatter plot next to a parallel coordinate plot or on top/on bottom of it).

3.3 High-level tasks

As already mentioned, we have identified seven high-level tasks that are relevant in interactive multiobjective optimization. From the visual interaction point of view, those can be facilitated by the low-level tasks introduced in Section 3.2. In this section, we will discuss the high-level tasks in more details. In the following subsections, the graphical representations associated with each high-level task are shown next to the descriptions and they follow the style applied in Bernold, Matkovic, Gröller, and Raidou (2019). The green color represents the related low-level tasks while the yellow color denotes the high-level task in question. In addition, input and output of the high-level task is shown. We also give concrete examples of use cases in the style adopted from Lee et al. (2006).

3.3.1 Compare Pareto optimal solutions



In different phases of an interactive solution process, the DM has to compare different PO solutions. As already mentioned, without any additional information, all PO solutions are mathematically equally good so the DM can use her domain knowledge to make the comparison supported by visual interactions. Note that all the low-level tasks shown in the examples below are not necessarily needed in all the cases.

Examples

- *"I want to compare solutions that were generated in the last iteration"*

  [*filter* to get solutions from the last iteration + *encode* (use different ways of visualization) + *reconfigure* (use different arrangement/order of data) + *connect* individual solutions in different visualizations + *elaborate* to see more details on some solutions]

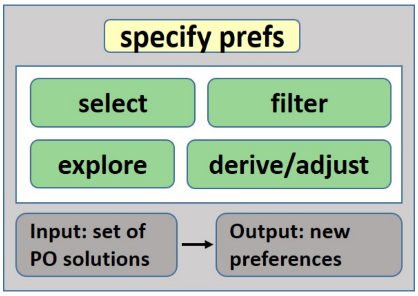
- *"I want to compare solutions where the first objective has values below 5.0"*

  [*filter* all solutions ($f_1 \leq 5.0$) + *encode* (use different ways of visualization) + *reconfigure* (use different arrangement/order of data) + *abstract* to get an overview + *connect* individual solutions in different visualizations + *elaborate* to see more details on some solutions]

- *"I want to compare solutions that I find promising"*

  [*abstract* to get an overview + *filter* to get solutions to be compared + *encode* (use different ways of visualization) + *reconfigure* (use different arrangement/order of data) + *connect* individual solutions in different visualizations + *elaborate* to see more details on some solutions]

3.3.2 Specify preferences



At each interaction, the DM is assumed to specify preferences. By specifying preferences, she wants to improve the current solution because the ones generated so far are not yet satisfying enough or she wants to explore other areas in the objective space. Therefore, this can be seen as an *explore* low-level task. In interactive methods based on solving scalarized problems, the preferences the DM specifies are used to change the parameters of the scalarized problems and, thus, the corresponding low-level task is *derive/adjust*. The DM does not need to know how the preferences are exactly converted to parameter values but only how she wants to improve the current solution. The style of expressing preference information depends on the method applied.

Examples

- *"I want to improve current solution by classifying objective functions"*

  [*select* current solution + *adjust* (specify preferences as classifying objectives)]

- *"I want to generate solutions in the area that I specify by giving corresponding ranges for each objective function"*

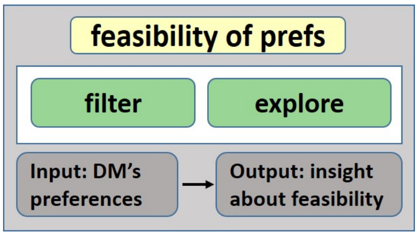
  [*filter* solutions to see whether there already exist such solutions + *explore* new area (generate new solutions) + specify preferences as ranges for each objective (*adjust* parameters for scalarization functions)]

- *"I want to see other solutions in the neighborhood of a selected solution"*

  [*select* one solution + *adjust* (specify preferences as desirable values for each objective)]

3.3.3 Check feasibility of preferences

In the learning phase, the DM often wants to explore different areas of the objective space, i.e., study different trade-offs. For this, she sets preferences accordingly to see whether there exist any PO solutions satisfying those preferences. If the method used can not find such solutions it means either that i) there does not exist such solutions or ii) the method is not able to find them. Here, *filter* and *explore* are the relevant low-level tasks.



Examples

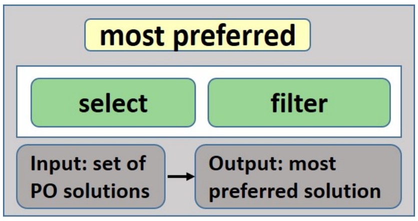
- *"I want to check whether there are any solutions generated in an area of the objective space that I am currently interested in"*

  [*filter* generated solutions + if no solutions available, *explore*]

- *"I want to see solutions that are different from the ones already generated"* [*filter* to find interesting areas that do not have any solutions generated + *explore* (try to find such solutions and obtain insight about feasibility)]

3.3.4 Determine most preferred solution

The DM has to determine a most preferred PO solution at the end of the interactive solution process and, sometimes, also during the solution process to continue with. This high-level task also requires to apply another high-level task: compare a set of PO solutions. The associated low-level tasks are *select* and *filter*.



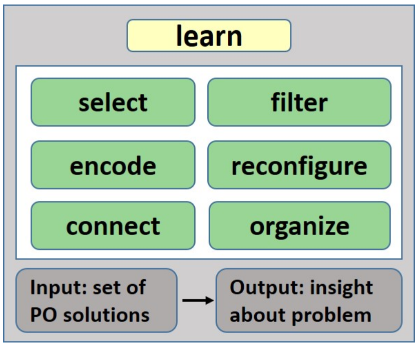
Example

- *"I want to select the most preferred solution among a set of solutions available"*

  [*filter* potential solutions + ***compare*** selected solutions + *filter* to exclude undesirable solutions + *select* the most preferred solution]

3.3.5 Learn about problem characteristics

The main task in multiobjective optimization is learning about trade-offs between conflicting objectives. The more objectives the problem has, the more challenging this learning becomes. Therefore, visual interaction plays here a crucial role. The six related low-level tasks can be seen on the right. Note that the low-level task *organize* is included here as an example but it can be present in any of the high-level tasks if needed.



