This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Interactivized: Visual Interaction for Better Decisions with Interactive Multiobjective Optimization

JUSSI HAKANEN[1], SANJIN RADOŠ[2], GIOVANNI MISITANO[1], BHUPINDER S. SAINI[1], KAISA MIETTINEN[1], and KREŠIMIR MATKOVIĆ[2].

[1]University of Jyvaskyla, Faculty of Information Technology, Finland (e-mail: author@jyu.fi)

[2]VRVis Research Center in Vienna, Austria (e-mail: {Rados|Matkovic}@VRVis.at)

Corresponding author: Jussi Hakanen (e-mail: jussi.hakanen@jyu.fi).

This work was partly supported by the Academy of Finland, grant numbers 311877 and 322221, and by the COMET program of the FFG (879730).

ABSTRACT In today's data-driven world, decision makers are facing many conflicting objectives. Since there is usually no solution that optimizes all objectives simultaneously, the aim is to identify a solution with acceptable trade-offs. Interactive multiobjective optimization methods are iterative processes in which a human decision maker repeatedly provides one's preferences to request computing new solutions and compares them. With these methods, the decision maker can learn about the problem and its limitations. However, advanced optimization software usually offer simple visualization tools that can be significantly improved. On the other hand, current approaches for multiobjective optimization from the visualization community provide superior visualization tools but lack advanced optimization. In this paper, we introduce a new term, interactivize, for integrating interactive multiobjective optimization and interactive visualization and present an interactivized approach supporting decision makers in visually steering interactive multiobjective optimization methods. We integrate state-of-the-art interactive visualization with the process of interactive multiobjective optimization in a visual analytics solution that significantly improves the analysis workflow of decision makers, like comparing selected solutions and specifying new preferences during the iterative solution process. To realize the new interactivized approach, we combine a coordinated multiple views system with DESDEO, an open-source software framework for interactive multiobjective optimization. We demonstrate our interactivized approach on a river pollution problem.

INDEX TERMS Visual analytics, Multiple criteria decision making, Interactive optimization.

I. INTRODUCTION

TYPICAL real-world optimization problems are multiobjective by nature, which means that their optimal solutions represent trade-offs between the conflicting objectives considered [1]. Since the goodness of the solutions is evaluated by multiple conflicting objectives, the number of so-called Pareto optimal solutions, where no objective can be improved without impairing some other one, can be very large (even infinite). Furthermore, these solutions are mathematically incomparable, which means that selecting one to be implemented requires additional information from a human decision maker (DM) to identify the most preferred trade-offs. A DM is an expert in the problem domain, who can express preferences related to the conflicting objectives considered.

There are many ways of finding a satisfactory solution for a multiobjective optimization problem. The term optimization has, however, a different meaning for people with distinct research backgrounds. The visual analytics community has provided many examples of exploration and analysis of complex multiobjective problems, often without actually optimizing but rather selecting from a pre-computed set of solutions (see, e.g., [2], [3]). In other words, the objectives are used to choose the most preferred solution after a sufficient sampling of the decision variable space has been done (a posteriori methods). On the other hand, the multiobjective optimization community advances optimization methods that conduct calculations to systematically find improved values of the objective functions to get Pareto optimal solutions.

In addition, the DM can be involved either before, after, or during the optimization process. Interestingly, although there are visual analytics systems that combine the two approaches (optimization and visual), only a few visual analytics approaches adequately support the multiobjective optimization community. As mentioned, preferences of a human DM in visual analytics approaches are typically asked after the optimization process, which means that many of the precomputed solutions are not used later as they obviously are not practically relevant and, thus, not interesting for the DM.

In this paper, we focus on finding the solution with the most preferred trade-offs by using interactive multiobjective optimization methods [1], [4]. So far, implementations of interactive multiobjective optimization methods have mostly concentrated on how optimization is performed, and the interaction between the DM and the method has received very little attention [5]. The term interactive here means that a human DM steers the solution process interactively and iteratively by evaluating the solutions computed so far and giving preference information so that the solutions to be generated in the following iteration would be more preferred. It is important to stress out that the methods are called interactive because a user (i.e., a DM) interacts with the optimization method itself in an iterative process. Usually, the interaction with the method is simple (e.g., including a text terminal or a single view). Thus, interactive multiobjective optimization provides an application area for visual analytics, which has not yet been properly studied [5], [6]. To avoid confusion with a DM interacting with visualizations, we use the term interactive visual analysis when we mean the latter one.

We introduce a new approach, where interactive visual analysis is integrated into the process of interactive multiobjective optimization. In other words, existing solutions are evaluated and new solutions are computed iteratively, fully utilizing visualization techniques to support the DM in steering the solution process. We introduce the term interactivize here to reflect including the interactive nature from both the optimization and the visualization fields. Therefore, we interactivize interactive multiobjective optimization and, thus, we call this an interactivized approach. As the method combines computational methods with human-in-the-loop approaches, and relies heavily on interactive visualization it represents a typical visual analytics system. Visual analytics is "the science of analytical reasoning facilitated by interactive visual interfaces" [7]. It combines strengths of humans and computers in order to solve complex problems which are hard to formally specify and solve automatically. Our contribution is fourfold and can be summarized as follows:

1) A widely applicable interactivized approach for a DM to steer the multiobjective optimization process towards the most preferred solution.
2) Detailed task abstraction and requirements for design of the new interactivized approach that are applicable in many domains.
3) Suggestions for improvements of visualization techniques in order to support the interactivized approach by efficiently specifying preferences, exploring a set of Pareto optimal solutions, and visualizing additional metrics that help in decision making.
4) Demonstrating a possible realization of the new approach by integration of a coordinated multiple views (CMV) system with an open-source interactive multiobjective optimization framework. Comparison to a state-of-the-art interactive multiobjective optimization approach.

We provide detailed tasks abstraction and requirements analysis for interactivized interactive multiobjective optimization. These tasks and requirements can be used in any domain when designing and implementing an interactive multiobjective optimization system. We also demonstrate our interactivized approach through a usage scenario dealing with a decision problem related to the pollution of a river [8]. When compared to a state-of-the-art interactive multiobjective optimization system, our approach enables the DM to analyze the optimized solutions in a more comprehensive way by using enhanced CMV systems. Therefore, specifying preferences for improving the solutions already computed becomes easier and more focused through learning about already computed solutions. Note that trade-offs between Pareto optimal solutions become harder and harder to comprehend with an increasing number of objectives to be optimized.

In our interactivized approach, we rely on the optimization algorithm to suggest only solutions that are close to the DM's preferences to save computational resources and, at the same time, reduce the cognitive load in evaluating a huge number of solutions. In other words, we allow the DM to focus on analyzing practically relevant solutions to find the most preferred one. Ultimately, this helps the DM better understand the consequences of the decisions in relation to the conflicting objectives.

The structure of the paper is the following. First, the basics of multiobjective optimization in brief is given in Section II and the related work is described in Section III. The proposed approach is then introduced in section IV. A visual analytics system design for our interactivized approach is introduced in Section V, and an example implementation is described in Section VI. The proposed interactivized approach is demonstrated in Section VII with a river pollution problem, and the lessons learned are discussed in Section VIII. Finally, Section IX concludes the paper and gives some future research ideas.

II. MULTIOBJECTIVE OPTIMIZATION

In the following section, we provide background information on multiobjective optimization and describe basic workflow and corresponding tasks for interactive multiobjective optimization.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

A. BASICS OF MULTIOBJECTIVE OPTIMIZATION

We consider multiobjective optimization problems of the following form:

$$\begin{array}{ll} \text{minimize} & \{f_1(\mathbf{x}), \ldots, f_k(\mathbf{x})\} \\ \text{subject to} & \mathbf{x} \in S \end{array} \tag{1}$$

with k (≥ 2) (conflicting) objective functions $f_i : S \rightarrow \mathbf{R}$ to be minimized simultaneously. The decision variable vectors (or decision variables, for short) $\mathbf{x} = (x_1, \ldots, x_n)^T$ belong to the nonempty feasible set $S \subset \mathbf{R}^n$ defined by constraint functions in the decision space. Besides decision vectors, we are interested in objective vectors consisting of objective (function) values $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T$ in the objective space and the image of the feasible set is called a feasible objective set. Because the dimension of the objective space is often lower than that of the decision space, special attention is usually paid to visualizing objective vectors (see, e.g., [1]).

A decision variable vector is called non-dominated within a given set of decision variable vectors if none of the corresponding objective vector components can be improved without impairing any of the others. In that case, we say that the corresponding objective vector is non-dominated. If a decision vector is non-dominated within all the feasible decision vectors, it is called Pareto optimal. The ranges of the objective values among the set of Pareto optimal solutions are often shown to the DM as supporting information. These ranges are defined by the ideal and nadir vectors representing the best and worst values for each objective, respectively.

We can classify multiobjective optimization methods according to the role of the DM in the solution process: i) a priori methods, where the DM specifies preferences before optimization, ii) a posteriori methods, where DM specifies preferences after optimization, and iii) interactive methods, where steps of specifying preferences and optimization alternate [1]. In this paper, we concentrate on interactive methods (see, e.g., [1], [4]), where the DM takes an active part in the solution process and directs it with one's preferences to find the most preferred solution. In this paper, an important concept is an iteration of an interactive method which consists of the DM specifying preferences and an optimizer returning solutions reflecting those preferences. Further, an interactive method consists of a series of iterations until the DM has found a most preferred solution. As mentioned, the strengths of interactive methods include computational efficiency when only solutions of interest to the DM are generated and cognitive efficiency while the DM needs to consider only limited amounts of information at a time. Furthermore, an important aspect is the DM learning about the trade-offs involved and the feasibility of preferences, which increases the DM's confidence in the most preferred solution. Because the preferences expressed by the DM during the solution process affect the generated solutions, (s)he is not restricted by the pre-generated solutions as in a posteriori methods. A posteriori methods need a rapidly increasing number of solutions as the number of objective functions increases to get a good representation of the Pareto optimal solutions, and still the set will be sparse.

B. INTERACTIVE MULTIOBJECTIVE OPTIMIZATION

Different types of preferences are used in various multiobjective optimization methods. However, we can characterize many interactive methods by the following steps: a) show general information about the problem to initialize the solution process, b) ask for preference information from the DM, c) generate solution(s) reflecting the preferences and show them to the DM, and d) continue from b) as long as the DM wishes. The basic interactive multiobjective optimization workflow is illustrated in Figure 1.

Interactive methods differ from each other, e.g., in the types of preference information they apply (see, e.g., [1], [4], [9]). In this paper, we apply so-called reference points. By setting their components, the DM indicates desirable values for each objective function. The reference point is then included in optimization when new solutions are computed, thus emphasizing those solutions that have objective values close to the reference point. It is important to note that since a DM can freely specify values for the reference point, there might not exist any solution that matches those due to the conflicting objectives. Thus, the reference point may not coincide with any of the solutions computed. Reference points have also been found to be cognitively easy to understand by the DMs since they are directly related to the objective function values [10].