Examples

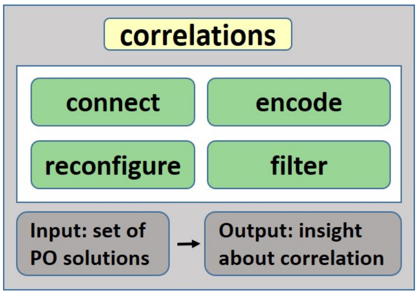
- *"I want to find out how the values of two or more objectives change within a given set of solutions"* [*filter* to get the set of solutions + decide the objectives of interest + *encode* the selected objectives differently + *reconfigure* (use different ways of visualization) + *connect* solutions in different visualizations + *organize* views differently (optional)]

- *"I want to see whether I can improve these two objectives simultaneously from the current values"*

  [*select* the current solution + decide two objectives of interest + *filter* to show solutions having better values for the selected objectives + *encode* the selected objectives differently + *reconfigure* (use different ways of visualization) + *connect* solutions in different visualizations]

3.3.6 Detect correlations

A part of learning about trade-offs is detecting correlations between different (pairs) of objectives. To support in this, different types of visualization techniques can be used together with different representations of the PO solutions to be visualized. Combining information from different visualizations is aided through linking.



Examples

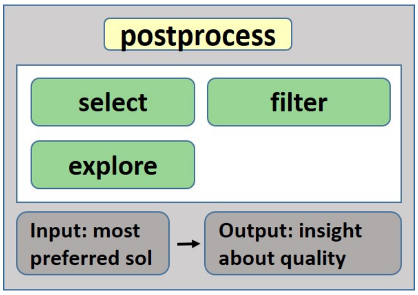
- *"I want to find out whether these two objectives are correlated"*

  [decide on two objectives of interest and *reconfigure* accordingly + *encode* the selected objectives differently + sort solutions based on objective values (*reconfigure*) + *reconfigure* (use different ways of visualization) + *connect* solutions in different visualizations]

- *"I want to learn what happens to the other objectives if I improve this objective function"*

  [decide the objective to be improved + *filter* to get solutions having better values for this objective function + *encode* the selected objectives differently + *reconfigure* (use different ways of visualization) + *connect* solutions in different visualizations]

3.3.7 Post-process the most preferred solution



In many decision making tasks it is important to take into account the quality of the most preferred solution, e.g., with respect to sensitivity, robustness or uncertainty. This task is optional and depends on what kind of information the optimization algorithms used can provide. If the algorithms directly provide the needed information, then this task deals with analyzing the information given. On the other hand, if the algorithms used do not produce any additional information about the quality, then analysis can be done manually, e.g., as described in the example below. The relevant low-level tasks here are *select, filter* and *explore.*

Example

- *"Is a selected solution sensitive to small changes in decision variable values?"*

  [*select* a solution + *filter* to find neighboring solutions in the decision space (if no solutions close enough, *explore* new area) + ***compare*** neighboring solutions in the objective space to get information about quality]

4 Example of implementing task-based visual interactions

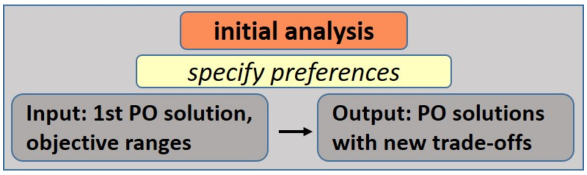
Next, we demonstrate how a DM can be supported in some of the high-level tasks identified in Section 3.3 by using visual interaction techniques. We focus on illustrating the usage of visual interaction techniques, not the design of a graphical user interface as such. Illustrations of visual interaction techniques in this section are produced with the ComVis software (Matkovic, Freiler, Gracanin, & Hauser, 2008).

In the following, we consider an example of how a real interactive solution process could be visually supported in case of a real-world problem with five objective functions related to wastewater treatment plant operation (Hakanen, Sahlstedt, & Miettinen, 2013). The objectives are to 1) minimize total nitrogen in effluent, 2) minimize consumption of aeration energy, 3) minimize methanol dose, 4) minimize dried sludge accumulation, and

5) maximize biogas production. The entire solution process is described in the above-mentioned reference.

Here, we present some parts of the solution process divided in three parts: the initial analysis, the learning phase, and selecting the final solution (i.e. the decision phase). A visual representation of each part including the relevant high-level tasks is included and they follow the same format as the ones used to represent high-level tasks in Section 3.3. The yellow color represents high-level tasks as before, and the red color denotes the part of the solution process. In addition, input and output for the part is also shown. In what follows, the objectives are named $f_1 - f_5$ instead of the real names for the sake of brevity.

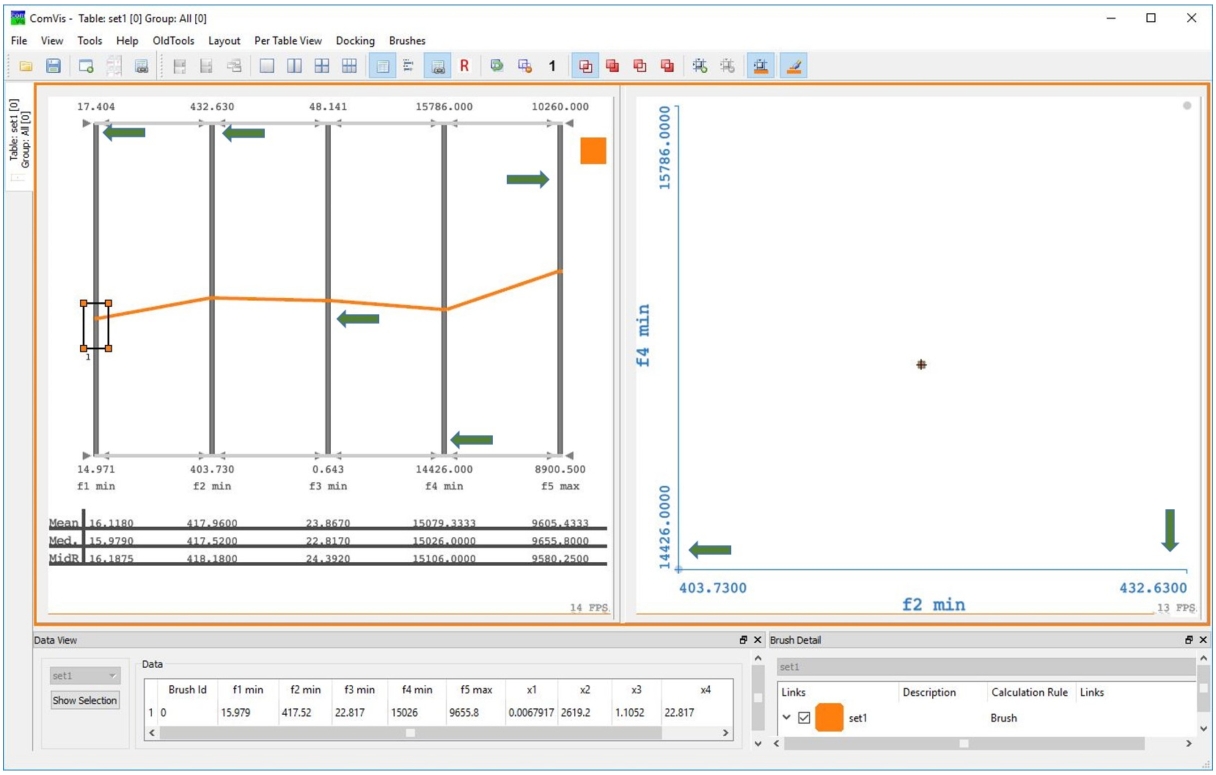
4.1 Initial analysis


initial analysis
specify preferences
Input: 1st PO solution, objective ranges
Output: PO solutions with new trade-offs

In the initial analysis, the relevant high-level task is *specify preferences*. At the beginning of the solution process, the DM can see the first current PO solution that is used as a starting point and the ranges of the objective function values that are reachable within the PO set, as shown in Figure 5.

As can be seen, the PO solution is visualized here by using two different visualization techniques, namely, a parallel coordinates plot (left) and a two-dimensional scatter plot (right). Note that the number of views shown can also be higher than two. The parallel coordinates plot shows the ranges each objective can reach within the PO set. This can be useful information for the DM when specifying preferences. At the moment, the DM has decided to view the objectives f_2 and f_4 in the scatter plot. However, they can be changed at any time, if needed. Since people have different abilities to interpret visualizations, the DM can configure the views by deciding to use such views that she finds the most informative. If desired, the colors used to show objectives and/or different PO solutions can also be changed.

The first four objectives are to be minimized while f_5 is to be maximized. There are options for visualizing this. In Figure 5, this is shown in the views with text min or max after the objective names and real values of f_5 are shown. Instead, function values could be converted into negative ones corresponding to minimizing $-f_5$. Another alternative would be to use the actual values but change the orientation of the axis such that smaller values would be on top and larger values at bottom. In this way, the trade-offs between



Mean	16.1180	417.9600	23.8670	15079.3333	9605.4333
Med.	15.9790	417.5200	22.8170	15026.0000	9655.8000
MidR	16.1875	418.1800	24.3920	15106.0000	9580.2500

set1

Show Selection

Data

	Brush Id	f1 min	f2 min	f3 min	f4 min	f5 max	x1	x2	x3	x4
1	0	15.979	417.52	22.817	15026	9655.8	0.0067917	2619.2	1.1052	22.817

set1

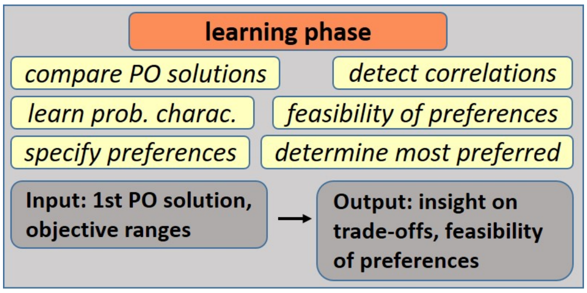
Links	Description	Calculation Rule	Links
☑ 🟧 set1		Brush	

Figure 5: Initial analysis. The starting point of an interactive solution process is visualized. Here, this PO solution is shown to the DM in two different ways. The preferences given by the DM are shown as green arrows.

objectives to be maximized and minimized could be seen more easily. The DM can be given options whether she prefers to see converted objective function values or reverted axis to enable analyzing the parallel coordinates plot (and other visualizations) or the original function values for the understandability of their magnitudes.

The DM must now figure out how to *specify preferences* to improve the current PO solution. This directly corresponds to one of the high-level tasks. The PO solution is in this case visualized by using a parallel coordinate plot and a two dimensional scatter plot projection of f_2 and f_4, as already mentioned, while also showing numerical values of the objectives and the decision variables (at the bottom). The DM can specify preferences (in this case, as a reference point) by interacting with any of the views available and the given preferences are shown in all of them (as green arrows) because they are linked.

4.2 Learning phase



learning phase

compare PO solutions

detect correlations

learn prob. charac.