A workflow for an iterative solution process can be seen to include four phases [9]. It starts with problem formulation, where the DM is also involved, but it is not in the scope of this paper. Then in the learning phase, the DM experiments with different preferences to see what kind of solutions are available and learns about the problem and the feasibility of one's preferences. In the decision phase, the DM fine-tunes the region of interest identified so far to pin down the most preferred solution. Note that it is not always possible to determine when the learning phase ends and the decision phase begins since the main difference is on the variation of the provided preference information (wider in the learning phase, narrower in the decision phase), which might not be easy to quantify precisely. The decision phase can be followed by a post-processing phase where the most preferred solution is further tested before implementation.

When realizing an interactive multiobjective optimization approach it is essential to support various tasks. There has not been much work in the literature on task abstraction for interactive multiobjective optimization. In the paper [11], six most important interaction techniques to be considered by the interface designer were identified for a method called Pareto Navigator [12]. These were on how to i) select the starting point, ii) elicit preference information, iii) visualize the navigation progress, iv) examine the set of approximated Pareto optimal solutions, v) ask for real Pareto optimal solutions and how to examine them, and vi) support the overall decision

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

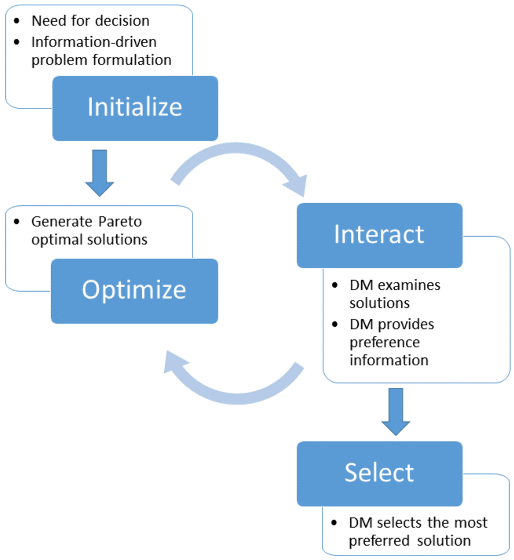

FIGURE 1: Basic workflow of interactive multiobjective optimization.

making process. However, the tasks iii, iv and v were specific to Pareto Navigator.

A more general approach was used in [5], where a task abstraction for interactive multiobjective optimization in general was presented. It consists of seven high-level tasks that the DM faces when solving a multiobjective optimization problem using an interactive method: compare Pareto optimal solutions, specify preferences, check feasibility of preferences, determine a most preferred solution in a subset of alternatives, learn about problem characteristics, detect correlations, and post-process the most preferred solution, e.g., in terms of sensitivity, robustness or uncertainty. The authors illustrated how these high-level tasks can be facilitated by nine low-level visual interaction techniques presented in the visual analytics literature. A taxonomic categorization into low-level and high-level tasks allows for a translation of user tasks into appropriate visual forms and representations. The previous work characterizing visualization usage has focused on low-level tasks (see, e.g., [13], [14]) and high-level tasks (see, e.g., [15], [16]), while only some have concentrated on both levels (see, e.g., [5], [17]). It is important to note that the tasks identified in [5] were not used in any visual design but only an illustration with an existing, general visualization system is given.

III. RELATED WORK

Our idea is to combine visualization and optimization. We review how visualization has typically been used in multiobjective optimization. We also present previous studies on how multiobjective optimization, together with visual analytics, have both been used for decision making.

A. USING VISUALIZATION IN MULTIOBJECTIVE OPTIMIZATION

In multiobjective optimization, visualization can be used in different ways to support the DM. A commonly used way is to visualize individual solutions or solution sets in the objective space when the DM needs to compare solutions [18], [19]. This approach is very typical in a posteriori methods, where the DM is expected to choose the most preferred one among a (large) set of pre-computed solutions. The Attribute Explorer [20] and the Influence Explorer [21] represent pioneering uses of interactive visualization in such a setup, i.e., in selection from a set of pre-computed multidimensional items, with no possibility to initiate a generation of new items from the visualization. Visualization of the solutions becomes much more challenging when the number of objectives increases since the dimensionality of the data to be visualized increases [22]. Moreover, the number of solutions needed to have a good representation of all the Pareto optimal solutions increases exponentially with respect to the number of objectives.

Using visualizations to support decision making is especially important in the context of interactive multiobjective optimization methods, where the DM is actively involved in the solution process. A typical way of using visualizations in interactive methods is to have a single type of visualization, which the DM can interact with (see, e.g., [23]–[31]). More versatile options for visualization are available in commercial process integration and design optimization software like modeFrontier and Optimus. However, they lack implementations of interactive multiobjective optimization methods and are not openly available to use the visualization tools together with external optimization methods.

B. VISUAL ANALYTICS FOR MULTIOBJECTIVE OPTIMIZATION

Visual analytics is, in particular, an interactive and iterative dialog between the analyst and the system [32], through which the analyst makes observations and derives useful information to help him/her understand the phenomena investigated [33]. This is in line with what a DM does by using interactive methods during an iterative solution process of finding the most preferred Pareto optimal solution. Sometimes, the analyst and the DM can be the same person but not necessarily always. In case they are different persons, the role of the analyst is to support the DM.

Visual analytics has been applied to many domains in order to analyze large and complex data. Parameter space exploration is probably the sub-field of visualization, which is mostly related to multiobjective optimization. Here, the term parameter is used instead of variable when compared to the definition given in Section II-A. For example, paper [34] provides a conceptual framework for visual parameter space analysis. In many related applications, after sampling the parameter space (by means of well-known design of experiments [35] methods), the analyst visually explores computed solutions and chooses the satisfactory ones.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

In cases when the parameter space is large (with respect to the number of parameters), it is often not feasible to sample the whole space densely in advance. To help in this, so-called Hybrid Visual Steering is introduced in the paper [36], where automatic optimization based on regression models and interactive visual simulation steering are combined in order to guide the analyst through a high dimensional parameter space. The main focus in Hybrid Visual Steering is on supporting analysts in parameter space sampling and communication of uncertainty of regression models for simulation ensembles. Further, although the method computes new cases, new data is provided as-is, it is not optimized and is, thus, clearly different from our focus. In the paper [37], the authors analyze trade-offs, uncertainty, and sensitivity in the fisheries industry by means of visual analysis. They use visualization to understand trade-offs of a set of pre-computed simulations. Finally, the parameter space of decision trees themselves has been explored [3], [38]. In both cases, a pre-computed set of decision trees, a well-known method in decision making, has been analyzed, and the best performing one selected. Although the user can guide the sampling of the tree generation parameters in the paper [3] to investigate local sensitivities, actual optimization algorithms are not utilized as we do.

So far in the literature, only preliminary steps have been taken in studying the potential of applying visual analytics for multiobjective optimization. We start analyzing the previous work with visual steering, where DM's preferences were used to search for the most preferred solution [39]. Although it was not using optimization and relied only on sampling the decision space, interactive visual analysis was used to support the DM in giving preferences that were used to guide further sampling. In the paper [40], a many-objective visual analytics (MOVA) framework was proposed as a new approach to the design of complex engineered systems. MOVA emphasizes learning through optimization problem reformulation, enabled by visual analytics and a posteriori multiobjective optimization with a large number of objectives. In [11], ideas from visual analytics were used in an incremental user interface development for an interactive multiobjective optimization method Pareto Navigator [12], and the task analysis done was specific to that.

The SAGESSE decision support methodology for exploring multidimensional spaces used coordinated multiple views to support DMs in analyzing solutions with multiple objectives and allowed the DM to guide optimization for improved solutions [41]. Finally, connections between visual analytics and interactive optimization were studied in [6], [42]. In [6] this was done from the perspective of brachytherapy. The authors derived a problem-solving loop that they connected to the sense-making loop commonly used in visual analytics to understand tasks. Their approach is application dependent and very much tied to brachytherapy except the very general problem-solving loop.

In [42], the authors concentrate on complex bus networks, and proposed a visual analytics system for selecting the best bus routes with respect to several objectives based on a real-time model that generates Pareto optimal solutions. Their approach has three stages where they i) explore to find a non optimal route in the current network, ii) manipulate to generate several new Pareto optimal routes to replace the non optimal one, and iii) evaluate the new generated routes to select the best one. In the second stage, the user can specify constraints that are taken into account in generation and, in the third stage, the user can specify ranges for objectives when selecting the best one. Differences to our paper are that this is a posteriori approach where user preferences are involved only in the final selection process but can not be used to control which kind of Pareto optimal routes are generated.

Compared to the studies mentioned above, our interactivized approach emphasizes the DM steering the optimization process towards the most preferred solution while learning from already computed solutions and one's own preferences. The closest of the mentioned studies to our work is SAGESSE [41], where parallel coordinates have also been used as the main view. The main differences are: i) The preference information assumed from the DM in SAGESSE (selecting one objective to be optimized and giving bounds for other objectives) is combined with a fixed method for computing new Pareto optimal solutions (the ϵ-constrained method). Our approach uses reference points as preference information, but how these are used in optimization is not fixed, enabling the usage of diverse optimization methods for generating improved solutions. ii) There is no support in SAGESSE for evaluating the progress of the DM's preference information during the solution process. iii) SAGESSE is not compared to other interactive multiobjective optimization methods. iv) The authors do not provide a task abstraction or requirements that could be used in other domains.

IV. INTERACTIVIZED INTERACTIVE MULTIOBJECTIVE OPTIMIZATION

In this section we present our novel approach to interactive multiobjective optimization. We introduce a comprehensive task abstraction with corresponding requirements which can be used by interactive multiobjective optimization researchers and system developers. Today, multiobjective optimization is used in many scientific, engineering, or commercial domains. Many of the solutions used in different domains will certainly require some customized visualizations, but all of them will have to provide solutions for tasks and requirements identified here. Thus, our main goal is to provide guidelines which will lead to improved interactive multiobjective optimization systems. As we rely on interactive visualization as the main mean of interaction between a DM and an optimizer, we call our approach Interactivized multiobjective optimization. The workflow of the interactivized approach is shown in Figure 2. When compared to the basic workflow of interactive multiobjective optimization shown in Figure 1, the interactivized one includes interactive visual analysis as an integrated part.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

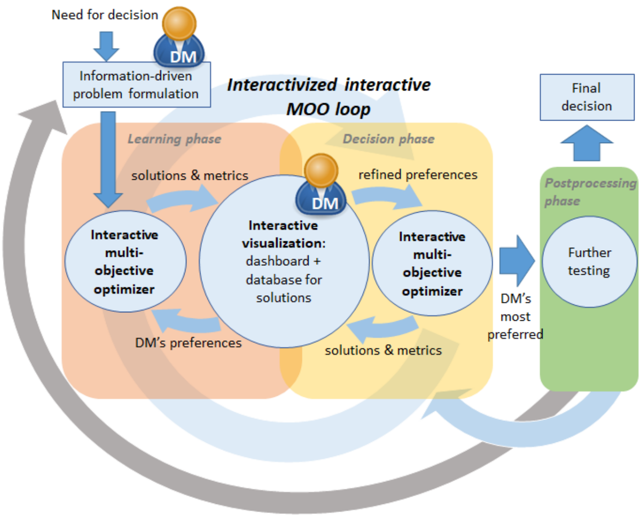

FIGURE 2: Workflow of interactivized interactive multiobjective optimization, including the learning, decision, and post-processing phases. The workflow starts from recognizing a need for optimization followed by a problem formulation (DM involved). Then, new solutions are generated by optimization and evaluated by the DM in the interactivized interactive multiobjective optimization loop. The workflow is not linear, but one can go back to all the previous phases if needed (getting back to problem formulation is not supported by our approach and, thus, indicated with a gray arrow).