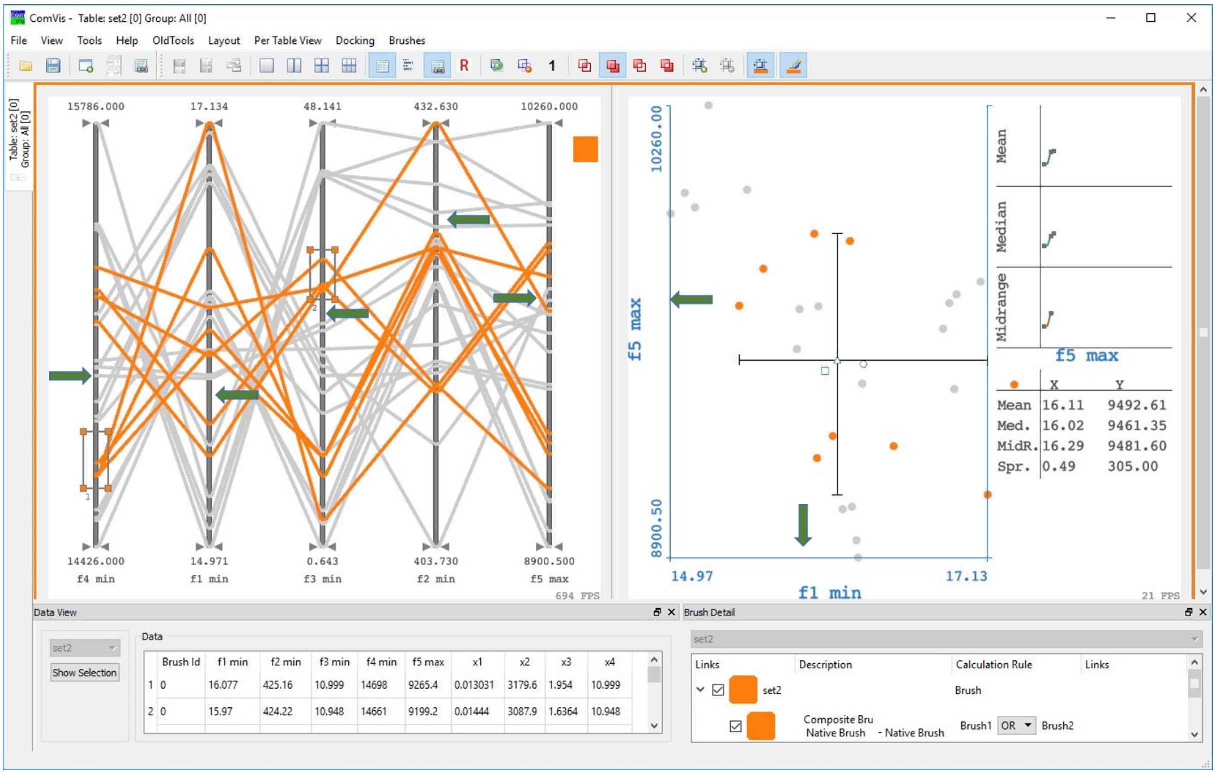
feasibility of preferences

specify preferences

determine most preferred

Input: 1st PO solution, objective ranges → Output: insight on trade-offs, feasibility of preferences

Typically, at the beginning of a solution process, the DM wants to explore what kind of trade-offs exist between the objectives. The aim is to learn about the problem, its possibilities and limitations. Therefore, the most relevant high-level tasks are *compare PO solutions, detect correlations, learn about problem characteristics, check feasibility of preferences*, and *specify preferences*. Furthermore, *determine most preferred solution* is used as a way to select a solution to continue with, if exploration is continued.



	X	Y
Mean	16.11	9492.61
Med.	16.02	9461.35
MidR.	16.29	9481.60
Spr.	0.49	305.00

	Brush Id	f1 min	f2 min	f3 min	f4 min	f5 max	x1	x2	x3	x4
1	0	16.077	425.16	10.999	14698	9265.4	0.013031	3179.6	1.954	10.999
2	0	15.97	424.22	10.948	14661	9199.2	0.01444	3087.9	1.6364	10.948

Links	Description	Calculation Rule	Links
☑	set2	Brush	
☑	Composite Bru Native Brush - Native Brush	Brush1 OR Brush2	

Figure 6: Learning phase. PO solutions generated based on DM's preferences are analyzed to learn about the problem characteristics.

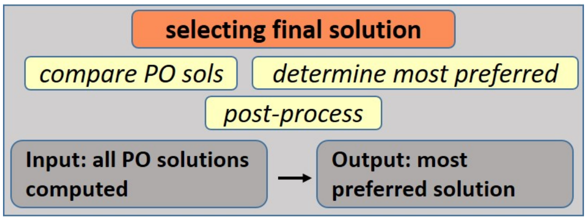
In Figure 6, a set of PO solutions has been generated based on the preferences the DM has given after the initial analysis. Compared to Figure 5, the DM has changed the order of the objective functions in the parallel coordinate plot. This is because one can more easily study trade-offs between the objectives that are next to each other, which should support *detecting correlations* that potentially exist between the objectives and *learning* *about problem characteristics* in general.

Now, *comparing PO solutions* also becomes important since there are more solutions available. As a part of this comparison, the DM has used filtering by applying two different brushes (*Brush1* and *Brush2*) in the parallel coordinates plot for the objectives f_4 and f_3, respectively. The PO solutions that remain within either of the brushes (OR operation as described in Section 3.1) are shown in the orange color, while the ones that do not, have the grey color. The same coloring is also shown in the scatter plot.

The DM has changed the objectives shown in the scatter plot to be f_1 and f_5. Furthermore, statistics (mean, median and midrange) about the solutions satisfying the filtering are shown numerically next to the scatter plot and the ranges of those solutions graphically in the scatter plot (a combination of a vertical and a horizontal line in the scatter plot representing the ranges of the solutions shown in orange).

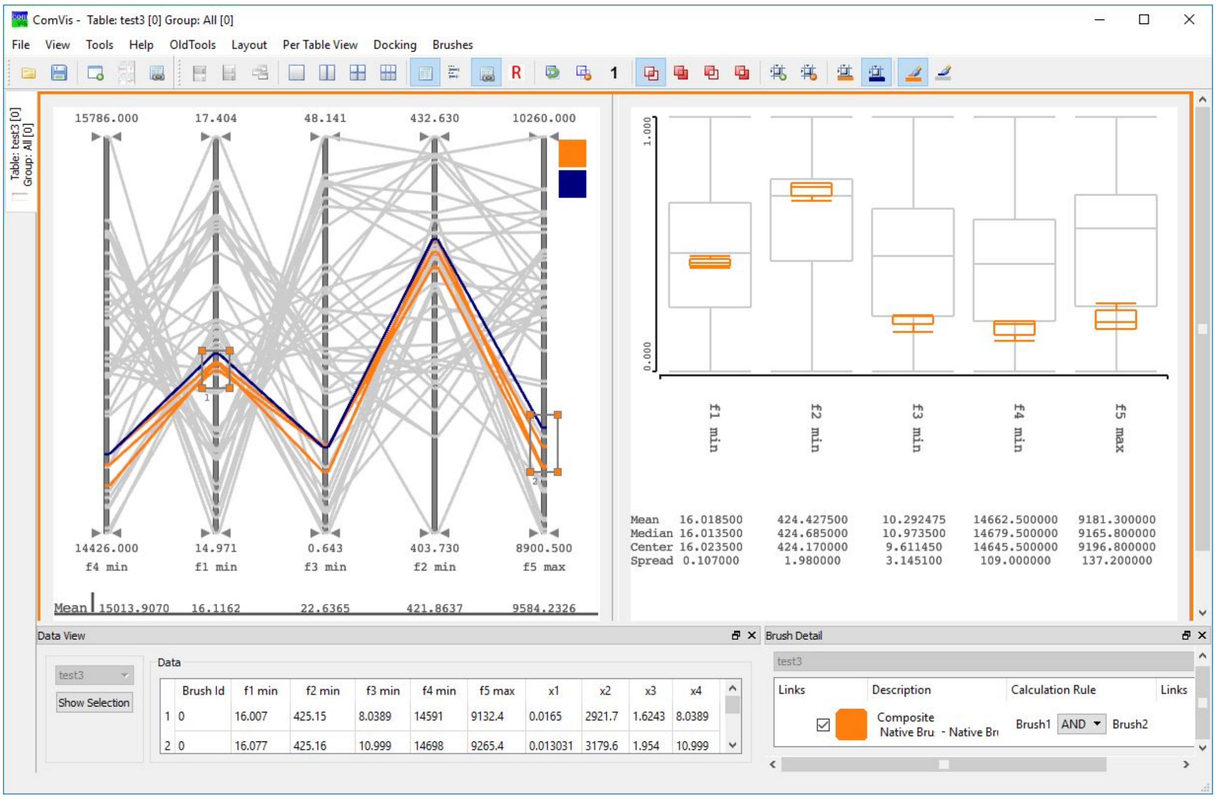
When the DM wants to explore new solutions during the learning phase, she has to *select a most preferred solution* to continue with and *specify new preferences* to search for improved solutions. Again, the preferences are denoted by green arrows in Figure 6. After having explored potential PO solutions enough, the DM is ready to move to the last part of the solution process and select the most preferred solution.

4.3 Selecting the final solution



After the DM has explored different PO solutions sufficiently, she can finally identify a region of interest to fine-tune the solution in it and eventually select the most preferred one. The idea of the decision phase is to convince the DM that no better solutions exist, which makes her trust more in the solution she identifies as the most preferred one. In this example, the selection of the most preferred solution is done within the set of PO solutions generated so far. In other words, no new PO solutions are generated anymore. Therefore, the most relevant high-level tasks are *compare PO solutions* and *determine most preferred PO solution*. Typically, the most preferred solution has to be thoroughly analyzed and then, *post-processing the most preferred solution* becomes relevant as well.

As shown in Figure 7, when analyzing all the PO solutions generated so far, the DM has used filtering within the parallel coordinates plot to identify interesting solutions.



	f1 min	f2 min	f3 min	f4 min	f5 max
Mean	16.018500	424.427500	10.292475	14662.500000	9181.300000
Median	16.013500	424.685000	10.973500	14679.500000	9165.800000
Center	16.023500	424.170000	9.611450	14645.500000	9196.800000
Spread	0.107000	1.980000	3.145100	109.000000	137.200000

	Brush Id	f1 min	f2 min	f3 min	f4 min	f5 max	x1	x2	x3	x4
1	0	16.007	425.15	8.0389	14591	9132.4	0.0165	2921.7	1.6243	8.0389
2	0	16.077	425.16	10.999	14698	9265.4	0.013031	3179.6	1.954	10.999

Links	Description	Calculation Rule	Links
☑ 🟧	Composite Native Bru - Native Bri	Brush1 AND Brush2	

Figure 7: Selecting the final solution. The most preferred PO solution is determined among the set of PO solutions generated. After that, it is further post-processed, if necessary.

She has this time applied brushes to f_1 and f_5 using the AND operator (as described in Section 3.1) which highlights in orange color only those PO solutions that satisfy both the brushes. When compared to Figures 5 and 6, the DM has replaced the scatter plot by a box plot having individual box plots for each objective. The filtering is also visible in the box plot (in orange for the PO solutions satisfying the brushes and in grey for all the PO solutions).

The DM decides to stop the solution process here and has to *determine the most preferred solution*. In addition, she might want to *post-process the final solution*. The solution selected by the DM as a final solution is shown in dark blue color in the parallel coordinate plot.

Note that in addition to visualizing only the objective function values, the values of the decision variables could also be visualized. This could be done either numerically, graphically or, in specific applications, by visualizing the actual design/solution that they correspond with. This is not shown here, but can be optional for the DM when selecting which views to have. If the number of decision variables is large, then some dimension reduction techniques may be needed since, typically, the number of decision variables is higher than the number of objectives.