In order to design the new interactivized approach, we followed a participatory design process in close cooperation between the authors of this paper (visualization experts and experts of multiobjective optimization). Our design process can be structured in three phases: inception, design, and evaluation. During the inception phase, we had numerous meetings to better understand the multiobjective optimization experts' tasks and communicate possibilities for visual analysis. The inception phase resulted with task abstraction for interactivized approach. The design phase then led to the detailed requirements for design of an exemplary interactivized system. Finally, the implemented system has been evaluated in the evaluation phase.

A. TASKS

We build on the tasks presented in [5]. We use all of them here except the task post-process the most preferred solution, since it is highly dependent on the optimization problem considered. In addition, we introduce two new important tasks of evaluating the progress of the DM's evolved preferences during the iterative solution process and analyzing similar solutions for detecting local improvements. We present the tasks in a more general form (compared to [5]) to allow connecting them to tasks in other domains [43]. In addition, in contrast to [5], where only the task analysis was performed, we here generate requirements from the tasks, use them in visual analytics system design, and compare the resulting interactivized approach to a state-of-the-art interactive multiobjective optimization framework.

We propose the eight tasks shown in Table 1. We give them a short name, a general description applicable to other domains, and a specific description of how they are applied in interactive multiobjective optimization. We also describe the tasks by answering the six questions as suggested in [43]: WHY is a task pursued, HOW is a task carried out, WHAT does a task seek, WHERE in the data does a task operate, WHEN is a task performed, and WHO is executing a task? Next, we briefly describe these tasks.

In the first task, the DM compares solutions produced by the optimization algorithm during the iterative solution process. Since the solutions represent trade-offs between the objectives (and cannot be ordered without preferences), comparing them is not a trivial task and relies on the expertise of the DM in the problem domain. The second task specifying preferences is based on the analysis of the solutions computed so far. The preferences given by the DM are used to steer optimization in finding improved solutions or solutions of different performance and, thus, can be used to scan the decision space by using optimization. The preferences can be totally new, a refinement of the last one, or any of the earlier ones. When new solutions are computed based on the preferences, the DM is able to detect correlations between the conflicting objectives, gain insight, and learn about other characteristics of the problem like trade-offs or one's own preferences. When the DM specifies preferences iteratively, it is also possible to analyze the progress of the preferences and see, e.g., whether they are converging or not. After seeing a promising solution, the DM can analyze similar solutions to detect local improvements. At the end of the solution process, the DM needs to choose the most preferred solution for implementation or further testing.

B. REQUIREMENTS

After the tasks were defined in the inception phase, the design phase then led to the following detailed requirements from the experts based on numerous prototypes, which were validated together:

- **R1:** Compare multidimensional solutions from one iteration, and between iterations (Task1).
- **R2:** Compare solutions in the context of selected reference points (Task1).
- **R3:** Compare a selected subset of best ranked solutions (Task1).
- **R4:** Quickly specify new preferences or adjust earlier ones, and fine-tune them if needed (Task2).
- **R5:** Easily change perspective between a single iteration and multiple iterations (Task3).
- **R6:** Easily change visual mapping in order to scan solution characteristics (Task3).
- **R7:** Provide means to visualize decision variables in linked views (Task3).
- **R8:** Analyze dependencies between the values of conflicting objectives (Task4).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

TABLE 1: Task abstraction. Each task is given a name, both a general and a specific description and is classified with respect to six questions.

Name	General description	Specific description	Why is a task pursued?	How is a task carried out?	What does a task seek?	Where in data is task operating?	When is a task performed?
1. compare	compare multidimensional vectors	compare Pareto optimal solutions	compare the solutions shown to gain insight	visually analyze the solutions shown, no metric available	identify preference order within the solutions shown	shown solutions	during the solution process
2. specify	guide sampling for new data points	specify new preferences for steering	tell optimizer how solutions should be improved	interact with visualizations or provide numerical values	find preferences to improve already computed solutions	preferences	at the start and during the solution process
3. scan	sample for new attribute/feature values not existing before	compute new solutions in the unexplored areas in the objective/decision space	find solutions in unexplored areas	specify preferences corresponding to potential unexplored areas of the solution space	see solutions in potential areas of the solution space that are not yet computed	all solutions	during the solution process
4. detect	detect correlations between features	detect correlations between conflicting objectives	learn about correlations between the objectives	visually analyze different subsets of solutions available	understand the nature of correlations among the objective functions	all solutions	during the solution process
5. learn	gain insight from data	learn about problem characteristics	learn what kind of solutions are available for the problem	visually analyze different subsets of solutions available and the corresponding preferences given	understand the potential and limitations of solutions	all solutions and preferences	during the solution process
6. progress	analyze progress of a sequence of multidimensional data points	analyze progress of DM's preferences during the iterative solution process	see how preferences have changed during the solution process	visually analyze the changes in preference history	understand how preferences have evolved	history of preferences	during the solution process
7. analyze	analyze neighboring solutions to detect local improvements	analyze similar solutions to detect local improvements	find local improvements for interesting solutions	visually analyze a given cluster of solutions; use additional metrics	understand availability of local improvements	cluster of solutions	during the solution process
8. choose	analyze a subset of data points and choose best	determine a most preferred solution in a subset	find the most preferred solution	use visualizations in the objective/decision spaces and additional metrics to select the most preferred solution	select the most preferred solution	potential solutions	during or at the end of solution process

- **R9:** Analyze how the DM's preferences have changed through the iterations of the solution process (Task6).
- **R10:** Assess the neighboring solutions of a selected candidate by expanding the corresponding cluster. (Task7)
- **R11:** Reduce the set of candidates for the most preferred solution, provide details for them, and choose the most preferred solution (Task8).

These requirements represent a basis for the visualization design. There are no specific requirements for Task5, learn, as the whole new approach supports gaining insight by amplifying cognition through interactive visual analysis.

V. VISUAL ANALYTICS SYSTEM DESIGN

In this section, we describe an exemplary visual analytics system design consisting of optimization and visualization tools. The design is motivated by analysis tasks and requirements described above. In the next section, we provide details of coupling the visualization and the optimization component into a visual analytics system for the interactivized approach. We start by describing the optimization methods used, followed by the visualization design.

A. OPTIMIZATION METHODS

Interactive multiobjective optimization used in the interactivized approach is based on multiobjective evolutionary algorithms (MOEAs). The benefits of MOEAs include that they can be applied to different types of optimization problems and, while being population-based, they can generate a set of solutions instead of a single solution for a single optimization run [44]. However, it must be noted that MOEAs produce only non-dominated solutions within the solution set it generates, and the Pareto optimality of the solutions is not guaranteed. Originally, MOEAs have been developed as a posteriori methods, but some of them also take into account the DM's preferences in generating desired solutions [45]. In this paper, we will use an interactive reference vector guided evolutionary algorithm (interactive RVEA) [46] to generate a set of solutions corresponding to the reference point given by the DM. However, our approach is not limited to this algorithm. Note that the chosen algorithm is able to handle both achievable and infeasible reference points. Further, the interactive and iterative nature of the optimization algorithms used allows the DM to adjust the distribution of the solutions generated in the areas of interest. The number of solutions generated at each iteration depends on the parameters of the interactive RVEA. Typically, the number is quite high (from tens to few hundreds), and the generated solutions can be rather similar due to the reference point, so there is a need to reduce the number before visualizing them to the DM.

One way to select a subset of solutions for visualization is to use clustering, where the given set is divided into (a given number of) clusters consisting of similar solutions. After clustering, the solution closest to a cluster centroid for each cluster can then be chosen for visualization to represent the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

cluster. Thus, at each iteration, the solutions generated by the interactive RVEA are clustered first to reduce the number of similar solutions shown to the DM and, second, to reduce the total number of solutions shown to the DM to a manageable yet representative level. The goal of the clustering is thus to decrease the cognitive burden of the DM. The number of solutions shown (i.e., the number of clusters) can be set based on the DM's desires, or a default number can be set in case the DM does not want to set it. We use here k-means as the clustering method but different clustering methods can be used, instead [47]. If so desired, the DM can see all the solutions for a selected cluster in the main view (described later in this section) to look for local improvement for the chosen solution since none of the solutions in the clusters is deleted.

B. VISUALIZATION DESIGN

In our example we focus on cases with less than a dozen conflicting objectives. Considering more than a dozen conflicting objectives at once is rarely done in practice, as it becomes more and more complicated to comprehend the problem as the number of objectives increases.

In order to provide a good overview of multidimensional solutions, and to make it easy to specify preferences for a new iteration to steer the optimization, we have chosen parallel coordinates as the central view for interaction. On the one hand, parallel coordinates are well-known in the multiobjective optimization domain (see, e.g., [18]) and, on the other hand, they represent a solid basis for answering requirements, especially those related to comparison (**R1-R3**). However, basic parallel coordinates are not sufficient for all identified tasks and requirements. We propose several extensions in the view itself as well as some additional views and corresponding extensions for tasks where they are required. In addition to the extended parallel coordinates, we use a scatter plot matrix, a view to show the progress of preferences, extended box plots view, and several standard views, such as scatter plots, for example. We extend all views used to meet our requirements. All our main design decisions are justified in the following sections.

1) Extended parallel coordinates

In addition to visualizing multidimensional data, parallel coordinates are often used to specify multidimensional input. In our case, the DM specifies preferences as a reference point and, then, the optimization software computes a set of non-dominated solutions "closest" to the reference point. The reference point and the set of solutions are depicted in the parallel coordinates. In order to support requirement **R4**, we provide a quick way of specifying a reference point by clicking corresponding places in the axis of the parallel coordinates, and, if needed, the DM can enter exact values.

In order to assess the neighboring solutions in a cluster (requirement **R10**), we also provide the option to visualize all solutions covered by one representative solution. Figure 3 shows parallel coordinates with the reference point provided

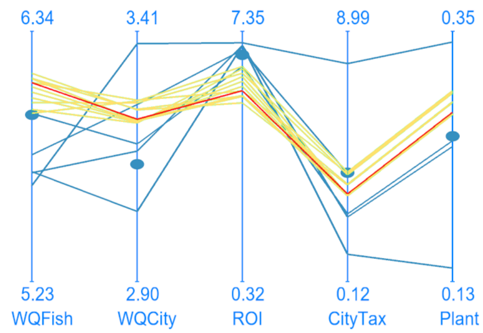

FIGURE 3: Parallel coordinates are used to specify and show components of the reference point (blue dots). They also show representatives of clusters of non-dominated solutions closest to the reference point (blue lines), and, on demand, all non-dominated solutions (orange lines) that belong to the chosen cluster (red line).