5 Recommendations

In this section, we present some recommendations for the developers of interactive multiobjective optimization methods on how to implement the interaction between a method and a DM. The recommendations are based on the high-level tasks from Section 3.3, demonstrations of their implementations shown in Section 4, and best practices from visual analytics.

First of all, as with all interactive systems, implementations should be responsive, i.e., a DM's action should result in some change in the views without too long a delay. Card, Robertson, and Mackinlay (1991) describe three human time constants which cannot be changed, and which have to be met. One of them is an immediate response time constant. According to it, a human can make an unprepared response to some stimulus within about a second. We use it in every-day human-to-human communication. In a dialog, a pause of more than a second interrupts the communication. We should, thus, strive to provide (at least partial) response to DM's requests in about a second.

Secondly, we highly recommend using coordinated multiple views that are often used in visual analytics, as already mentioned in Section 3.1. They can support a DM in analyzing PO solutions in a more versatile way than individual visualizations, especially when problems have more than three objective functions. As the number of objectives increases, the cognitive capabilities of a DM are stretched and the ability to inspect PO solutions from different perspectives (even seeing some projections to lower dimensions) may help.

Based on what was recommended above and the examples shown in Section 4, we propose here a minimal set of views that can be used in implementing interactive multiobjective optimization methods. The set contains two different views for visualizing solutions (in our example, a parallel coordinate plot and a scatter plot were used). Optionally, the set of views may also contain a view for post-processing (which in our example was a box plot). Furthermore, we recommend including a view for showing the numerical values of the solutions considered (objective function and decision variable values) in case more accurate understanding of the solutions is required. If more views are needed, it should be possible to add new ones.

Special attention should be paid to the interaction and usability. In case of interactive methods, a brush interpretability plays an important role. With a parallel coordinate plot, the most simple and intuitive brush is the one enabling a selection of a range on an axis. Another possibility, which is probably more suitable for special cases, is an angular brush (Hauser, Ledermann, & Doleisch, 2002). It selects only lines of a specific slope range in a parallel coordinate plot. It certainly offers more possibilities, but is harder to interpret.

In what follows, we provide some more tips on brushing. In case of a two-dimensional scatter plot, a free-form selection (as available in most image editing software), or a polygonal brush are easy to implement. A rectangular brush with axis aligned edges may seem restricting when a selection of point clusters is needed, but it is clear how to interpret such a brush. The DM immediately sees what is selected, a range on the x-axis and a range on the y-axis. A possibility to exactly specify brush ranges, as well as some means to increase reproducibility help adoption of the linking and brushing methods.

Composite brushing makes an iterative drill-down possible. A DM gradually reduces the potential range of solutions in several views. There are basically two possibilities how to implement composite brushing. One takes the current selection and combines it with a brush using a Boolean operator. Then, the process continues. Another possibility is to use a feature specification language as proposed by Doleisch, Gasser, and Hauser (2003). The main idea is to provide a user interface which allows an exact mathematical specification of the brushes' operations. While very powerful, the system is more complicated to use and has a steeper learning curve.

As mentioned in the introduction, our objective is to build bridges between the fields of multiobjective optimization and visual analytics. The recommendations above strive mostly from visual analytics and should lower the threshold of implementing interactive methods that are convenient and informative for DMs to apply.

6 Conclusions

In this paper, we have studied how visual interaction techniques studied in visual analytics can be utilized in implementing interactive multiobjective optimization methods. The interaction between an interactive method and a human decision maker benefits highly from applying visual means to support the DM's tasks.

We have identified seven high-level tasks of a DM that are typical of interactive solution processes. To facilitate them, we have selected nine low-level tasks proposed in the visual analytics literature. They match best the needs of applying interactive methods.

As a concrete example, we have considered a real five-objective optimization problem and demonstrated how the low-level tasks can support the DM in her high-level tasks in different phases of the interactive solution process.

This consideration is hoped to support the development of implementations of interactive multiobjective optimization methods and making them more widely available. Because of the iterative nature of interactive methods, various types of visualizations play an important role in supporting the DM in finding the most preferred solution. In this, findings of the visual analytics literature are of high value. We decided to focus on visual interaction techniques but also user interface design plays an important role in making methods convenient to apply.

To take this research further, some future research ideas are identified. First of all, a practical implementation of an interactive method that utilizes the ideas of this paper is needed. Secondly, getting feedback from real-world DMs on the above mentioned implementation when solving a real-world decision making problem would give ideas for further development. Lastly, it would be interesting to study how the ideas of this paper would work when applied to interactions using e.g. touch screens that are nowadays widely used in different mobile devices and whether they present new challenges in visual interaction for interactive multiobjective optimization. This is closely related to a novel research area called *immersive analytics*.

Acknowledgements

The authors wish to thank MSc Suvi Tarkkanen for initiating interest in the visual analytics field. This research was supported by the Academy of Finland (grant no 311877) and is related to the thematic research area DEMO (Decision Analytics utilizing Causal Models and Multiobjective Optimization, `jyu.fi/demo`) of the University of Jyväskylä. VRVis is funded by BMVIT, BMDW, Styria, SFG and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (854174) which is managed by FFG.

References

Amar, R., & Stasko, J. (2004). A knowledge task-based framework for design and evaluation of information visualizations. In *Ieee symposium on information visualization (2004)* (pp. 143–150). IEEE.

Berger, W., & Piringer, H. (2010). Interactive visual analysis of multiobjective optimizations. In *Proceedings of 2010 ieee symposium on visual analytics science and technology* (p. 215-216). IEEE.

Bernold, G., Matkovic, K., Gröller, M. E., & Raidou, R. G. (2019). preha: Establishing precision rehabilitation with visual analytics. In B. Kozlíková, L. Linsen, & P.-P. Vázquez (Eds.), *Proceedings of the eurographics workshop on visual computing for biology and medicine* (pp. 79–89). The Eurographics Association.

Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991). The information visualizer, an information workspace. In *Proceedings of the sigchi conference on human factors in computing systems* (pp. 181–186). New York, NY, USA: ACM.

Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2001). Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, L. Jain, & R. Goldberg (Eds.), *Evolutionary multiobjective optimization. theoretical advances and applications* (p. 105-145). Springer.

Doleisch, H., Gasser, M., & Hauser, H. (2003). Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the symposium on data visualisation 2003* (pp. 239–248). Aire-la-Ville, Switzerland: Eurographics Association.

Eskelinen, P., Miettinen, K., Klamroth, K., & Hakanen, J. (2010). Pareto Navigator for interactive nonlinear multiobjective optimization. *OR Spectrum*, *32*, 211-227.

Gardiner, L. R., & Steuer, R. E. (1994a). Unified interactive multiple objective programming. *European Journal of Operational Research*, *74*(3), 391–406.

Gardiner, L. R., & Steuer, R. E. (1994b). Unified interactive multiple objective programming: an open architecture for accommodating new procedures. *Journal of the Operational Research Society*, *45*(12), 1456-1466.

Gresh, D. L., Rogowitz, B. E., Winslow, R. L., Scollan, D. F., & Yung, C. K. (2000). WEAVE: A system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *Proceedings of the ieee visualization conference (vis 2000)* (pp. 489–492).

Hakanen, J., Sahlstedt, K., & Miettinen, K. (2013). Wastewater treatment plant design and operation under multiple conflicting objective functions. *Environmental Modelling & Software*, *46*, 240–249.

Hartikainen, M., Miettinen, K., & Klamroth, K. (2019). Interactive nonconvex Pareto Navigator for multiobjective optimization. *European Journal of Operational Research*, *275*(1), 238–251.

Hauser, H. (2006). Generalizing focus+context visualization. In G.-P. Bonneau et al. (Ed.), *Scientific visualization: The visual extraction of knowledge from data* (pp. 305–327). Berlin, Heidelberg: Springer.

Hauser, H., Ledermann, F., & Doleisch, H. (2002). Angular brushing of extended parallel coordinates. In *Proceedings of the ieee symposium on information visualization (infovis 2002)* (p. 127-130).

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

Heer, J., & Schneiderman, B. (2012). Interactive dynamics for visual analysis. *ACM Queue*, *10*(2), 30–55.

Hettenhausen, J., Lewis, A., & Mostaghim, S. (2010). Interactive multi-objective particle swarm optimization with heatmap-visualization-based user interface. *Engineering Optimization*, *42*(2), 119–139.

Hwang, C.-L., & Masud, A. S. M. (1979). *Multiple objective decision making – methods and applications: A state-of-the-art survey.* Berlin, Heidelberg: Springer.

Ibrahim, A., Rahnamayan, S., Martin, M. V., & Deb, K. (2016). 3d-radvis: Visualization of Pareto front in many-objective optimization. In *2016 ieee congress on evolutionary computation (cec)* (pp. 736–745). IEEE.

Jaszkiewicz, A., & Branke, J. (2008). Interactive multiobjective evolutionary algorithms. In J. Branke, K. Deb, K. Miettinen, & R. Slowinski (Eds.), *Multiobjective optimization: Interactive and evolutionary approaches* (pp. 179–194). Berlin: Springer.

Jaszkiewicz, A., & Slowinski, R. (1999). The 'light beam search' approach – an overview of methodology and applications. *European Journal of Operational Research*, *113*, 300–314.

Kaliszewski, I. (2004). Out of the mist–towards decision-maker-friendly multiple criteria decision making support. *European Journal of Operational Research*, *158*(2), 293–307.

Keim, D., Andrienko, G., Fekete, J.-D., Görg, C., Kohlhammer, J., & Melançon, G. (2008). Visual analytics: Definition, process, and challenges. In A. Kerren, J. T. Stasko, J.-T. Fekete, C. Görg, J. Kohlhammer, & G. Melançon (Eds.), *Information visualization: Human-centered issues and perspectives* (pp. 154–175). Berlin, Heidelberg: Springer.

Kerren, A., & Schreiber, F. (2012). Toward the role of interaction in visual analytics. In C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, & A. M. Uhrmacher (Eds.), *Proceedings of the 2012 winter simulation conference* (pp. 420:1–420:13). Berlin: IEEE.

Konyha, Z., Lež, A., Matković, K., Jelović, M., & Hauser, H. (2012). Interactive visual analysis of families of curves using data aggregation and derivation. In *Proceedings of the 12th international conference on knowledge management and knowledge technologies* (pp. 24:1–24:8). ACM.

Korhonen, P. (2005). Interactive methods. In J. Figueira, S. Greco, & M. Ehrgott (Eds.), *Multiple criteria decision analysis: State of the art surveys* (pp. 641–661). New York: Springer.

Lee, B., Plaisant, C., Parr, C. S., Fekete, J.-D., & Henry, N. (2006). Task taxonomy for graph visualization. In *Proceedings of the 2006 avi workshop on beyond time and errors: Novel evaluation methods for information visualization* (pp. 1–5). ACM.

Lotov, A. V., Bushenkov, V. A., & Kamenev, G. K. (2004). *Interactive decision maps: Approximation and visualization of Pareto frontier.* Boston: 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: Interactive and evolutionary approaches* (pp. 213–243). Berlin: Springer.

Luque, M., Ruiz, F., & Miettinen, K. (2011). Global formulation for interactive multiobjective optimization. *OR Spectrum*, *33*, 27-48.

Luque, M., Yang, J. B., & Wong, B. Y. H. (2009). PROJECT method for multiobjective optimization based on gradient projection and reference points. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, *39*(4), 864–879.

Matkovic, K., Freiler, W., Gracanin, D., & Hauser, H. (2008). ComVis: A coordinated multiple views system for prototyping new visualization technology. In *Proceedings of the 12th international conference on information visualisation* (p. 215-220).