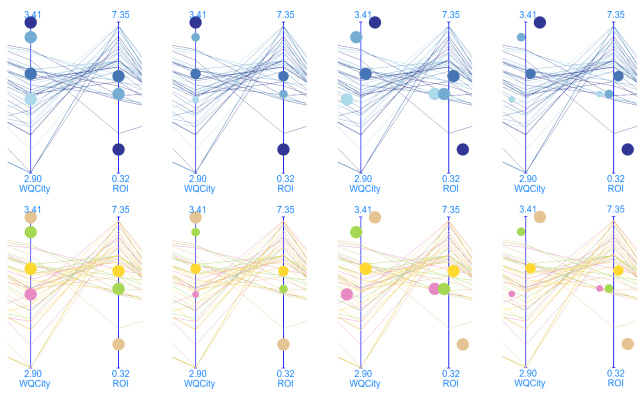

FIGURE 4: Four possibilities to depict temporal evolution of reference points. Top row uses different blue tones (darker is more recent), and bottom row uses different hues. Only color can be used (left), color and point size (left middle), position — points are scattered around axis (right middle), and size and position (right).

by the DM (blue dots on axis), the representative solutions (cluster representatives) as computed by optimizer (blue poly-lines), one of the representative solutions that is selected for a detailed inspection (red poly-line), and all solutions that belong to the selected cluster (orange poly-lines). The minimum and maximum values of the objective functions are determined based on all the solutions computed.

The DM also has the option to see solutions from one or more of the previous iterations. Color coding is used to distinguish individual iterations. We either use constant hue and vary lightness, or we vary hues for different iterations. The ability to see solutions from selected iterations only reduces the cognitive demand, and the DM can concentrate on analyzing smaller subsets of solutions. The ranges of the parallel coordinates are adjusted to accommodate all iterations, and not only the visible ones, to preserve consistency

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

and ease comparison.

Besides differentiating iterations, it is possible to see the temporal evolution of the reference points and solutions. There are several possibilities of how to visualize reference points so that they can be easily ordered to see the temporal evolution. As color hues are not easy to order, it would be better to use lightness or size as a time indicator [48]. We offer the basic choice to use hues or lightness to display reference points (and corresponding solutions). In addition, for each of the two, point size and position can be changed to augment the perception of the temporal evolution. Figure 4 shows all the options for parallel coordinates. Only two axes are shown due to space constraints. In the top row, lightness is used as the main indicator of time, and darker points are more recent. In addition to lightness only, combinations with point size (larger points are more recent), position (points on the right are more recent), or a combination of size and position are shown. The bottom row shows the same variations but for hue as the main indicator of time. Hue only, without size or position, is hardly usable as we cannot order hues. However, if it is combined, the ordering becomes easy. Depending on personal preferences and the task to be solved, DMs can choose different strategies.

Finally, during the optimization process, the DM can select individual solutions as candidates for the most preferred solution. We provide a way of keeping track of these solutions (the DM can save them), and displaying them for comparison at any time during the analysis. Figure 5 illustrates the control pane for the parallel coordinates. The control pane is an important part of the view. Besides color coding, the DM also selects which iterations are shown, can name the iterations, and exactly specify components of a reference point. Finally, the control pane is also used for book-keeping of potential solution candidates. The DM can name and show them at any time.

The new extensions to the parallel coordinate view make it the central interface of interactive multiobjective optimization. All above-described extensions of the parallel coordinates directly support the following requirements: The DM can quickly specify a reference point by clicking the axis in the parallel coordinates (**R4**). The DM can learn about the feasibility of one's own preferences when the given preferences and the corresponding solutions computed are color-coded (**R2, R5, R6, R9**). The DM can assess the neighboring solutions of a selected candidate by expanding the corresponding cluster (**R10**), and the DM can choose the most preferred solution among saved candidate solutions (**R1-R3, R11**).

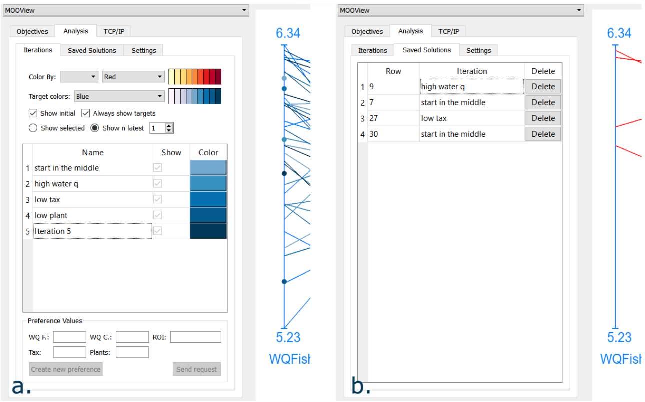

FIGURE 5: The control pane of the parallel coordinate view is an important part of the interface. **a.** The analysis tab shows iterations and their colors. Each iteration can be shown or hidden and the DM can name each solution, specify exact numeric values for reference point components, and set the color coding. **b.** The solutions tab lists all solutions that are selected as interesting.

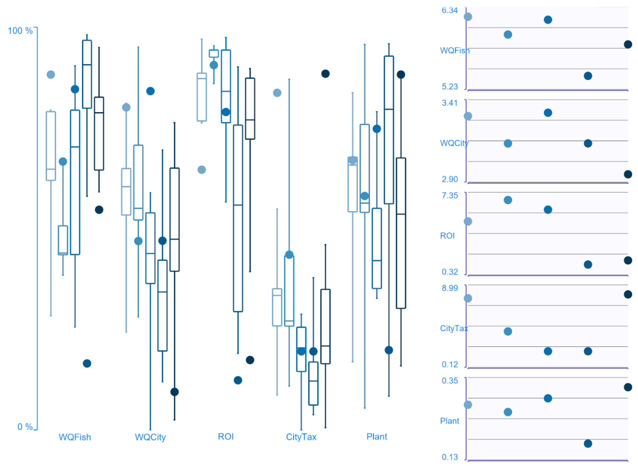

FIGURE 6: Views to show the temporal evolution of reference point components. The box plot view on the left shows them for each iteration along with basic descriptive statistics. The DM can see relations between computed solutions and the reference point. On the right, the preference history view depicts only components of the reference points. Temporal evolution can be better observed by sacrificing information on computed solutions.

2) Additional views

Examination of reference points and their temporal evolution is possible in parallel coordinates, but due to additional information depicted, it is not always easy to perceive it efficiently. We provide two additional views to support this requirement (**R9**). The box-plot view (Figure 6 left) shows reference points and descriptive statistics of all solutions per iteration. The DM can see if there are solutions that are close to the preferences or not. Note that in this view, we see it for individual objectives. In the parallel coordinates, we also see how they are connected, but here we can see a better overview. The second view is called the preference history view and it shows the reference points only (Figure 6 right). It clearly depicts the evolution itself, but it does not show anything about solutions. The color coding is consistent throughout all views, and view parameters (line widths,

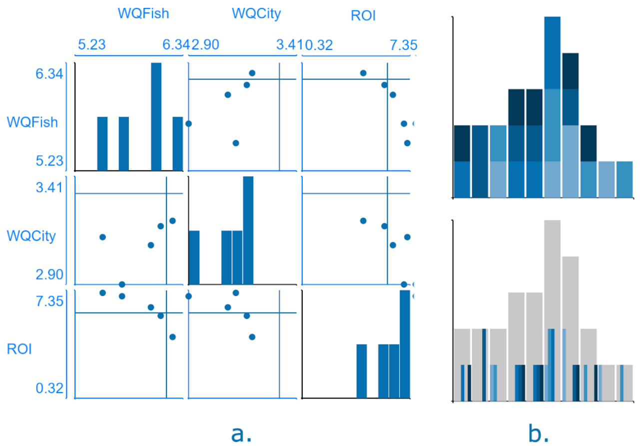

FIGURE 7: **a.** A scatter plot matrix shows solutions for pairs of objectives. The colors are the same as in the extended parallel coordinates. The reference points (circles in the parallel coordinate plots) are shown as horizontal and vertical lines in the individual scatter plots. Here only one iteration is shown. **b.** Two options that can be used to show histograms on the diagonal of the matrix.

colors, captions, etc.) can be configured. There is no ultimate configuration that works all the time, but the DM chooses the right view depending on the task that is being solved.

In addition to parallel coordinates, a scatter plot matrix is a view often used in the optimization community. It shows all possible objective pairs. Besides multidimensional solutions, we also depict reference points in the scatter plot matrix. In order to clearly distinguish them from the solutions shown as points, we use lines to show components of the reference points. As there are two objectives in each scatter plot, we need two axis-aligned lines per scatter plot to show the components (Figure 7a.). Interestingly, we use lines to show solutions and points to show preferences in the parallel coordinates, and exactly opposite in the scatter plot matrix. Such an arrangement particularly supports tasks of detect and learn: Firstly, the scatter plot matrix allows detecting correlations between any pair of objectives without the need of changing anything in the view (**R8**). Secondly, the same color coding used in all the views for different iterations helps gaining insight into the connection between the preferences and the corresponding solutions. The matrix shows only iterations that are selected in the extended parallel coordinates.

On the matrix diagonal, we depict histograms of individual objectives. In order to provide different comparisons, we support two display modes, as shown in Figure 7b. Distributions of individual iterations are shown either on top of each other in a bin , or next to each other. It is easier to compare iteration-wise contribution to bins in the second arrangement, but the individual bars may become narrow in a case when many iterations are shown simultaneously.

In order to get a full insight into the solutions, the DM may also want to visualize the decision variable values in addition to the objective values. For this, we provide standard views such as a scatter plot, a histogram, or parallel coordinates, which can be freely configured (supports **R7**). Alternatively, we can use a problem domain-specific visualization of decision variable values.

As mentioned above, we have realized a solution-based color coding in all the views in order to reduce the cognitive load of the DM and to provide a way to easily relate solutions across all views. The default color coding is based on the iterations, as mentioned before. Further, within a single iteration, we provide the DM an option to use color coding based on additional metrics. Here, we have as examples distance to the related reference point or to the ideal vector. This coloring helps in differentiating solutions within an iteration and, thus, supports the DM in choosing a candidate solution to be saved (**R1, R2, R6, R11**). It is good to note that color coding based on the additional metrics can not be applied for the saved candidate solutions coming from different iterations with different reference points. The video which accompanies the paper shows further configuration possibilities of the interactivized approach.

3) Interaction and linked views

Each of the above-described views suits different requirements and shows different aspects of data. We have already described how the user specifies preferences by clicking on the parallel coordinate axes, or how all cluster elements can be shown on mouse-over. In order to fully exploit the potential of interactive visualizations, we link all the views into a CMV system [49]. The main idea of the CMV system is linking&brushing, that is, to allow the user to select a subset of data in one view (brush) and highlight the brushed subset in all other views (linking). Let us illustrate this idea on a simple example. The user brushes relatively low values of City Tax in the parallel coordinate view by simply drawing a brush on the corresponding axis (Figure 8a.). This action selects a subset of all data. Selected solutions are then highlighted in all other views. In our case, the scatter plot matrix shows the same data as the parallel coordinates, and the scatter plot shows two decision variables. The selection can be refined by another brush.

Figure 8b. shows the case where the user drilled-down by limiting the selection to the high values of two decision variables. The two brushes, one in the parallel coordinates and the other in the scatter plot are combined with the logical AND operation. The solutions that are not brushed are depicted in gray to provide context. The exploration process can be arbitrarily extended by adding additional brushes with the same or different operators and by modifying existing brushes. We support axis brushes in the parallel coordinates and rectangular brushes in the scatter plots. These two types of brushes are easy to interpret and the user can easily specify intervals that define a brush.

In order to support quick inspection of solutions we provide mouse-over highlighting as well. By hovering over views, the solutions under the cursor are highlighted in all views. Figure 9 demonstrates such a case. In addition to the brushing (note that the user moved the scatter plot brush as compared to the previous example) the user hovers over the parallel coordinate plot and the solution under the mouse cursor is highlighted in red in all views. The video which accompanies the paper illustrates linking&brushing more efficiently.

4) Scalability

Increasing the complexity of the data to be visualized increases the difficulty of the decision making process. We identify three sources of complexity in interactive multiobjective optimization workflows: the number of objective functions, the number of iterations, and the number of solutions produced by the optimizer within a single iteration. We tackle the issue of complexity such that our interactivized approach is scalable in all three aspects.

Human cognitive capabilities are limited [50] but vary among DMs. If the number of objectives is higher than the DM can handle at a time, we provide an option to reduce the number of objectives shown in each plot. When it comes to the number of iterations, it naturally depends on the problem considered and the DM. However, it is typical that DMs find a satisfying solution in 3–8 iterations [51]. More iterations may be needed if the DM is focused on exploring different kinds of solutions. For such cases, we provide the DM a functionality to select a subset of iterations, which the DM is interested in, to be displayed in the views. Finally, scalability for the number of solutions is achieved by giving the DM an opportunity to decide how many solutions one wants to consider at a time. Clustering is used for decreasing the number of solutions shown accordingly, as described in Section V-A.

5) Configuring views

All the above-described views are parts of the CMV system. They can be freely configured so the DM can select the views and their placement to be comfortable working with. From our experience, we advise to keep the configuration stable. The DM knows immediately where the views are and what is shown. The cognitive load is reduced, and full attention can be paid to the analysis.

Color selection is another important issue. We enable color coding of the solutions based on iterations or additional metrics within solutions for a single iteration. In such a complex color mapping, it is essential to use consistent color coding across all views.

Based on the visual analytics design described above, and our observations, we provide an example configuration of a CMV system that we here refer to simply as a dashboard for supporting the DM in interactive multiobjective optimization as shown in Figure 11. The dashboard is divided into two main parts: the top part is reserved for various views (a.–e.), and the bottom part provides quantitative information of data (f.). The extended parallel coordinate view (a.) is the central view in the analysis workflow. It is placed at the top left where most of the interaction happens. The scatter plot matrix (b.) next to it provides details on pair-wise correlations. The progress of the preferences (d.), which is not checked all the time, is placed on the far right side. The box-plots (c.) below the scatter plot matrix show descriptive statistics and reference points from selected iterations. Finally, the scatter plot (e.) on the right shows selected decision variables. Consistent color coding is used in all views. We are aware that this particular configuration is not the only one possible. However, we propose it as a starting point for DMs that are not yet familiar with visual analytics.

VI. IMPLEMENTATION

The interactivized approach designed in Section V was implemented by combining the open-source software framework DESDEO [52], [53] for interactive multiobjective optimization methods and the CMV system. Next, we briefly describe our implementation based on these two systems.

The DESDEO software framework is a collection of open-source Python packages, which make it easy for researchers to use interactive multiobjective optimization methods, as well as to develop new ones. These packages have been designed to tackle specific sub-domains of interactive multiobjective optimization. Here, we focus on the desdeo-problem and the desdeo-emo packages. Multiobjective optimization problems are formulated using the desdeo-problem package. This package supports the formulation of problems based on analytical functions, offline data, simulators, or any combination of them. The instances of the problem, created using desdeo-problem, can then be used to solve the problem using the desdeo-emo package. For example, it contains interactive versions of popular evolutionary algorithms such as RVEA [54] and NSGA-III [55]. All methods in this package can be easily experimented with a Jupyter Notebook interface.

The interactive dashboard is implemented as an extension to ComVis [56], a standard CMV system that supports multiple composite brushing and freely configurable views. It is designed for high configurability, i.e., an user can add or remove different views based on their needs. However, we provide a default design with a set of configured views that work well for most problems. It provides an interface to connect to external Python programs that can be used to extend available computational possibilities.

The dashboard designed in Section V runs on a client computer as a stand alone application and communicates with the server. Computations to generate new solutions are performed on a dedicated server by utilizing the DESDEO framework. This allows detaching the computationally expensive routines for solving multiobjective optimization problems from the interactive visual analysis, which can result in a significant speed-up in computation times if a server with powerful processing power is used. Such servers are seldom available to be used through a graphical interface and are often available as headless servers instead. The interactive dashboard is utilized to provide a graphical user interface to these servers, therefore, allowing the DM to easily utilize

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

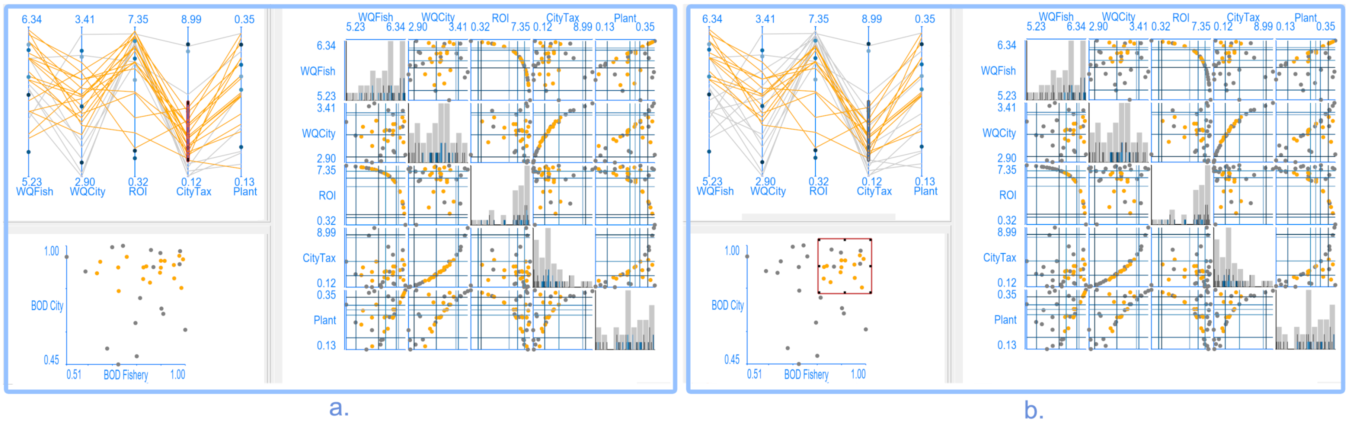

FIGURE 8: *Linking&brushing* is the key concept of the coordinated multiple views. Three views are used for illustration: a parallel coordinate view, a scatter plot matrix, and a scatter plot. **a.** The user draws the axis aligned brush on the City Tax axis in the top left view. All corresponding solutions are highlighted in all views, while the remaining solutions are shown as context. **b.** The user refines the selection by introducing the rectangular brush in the lower left scatter plot. Only solutions that belong to both brushes are highlighted now.

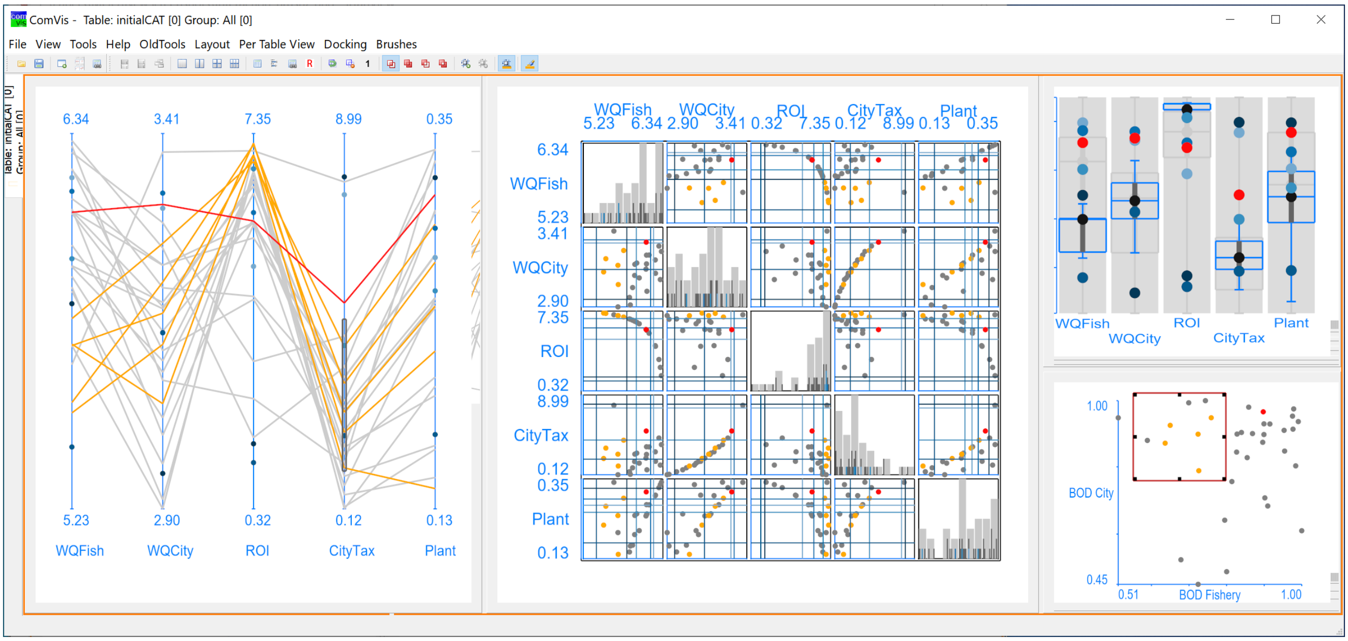

FIGURE 9: In addition to the composite brushing, when the user hovers over the views, the solutions under the mouse cursor are highlighted in red in all views.

the computational power offered by the servers for solving the optimization problems with the various tools available in DESDEO.

The communication between the client and the server happens over a TCP/IP connection. TCP/IP has been chosen because it offers a lossless communication channel, allowing for minimal information loss between the client and the server. The communication protocol consists of simple ASCII character strings delivered between the client and the server over the TCP/IP connection. These messages can then be easily parsed for relevant information on both the client and server sides. For instance, the information includes — but is not limited to — new solutions to the problem (both the objective and decision variable values), and preference information in the form of reference points given by the DM. The connection between a client and a server, alongside an example of a possible message, is shown in Figure 10.