Miettinen, K. (1999). *Nonlinear multiobjective optimization.* Boston: Kluwer Academic Publishers.

Miettinen, K. (2006). IND-NIMBUS for demanding interactive multiobjective optimization. In T. Trzaskalik (Ed.), *Multiple criteria decision making '05* (pp. 137–150). Katowice: The Karol Adamiecki University of Economics in Katowice.

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

Miettinen, K., Hakanen, J., & Podkopaev, D. (2016). Interactive nonlinear multiobjective optimization methods. In S. Greco, M. Ehrgott, & J. Figueira (Eds.), *Multiple criteria decision analysis: State of the art surveys* (2nd ed., pp. 931–980). New York: Springer.

Miettinen, K., & Mäkelä, M. M. (2006). Synchronous approach in interactive multiobjective optimization. *European Journal of Operational Research*, *170*, 909–922.

Miettinen, K., Ruiz, F., & Wierzbicki, A. P. (2008). Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, & R. Slowinski (Eds.), *Multiobjective optimization: Interactive and evolutionary approaches* (pp. 27–57). Berlin: Springer.

Munzner, T. (2014). *Visualization analysis and design.* Natick: Taylor & Francis.

Nakayama, H., & Furukawa, K. (1985). Satisficing trade-off method with an application to multiobjective structural design. *Large Scale Systems*, *8*, 47–57.

North, C., & Shneiderman, B. (2000). Snap-together visualization: A user interface for coordinating visualizations via relational schemata. In *Proceedings of the working conference on advanced visual interfaces* (pp. 128–135). ACM.

Ojalehto, V., Miettinen, K., & Laukkanen, T. (2014). Implementation aspects of interactive multiobjective optimization for modeling environments: the case of GAMS-NIMBUS. *Computational Optimization and Applications*, *58*(3), 757–779.

Poles, S., Vassileva, M., & Sasaki, D. (2008). Multiobjective optimization software. In J. Branke, K. Deb, K. Miettinen, & R. Slowinski (Eds.), *Multiobjective optimization: Interactive and evolutionary approaches* (pp. 329–348). Berlin: Springer.

Purshouse, R., Deb, K., Mansor, M., Mostaghim, S., & Wang, R. (2014). A review of hybrid evolutionary multiple criteria decision making methods. In *2014 ieee congress on evolutionary computation (cec)* (pp. 1147–1154). IEEE.

Radoš, S., Splechtna, R., Matković, K., Duras, M., Gröller, E., & Hauser, H. (2016). Towards quantitative visual analytics with structured brushing and linked statistics. *Computer Graphics Forum*, *35*(3), 251-260.

Roberts, J. C. (2007). State of the art: Coordinated multiple views in exploratory visualization. In *Proceedings of the fifth international conference on coordinated and multiple views in exploratory visualization (cmv '07)* (pp. 61–71). IEEE.

Ruiz, F., Luque, M., & Miettinen, K. (2012). Improving the computational efficiency in a global formulation (GLIDE) for interactive multiobjective optimization. *Annals of Operations Research*, *197*(1), 47–70.

Steuer, R. E., & Choo, E.-U. (1983). An interactive weighted Tchebycheff procedure for multiple objective programming. *Mathematical Programming*, *26*(3), 326–344.

Stewart, T., Bandte, O., Braun, H., Chakraborti, N., Ehrgott, M., Göbelt, M., . . . Di Stefano, D. (2008). Real-world applications of multiobjective optimization. In J. Branke, K. Deb, K. Miettinen,

& R. Slowinski (Eds.), *Multiobjective optimization: Interactive and evolutionary approaches* (pp. 285–327). Berlin: Springer.

Tarkkanen, S., Miettinen, K., Hakanen, J., & Isomäki, H. (2013). Incremental user-interface development for interactive multiobjective optimization. *Expert Systems with Applications*, *40*, 3220–3232.

Thomas, J. J., & Cook, K. A. (Eds.). (2005). *Illuminating the path: The research and development agenda for visual analytics.* Los Alamitos, CA, USA: IEEE Computer Society.

Trinkaus, H. L., & Hanne, T. (2005). knowCube: A visual and interactive support for multicriteria decision making. *Computers and Operations Research*, *32*(5), 1289–1309.

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

Valiati, E. R. A., Pimenta, M. S., & Freitas, C. M. D. S. (2006). A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In *Proceedings of the 2006 avi workshop on beyond time and errors: Novel evaluation methods for information visualization* (pp. 1–6). ACM.

Wang, R., Purshouse, R. C., & Fleming, P. J. (2013). "Whatever Works Best for You"- a new method for a priori and progressive multi-objective optimisation. In R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, & J. Shaw (Eds.), *Evolutionary multi-criterion optimization: 7th international conference, emo 2013, proceedings* (pp. 337–351). Berlin, Heidelberg: Springer.

Weistroffer, H. R., & Narula, S. C. (1997). The state of multiple criteria decision support software. *Annals of Operations Research*, *72*, 299–313.

Wierzbicki, A. P. (1982). A mathematical basis for satisficing decision making. *Mathematical Modelling*, *3*(25), 391–405.

Wierzbicki, A. P. (1999). Reference point approaches. In T. Gal, T. J. Stewart, & T. Hanne (Eds.), *Multicriteria decision making: Advances in MCDM models, algorithms, theory, and applications* (pp. 9-1–9-39). Boston: Kluwer Academic Publishers.

Wierzbicki, A. P., Makowski, M., & Wessels, J. (Eds.). (2000). *Model-based decision support methodology with environmental applications.* Dordrecht: Kluwer Academic Publishers.

Woodruff, M. J., Reed, P. M., & Simpson, T. W. (2013). Many objective visual analytics: rethinking the design of complex engineered systems. *Structural and Multidisciplinary Optimization*, *48*(1), 201–219.

Yi, J. S., Kang, Y., Stasko, J. T., & Jacko, J. A. (2007). Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, *13*(6), 1224–1231.