VII. EVALUATION

Next, we evaluate our interactivized approach by solving a multiobjective optimization problem related to river pollution. First, the problem is described, and we refer to it as a usage scenario as suggested in the paper [57]. By this usage scenario, we demonstrate the potential of our approach. We

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

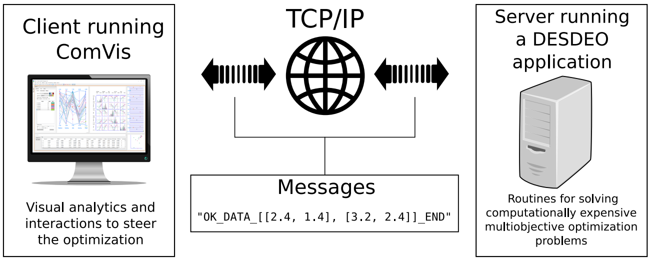

FIGURE 10: The communication between a client and a server over a TCP/IP connection conceptualized. The message shows an example what the server sends the client when new solutions are computed.

also compare it to the DESDEO framework with a Jupyter Notebook interface that represents a state-of-the-art interactive multiobjective optimization tool.

A. PROBLEM DESCRIPTION

The problem considered describes a pollution problem of a river, where a fishery and a city are polluting water [8]. The two decision variables represent the proportional amounts of biochemical oxygen demanding material removed from water in two treatment plants located after the fishery (BOD Fishery) and after the city (BOD City). The problem has five objective functions [58]: 1) maximize water quality in the fishery (WQ Fish), 2) maximize water quality in the city (WQ City), 3) maximize fishery return on interest (ROI), 4) minimize city tax increase (City Tax), and 5) minimize deviation of the treatment plants from optimal operating conditions (Plant). The constraints for this problem are the lower and the upper bounds for the two decision variables. The ideal and nadir vectors of the problem are computed before the solution process and are thus assumed to be known. Methods for computing the ideal and nadir points can be found, for example, in book [1].

B. INTERACTIVE SOLUTION PROCESS

In what follows, we present a usage scenario as an example of the interactive solution process a DM may go through when solving the problem considered utilizing the proposed interactivized approach. From the DM's point of view, this process can be divided into three distinct phases as mentioned in Section IV : (i) the learning phase, (ii) the decision phase, and (iii) the post-processing phase. The first two phases will be described in detail, but the post-processing phase is only briefly discussed at the end of this subsection, as it is not the main focus of our proposed approach.

As already mentioned, we apply the interactive RVEA method [46]. The number of representative solutions the computed solutions are clustered into after each iteration can be set by the DM or a default value can be used. During the solution process, the DM is able to expand any of the representative solutions into the original set of solutions that they represent. This is to further support the DM by supplying additional details about the solutions computed and to provide concrete evidence of the other solutions a representative solution is covering.

The learning phase The solution process started with a learning phase where the DM begins to analyze an initial set of solutions in the objective space. This initial set can be provided a priori, or it can be computed utilizing RVEA (which we used in this example). The ideal vector was used as a reference point when computing the additional metrics for the initial set of solutions. In this case, the DM chose the number of solutions to be shown at each iteration to be five, which was automatically set as the number of clusters for k-means. Alongside the computed objective vectors, the server running the DESDEO software also computed the L_2 (Euclidean) and the L_∞ (Chebyshev) distances to each of the solutions from both the ideal vector and the reference point. These distances functioned as additional metrics to aid the DM in analyzing the available solutions. Additionally, each solution's decision variable values were returned as well since the DM may wish to incorporate information about those in the decision making process.

The dashboard shown to the DM throughout the solution process in this usage scenario can be seen in Figure 11. The solutions computed in different iterations together with reference points given by the DM are shown in Figure 12.

First iteration. The DM started the solution process by analyzing the initial set of solutions. In our usage scenario, the DM initially chose components of the reference point for the water qualities of the fishery and city close to their respective ideal values and the remaining three components close to the middle of the objective ranges. The ranges are shown on the top and bottom of the axes visible in the MOO view seen in Figure 11a and also in Figure 12. The reference point chosen by the DM in the first iteration was $[6.24, 3.35, 3.55, 4.58, 0.24]$. By choosing high values for the water qualities, the DM wished to see what kind of objective values are achievable for the other objectives when the environmental perspective is emphasized.

Second iteration. The DM analyzed the new solutions computed by interactive RVEA based on the given reference point. From the scatter plot matrix shown in Figure 11b, the DM noticed that substantial gains can be achieved in ROI by still keeping the water qualities somewhat high. Therefore, the DM chose the next reference point to be similar to the previous one, but with a higher value for ROI and slightly lower values for the water qualities. The values of the reference point chosen by the DM were $[5.90, 3.10, 4.18, 2.81, 0.17]$. By doing so, the DM wished to find a solution with better economic consequences. The desirable values for water qualities were also impaired in the second reference point to allow for a wider variety of solutions to be computed with a higher ROI. This was done because the DM knows that the qualities are in conflict with ROI.

Third iteration. From the new solutions, the DM noticed that the fishery's water quality could be improved from what was given in the previous reference point while still keeping

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

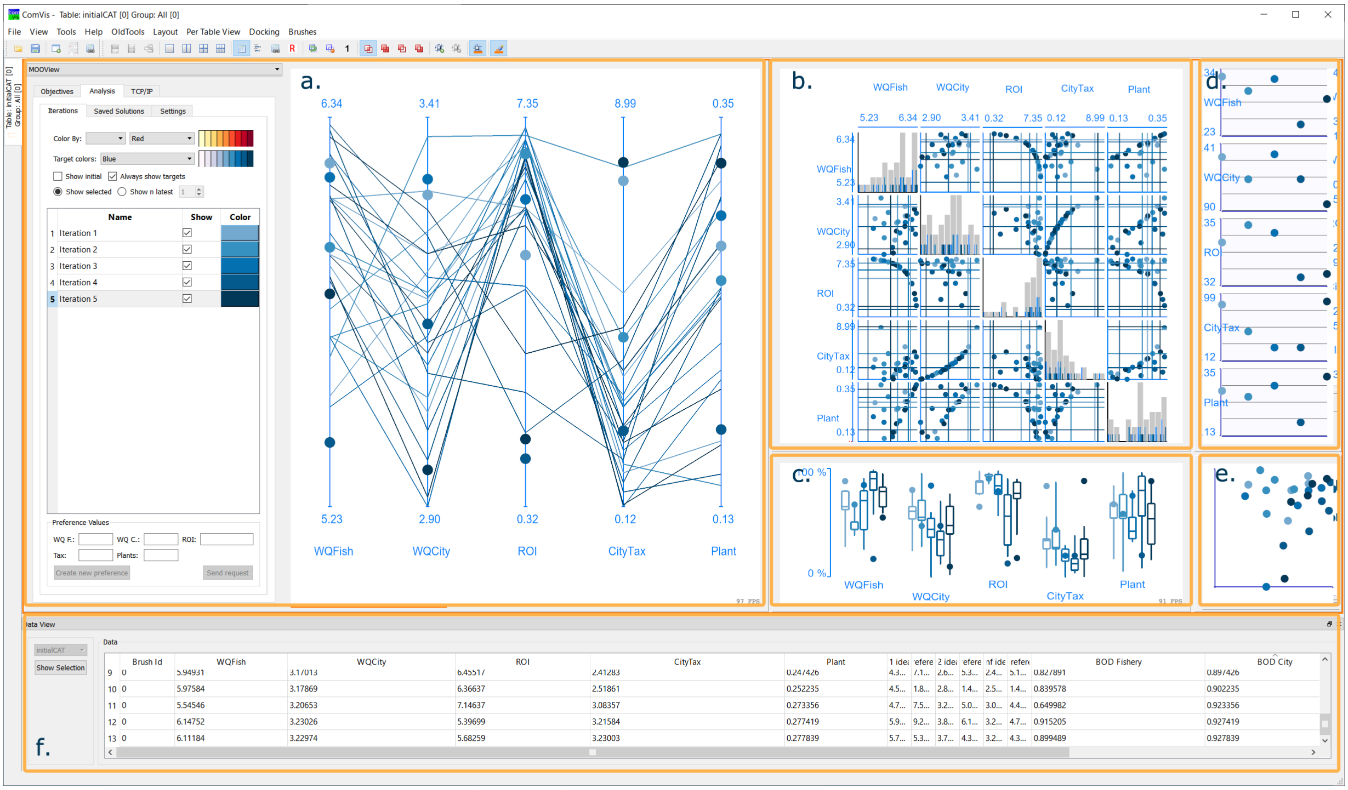

FIGURE 11: A dashboard to support solving a multiobjective optimization problem in an interactivized way. In an iterative solution process, a DM steers computation of new Pareto optimal solutions by giving preferences using the extended parallel coordinates. The DM can analyze solutions and one's own preferences by using CMVs (a. extended parallel coordinates, b. scatter plot matrix, c. extended box plot, d. preference history view, e. scatter plot, and f. numerical values) and focus on finding the most preferred trade-offs since all the solutions are candidates for the final one. The optimization server, running in the background, is not seen here.

ROI high. The DM was satisfied with the trade-offs found between the first three objectives and wished now to explore the last two objectives, the city tax increase and the deviation from the optimal operating conditions in the treatment plants. The DM wanted to explore the available solutions with high water quality, a low city tax increase, and a low deviation from the optimal operating conditions. Thus, the DM set the ROI to a low value to allow the water quality to reach high values, the city tax increase to reach low values, and the operating conditions to deviate as little as possible from the optimal value. The reference point chosen by the DM was $[6.34, 3.41, 1.73, 5.46, 0.27]$. After seeing the solutions computed in the third iteration, the DM felt satisfied with the variety of the available solutions and started to converge toward a final decision.

Moreover, during the learning phase, the DM also had the option to save solutions of interest for later consideration. This facilitates the comparison of solutions during or after the learning phase by allowing the DM to focus on a smaller subset of solutions, which have already been regarded as interesting.

The decision phase All the different solutions computed during the iterations in the learning phase were stored alongside the given reference points. The DM was now able to compare them to find the most satisfying solutions leading to a final decision. The DM may also explore the preferences given and their relations by using the preference history view, which can help the DM to learn about how the preferences have evolved during the learning phase and potentially even justify the means that led the DM to a certain final decision. Additionally, the decision variable values were also available for the DM to analyze in the scatter plot allowing incorporation of information about the decision variables in the decision phase.

The DM then went through the iterations retrospectively and cross-compared objective vectors of different solutions. Using one of the computed metrics, the DM can choose to color-code solutions based on the selected metric within a single iteration to see, for example, how close a solution is to a given reference point. As said, the promising solutions that the DM had saved during the learning phase can not be colored based on the additional metrics. However, linking and brushing can be used to further reduce the number of saved solutions.

In addition to comparing the solutions saved during the learning phase, the DM is still able to generate new solutions

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

by defining additional reference points as in the three iterations described previously. This need could emerge if the DM wants to see a new solution between some previously saved solutions. These new reference points are, however, not defined in an exploratory fashion by the DM like in the learning phase, but with determination driven by learned characteristics – e.g., the trade-offs between the objectives.

After the DM had finished comparing different solutions, (s)he could reach a final decision. In this example, the final decision had high water qualities, moderate values for ROI and city tax increase, and a high value for the deviation of the plants from their optimal operating conditions (corresponding objective values were $[6.20, 3.05, 3.40, 4.60, 0.31]$). This decision had a clear focus on the ecological impact of the treatment plant's operation while keeping the city tax increase moderate. However, the ROI was low, and the deviation from the optimal operating conditions in the treatment plants itself was high. Further, because the DM had access to a selection of different solution candidates and was able to compare the said solutions, the final decision made could be regarded as having been made in a systematic fashion, where adequate attention had been given to different available candidate solutions.

Post-processing When analyzing the most preferred solution, the DM is able to expand the corresponding cluster. By doing so, (s)he can fine-tune the values of the objectives and find possibly an even better solution than the one (s)he had selected in the decision phase. In this example, the DM decided to explore the individual solutions in the cluster represented by the most preferred solution. The DM found a solution among the expanded cluster, which had slightly better values for the water qualities but also a slightly worse value for ROI. The solution found had the objective values $[6.22, 3.07, 3.40, 4.57, 0.31]$ and was deemed more preferable by the DM when compared to the most preferred solution made originally in the decision phase. Lastly, it is important to note that exploring the solutions in a cluster is possible also during the learning and decision phases and is therefore not restricted to the post-processing phase.

Moreover, because the most preferred solution related to the final decision is generated by an evolutionary optimization method, which is inherently heuristic, the solution may not actually be Pareto optimal, as already mentioned. Hence, the computed solution may be projected to the closest Pareto optimal one or used as an initial reference point in another interactive multiobjective optimization method (e.g. from the desdeo-mcdm package)), which is able to find solutions closer to the true Pareto optimal set.

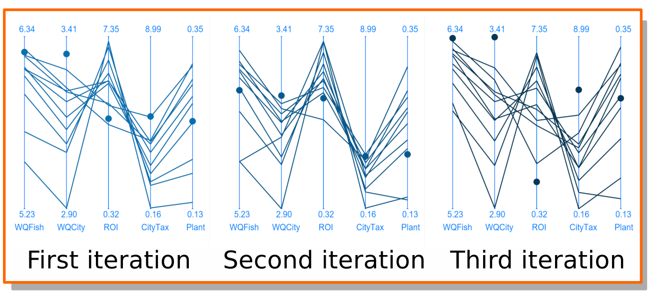

FIGURE 12: Reference points given and the resulting solutions computed in each iteration presented in the usage scenario described.

```
pref.response = pd.DataFrame([[5.4,3.3,7,0.5,0.3]],
                             columns=["WQFish","WQCity",
                             "ROI","CityTax","Plant"])
```

```
plot, pref = evolver.iterate(pref)
figure = animate_next_(
    plot.content['data'].values,
    figure, filename="river.html",
    generation=evolver._iteration_counter)
IFrame("river.html", width=700, height=350)
```

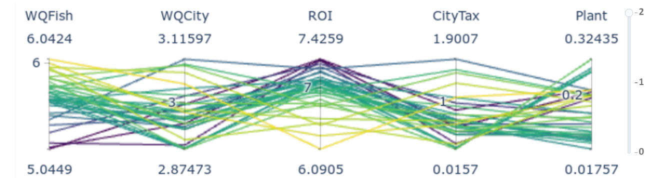

FIGURE 13: The Jupyter Notebook interface. The top cell shows a reference point provided by the DM to interactive RVEA. The bottom cell shows interactive RVEA running for one iteration using the DM's preferences, and creating the animated parallel coordinate view. The DM can see solutions from different iterations by using the slider on the right side of the visualization.

C. COMPARISON TO A JUPYTER NOTEBOOK INTERFACE

A similar interactive process can be conducted using a Jupyter Notebook interface for the DESDEO framework. To use this interface (Figure 13), a DM (or an assisting person) has to be fluent in the Python programming language. This interface provides an easy mechanism to control the various aspects of the optimization process. The parameter values of the optimization methods like interactive RVEA and the optimization method itself can be easily changed in DESDEO. The solutions returned by the method can be visualized in any form desirable to a DM, as long as someone writes the code for the visualization using suitable libraries like D3 or Plotly. Here we use Plotly.

The framework offers functionality to provide preferences and visualize solutions. Preference information can be provided as a reference point in the format of a Pandas dataframe by typing within the Jupyter Notebook interface, as shown in the top cell of Figure 13. The results of each iteration can be viewed as individual frames of an animated parallel coordinate view, shown in Figure 13. The DM can choose to display solutions from individual iterations by moving a slider, and brush solutions in each frame.

The solution process begins in a similar way to using the interactivized approach. The initial set of solutions is visualized as the first frame of the animated parallel coordinate view. However, no other views are created and, thus, no CMVs can be used to assist the DM. Moreover, no additional metrics like distances from the ideal vector or the reference point are visualized or shown in any other way. The solutions are not clustered either, so the DM is provided with all the solutions generated by interactive RVEA. All of these combined may make the interaction process cognitively challenging. Finally, to choose the most preferred solution, the DM may want to see all the solutions in a single view, which is not possible with the provided animated parallel coordinate view, which only shows solutions from one iteration at a time.

VIII. DISCUSSION

As shown in Section VII, our new interactivized approach provides improved ways for completing the tasks introduced in Section IV, and it satisfies the requirements described in Section V-B. We considered our usage scenario both with our interactivized approach as well as with the Jupyter Notebook interface for the DESDEO framework that represents the state-of-the-art in interactive multiobjective optimization. Next, we describe our findings.

The Jupyter Notebook interface can only use a single visualization at a time that the DM can interact with, as already mentioned. Bringing the CMV capabilities to a Jupyter Notebook (or a corresponding implementation of an interactive multiobjective optimization method) would require lots of work and, thus, it is not usually done as well as it should be done. For the other way around, adding modern optimization algorithms to a state-of-the-art visualization software is also not straightforward. Our interactivized approach enables the DM to specify preferences quickly via interactive visual analysis, while in the Jupyter Notebook interface they have to be manually typed as numbers. Finally, interactive visual analysis and computing new solutions with the optimization methods are done in separate machines as described in Section VI, which increases the efficiency of the interactivized approach and makes the iterative solution process fluent for the DM, especially for computationally expensive problems.

When writing this paper, we have taken advice from the reflection paper [57] to avoid many pitfalls in writing design study papers. In that paper, various pitfalls were identified, and we have recognized many of those during the progress of this research. Therefore, we have done our best not to fall into those. For example, among the authors, we have both visualization and multiobjective optimization experts who have collaborated before, and we have been able to identify relevant tasks that DMs face during the iterative solution process and, then, created visual design to answer the requirements arising from the tasks.

During this research, we have verified our hypothesis that visual analytics can provide improved decision support in interactive multiobjective optimization. When applying the new interactivized approach to the usage scenario, clear improvements could be identified when compared to the state-of-the-art in interactive multiobjective optimization. In addition, the task abstraction described in Section IV provided both general and specific descriptions for the tasks so that the visual analytics community can utilize the findings in some other domain than interactive multiobjective optimization. Overall, all four aspects of contribution outlined in the introduction are supported by our interactivized approach and related findings.

IX. CONCLUSIONS

We have proposed an interactivized approach by combining visual analytics and interactive multiobjective optimization to support decision makers in visually steering the iterative solution process in interactive multiobjective optimization. The approach combines a coordinated multiple views system and a multiobjective optimization software framework. It includes several extended views, based on the requirements extracted from the tasks abstracted for interactive multiobjective optimization, together with some basic ones. Special attention is paid to the user interaction. A powerful linking&brushing mechanism together with the mouse-over highlighting and cluster expansion enable efficient exploration, and provide much more possibilities than simple visualizations with isolated views. The new interactivized approach was compared to a state-of-the-art multiobjective optimization system, and the evaluation showed clear improvements, especially on how the decision maker can specify preference information, learn about the problem characteristics, and evaluate the progress of one's preferences. Thanks to visual analytics, the solution process is more comprehensive giving the decision maker better control of it, and better support to learn about the problem's characteristics to make a balanced decision. For future research, we identify challenges of dealing with different types of preference information and more versatile means to support the decision maker in gaining confidence in the most preferred solution found.

ACKNOWLEDGEMENTS

This work is a part of the thematic research area Decision Analytics Utilizing Causal Models and Multiobjective Optimization (DEMO, jyu.fi/demo) at the University of Jyvaskyla. VRVis is funded by BMK, BMDW, Styria, SFG, Tyrol and Vienna Business Agency in the scope of COMET - Competence Centers for Excellent Technologies (879730) which is managed by FFG.

REFERENCES

[1] K. Miettinen. Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston, 1999.

[2] W. Berger and H. Piringer. Interactive visual analysis of multiobjective optimizations. In Proceedings of 2010 IEEE Symposium on Visual Analytics Science and Technology, pages 215–216. 2010.

[3] T. Mühlbacher, L. Linhardt, T. Möller, and H. Piringer. TreePOD: Sensitivity-aware selection of Pareto-optimal decision trees. IEEE Transactions on Visualization and Computer Graphics, 24(1):174–183, 2018.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

[4] K. Miettinen, J. Hakanen, and D. Podkopaev. Interactive nonlinear multiobjective optimization methods. In S. Greco, M. Ehrgott, and J. Figueira, editors, Multiple Criteria Decision Analysis: State of the Art Surveys, pages 931–980. Springer, New York, 2nd edition, 2016.

[5] J. Hakanen, K. Miettinen, and K. Matković. Task-based visual analytics for interactive multiobjective optimization. Journal of the Operational Research Society, 72(9):2073–2090, 2021.

[6] J. Liu, T. Dwyer, K. Marriott, J. Millar, and A. Haworth. Understanding the relationship between interactive optimisation and visual analytics in the context of prostate brachytherapy. IEEE Transactions on Visualization and Computer Graphics, 24(1):319–329, 2018.

[7] J. J. Thomas and K. A. Cook. Illuminating the Path: The Research and Development Agenda for Visual Analytics. National Visualization and Analytics Ctr, 2005.

[8] S. C. Narula and H. R. Weistroffer. A flexible method for nonlinear multicriteria decision making problems. IEEE Transactions on Systems, Man, and Cybernetics, 19(4):883–887, 1989.

[9] K. Miettinen, F. Ruiz, and A. P. Wierzbicki. Introduction to multiobjective optimization: Interactive approaches. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, Multiobjective Optimization: Interactive and Evolutionary Approaches, pages 27–57. Springer, Berlin, 2008.

[10] O. I. Larichev. Cognitive validity in design of decision-aiding techniques. Journal of Multi-Criteria Decision Analysis, 1:127–138, 1992.

[11] S. Tarkkanen, K. Miettinen, J. Hakanen, and H. Isomäki. Incremental user-interface development for interactive multiobjective optimization. Expert Systems with Applications, 40:3220–3232, 2013.

[12] P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen. Pareto Navigator for interactive nonlinear multiobjective optimization. OR Spectrum, 32:211–227, 2010.

[13] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In IEEE Symposium on Information Visualization, 2005. INFOVIS 2005., pages 111–117, 2005.

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

[15] S. K. Card, J. D. Mackinlay, and B. Shneiderman. Readings in Information Visualization: Using Vision to Think. Morgan Kaufmann Publishers Inc., 1999.

[16] Z. Liu and J. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. IEEE Transactions on Visualization and Computer Graphics, 16(6):999–1008, 2010.

[17] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. IEEE Transactions on Visualization and Computer Graphics, 19:2376–2385, 2013.

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

[19] T. Tusar and B. Filipic. 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, 2015.

[20] L. Tweedie, B. Spence, D. Williams, and R. Bhogal. The attribute explorer. In Conference Companion on Human Factors in Computing Systems, CHI '94, page 435–436, New York, NY, USA, 1994. Association for Computing Machinery.

[21] L. Tweedie, B. Spence, H. Dawkes, and H. Su. The influence explorer (video)—a tool for design. In Conference Companion on Human Factors in Computing Systems, CHI '96, page 390–391, New York, NY, USA, 1996. Association for Computing Machinery.

[22] S. Liu, D. Maljovec, B. Wang, P.-T. Bremer, and V. Pascucci. Visualizing high-dimensional data: Advances in the past decade. IEEE Transactions on Visualization and Computer Graphics, 23(3):1249–1268, 2017.

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

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

[25] A.V. Lotov and K. Miettinen. Visualizing the Pareto frontier. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, Multiobjective Optimization: Interactive and Evolutionary Approaches, pages 213–243. Springer, Berlin, Heidelberg, 2008.

[26] R. Wang, R. C. Purshouse, and P. J. Fleming. "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, and J. Shaw, editors, Evolutionary Multi-Criterion Optimization: 7th International Conference, EMO 2013, Proceedings, pages 337–351. Springer, Berlin, Heidelberg, 2013.

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

[28] A. Ibrahim, S. Rahnamayan, M. V. Martin, and K. Deb. 3d-radvis: Visualization of Pareto front in many-objective optimization. In Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC), pages 736–745. IEEE, 2016.

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

[30] K. Miettinen and M. M. Mäkelä. Synchronous approach in interactive multiobjective optimization. European Journal of Operational Research, 170:909–922, 2006.

[31] A. K. A. Talukder and K. Deb. PaletteViz: A visualization method for functional understanding of high-dimensional Pareto-optimal data-sets to aid multi-criteria decision making. IEEE Computational Intelligence Magazine, 15(2):36–48, 2020.

[32] H. Hauser. The iterative process of interactive visual analysis, 2012. Keynote talk at the EuroVA 2012 workshop.

[33] D. Sacha, A. Stoffel, F. Stoffel, B. C. Kwon, G. Ellis, and D. A. Keim. Knowledge generation model for visual analytics. IEEE Transactions on Visualization and Computer Graphics, 20(12):1604–1613, 2014.

[34] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller. Visual parameter space analysis: A conceptual framework. IEEE Transactions on Visualization and Computer Graphics, 20(12):2161–2170, 2014.

[35] R.A. Fisher. The Design of Experiments. Oliver and Boyd, Edinburgh, 1935.

[36] K. Matković, D. Gračanin, R. Splechtna, M. Jelović, B. Stehno, H. Hauser, and W. Purgathofer. Visual analytics for complex engineering systems: Hybrid visual steering of simulation ensembles. IEEE Transactions on Visualization and Computer Graphics, 20(12):1803–1812, 2014.

[37] M. Booshehrian, T. Möller, R. M. Peterman, and T. Munzner. Vismon: Facilitating analysis of trade-offs, uncertainty, and sensitivity in fisheries management decision making. Computer Graphics Forum, 31(3):1235–1244, 2012.

[38] L. Padua, H. Schulze, K. Matković, and C. Delrieux. Interactive exploration of parameter space in data mining: Comprehending the predictive quality of large decision tree collections. Computers Graphics, 41:99–113, 2014.

[39] G. Stump, S. Lego, M. Yukish, T. W. Simpson, and J. A. Donndelinger. Visual steering commands for trade space exploration: User-guided sampling with example. Journal of Computing and Information Science in Engineering, 9(4):044501–1–044501–10, 2009.

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

[41] S. Cajot, N. Schüler, M. Peter, A. Koch, and F. Maréchal. Interactive optimization with parallel coordinates: Exploring multidimensional spaces for decision support. Frontiers in ICT, 5:1–28, 2019.

[42] D. Weng, C. Zheng, Z. Deng, M. Ma, J. Bao, Y. Zheng, M. Xu, and Y. Wu. Towards better bus networks: A visual analytics approach. IEEE Transactions on Visualization and Computer Graphics, 27(2):817–827, 2021.

[43] H. Schulz, T. Nocke, M. Heitzler, and H. Schumann. A design space of visualization tasks. IEEE Transactions on Visualization and Computer Graphics, 19(12):2366–2375, 2013.

[44] J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors. Multiobjective Optimization: Interactive and Evolutionary Approaches. Springer, Berlin, 2008.

[45] R. Purshouse, K. Deb, M. Mansor, S. Mostaghim, and R. Wang. A review of hybrid evolutionary multiple criteria decision making methods. In 2014 IEEE Congress on Evolutionary Computation (CEC), pages 1147–1154. IEEE, 2014.

[46] J. Hakanen, T. Chugh, K. Sindhya, Y. Jin, and K. Miettinen. Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms. In Proceedings of the 2016 IEEE Symposium Series on Computational Intelligence. IEEE, 2016.

[47] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: A review. ACM Computing Surveys, 31:264–323, 1999.

[48] J. Bertin. Semiology of Graphics: Diagrams, Networks, Maps. Esri Press, Redlands, 2010.

[49] Jonathan C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In Gennady Andrienko, Jonathan C. Roberts, and Chris Weaver, editors, Proceedings of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization, pages 61–71. IEEE CS Press, 2007.
[50] G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological review, 63(2):81–97, 1956.
[51] L. R. Gardiner and D. Vanderpooten. Interactive multiple criteria procedures: Some reflections. In J. Clímaco, editor, Multicriteria Analysis, pages 290–301, Berlin, Heidelberg, 1997. Springer.
[52] V. Ojalehto and K. Miettinen. DESDEO: An open framework for interactive multiobjective optimization. In S. Huber, M. J. Geiger, and A. T. de Almeida, editors, Multiple Criteria Decision Making and Aiding: Cases on Models and Methods with Computer Implementations, pages 67–94. Springer, 2019.
[53] G. Misitano, B. S. Saini, B. Afsar, B. Shavazipour, and K. Miettinen. DESDEO: The modular and open source framework for interactive multiobjective optimization. IEEE Access, 9:148277–148295, 2021.
[54] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff. A reference vector guided evolutionary algorithm for many-objective optimization. IEEE Transactions on Evolutionary Computation, 20(5):773–791, 2016.
[55] K. Deb and H. Jain. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, 18(4):577–601, 2014.
[56] K. Matković, W. Freiler, D. Gracanin, and H. Hauser. ComVis: a coordinated multiple views system for prototyping new visualization technology. In Proceedings of the International Conference on Information Visualisation (IV 2008), pages 215–220, Washington, DC, USA, 2008. IEEE.
[57] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. IEEE Transactions on Visualization and Computer Graphics, 18(12):2431–2440, 2012.
[58] K. Miettinen and M. M. Mäkelä. Interactive method NIMBUS for nondifferentiable multiobjective optimization problems. In J. Clímaco, editor, Multicriteria Analysis, Proceedings of the XIth International Conference on Multiple Criteria Decision Making, pages 310–319. Springer, Berlin, Heidelberg, 1997.

JUSSI HAKANEN is a senior researcher at the Faculty of Information Technology at the University of Jyväskylä, Finland. He received MSc degree in mathematics and PhD degree in mathematical information technology, both from the University of Jyväskylä. His research is focused on multiobjective optimization and decision making with an emphasis on interactive multiobjective optimization methods, data-driven decision making, computationally expensive problems, explainable/interpretable machine learning, and visualization aspects related to many-objective problems. He has participated in several industrial projects involving different applications of multiobjective optimization, e.g. in chemical engineering. He has been a visiting researcher in Cornell University, Carnegie Mellon, University of Surrey, University of Wuppertal, University of Malaga and the VTT Technical Research Center of Finland. He has a title of Docent (similar to Adjunct Professor in the US) in Industrial Optimization at the University of Jyväskylä, Finland.

SANJIN RADOŠ is a researcher at VRVis Research Center in Vienna, and a member of Interactive Visualization group since 2015. He is one of the contributors to the ComVis visualization tool. He received the BSc degree in Media Informatics and Visual Computing from TU Wien, Vienna, Austria. He is currently pursuing the master's degree in Visual Computing with TU Wien. His research interests include multidimensional data visualization, interaction techniques, education dashboards and machine learning.

GIOVANNI MISITANO is a doctoral student working in the Multiobjective Optimization Group at the University of Jyväskylä. He received his MSc degree from the university of Jyväskylä in 2020. Giovanni's research focuses on the interpretability aspects of interactive multiobjective optimization. This includes, but is not limited to, researching new ways to make interactive multiobjective optimization methods less opaque to the decision maker and analyst alike. In addition, he is interested in studying how to apply interpretable and explainable artificial intelligence to multiobjective optimization in general. He is also one of the main contributors to the DESDEO framework.

BHUPINDER S. SAINI is a doctoral student in the Multiobjective Optimization Group at the University of Jyväskylä. He recieved his M.Tech degree from the Indian Institute of Technology Kharagpur in 2018. His research interests include multiobjective optimization, data visualization, data-driven optimization, and development of evolutionary algorithms. He has worked on many open source implementations of the methods from the aforementioned topics with a focus on modularity and interpretability. He is also one of the primary contributors to the DESDEO framework.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3161465, IEEE Access

KAISA MIETTINEN received her Ph.D. degree in mathematical information technology from the University of Jyväskylä (JYU), Finland. She is currently a Professor of industrial optimization with JYU. She heads the Research Group on Multiobjective Optimization and is the Director of the thematic research area Decision Analytics utilizing Causal Models and Multiobjective Optimization (DEMO, jyu.fi/demo) at JYU. With her group, she develops an open source software framework for interactive multiobjective optimization methods (desdeo.it.jyu.fi). She has authored about 190 refereed journals, proceedings, and collection papers; edited 17 proceedings, collections, and special issues; and written a monograph on nonlinear multiobjective optimization. Her research interests include theory, methods, applications, and software of nonlinear multiobjective optimization. She is a member of the Finnish Academy of Science and Letters, Section of Science, and the Steering Committee of Evolutionary Multi-Criterion Optimization. She has been the President of the International Society on Multiple Criteria Decision Making (MCDM). She has received the Georg Cantor Award of the International Society on MCDM for developing innovative ideas. She belongs to the editorial board of seven international journals.

KREŠIMIR MATKOVIĆ is a senior researcher, area coordinator of the complex systems area, and head of the interactive visualization group at VRVis Research Center in Vienna. Since 2000, he has been leading basic and applied research projects at VRVis with numerous national and international partners. He is interested in extending visual analysis technology to challenging heterogeneous data, in particular to a combination of multi-variate data and more complex data types. He also focuses his research on developing a structured model for visual analysis that supports a synergetic combination of user interaction and computational analysis. He teaches at TU Wien where he received his doctoral degree and habilitation (in 1998 and 2016) and at the University of Zagreb, where he received his graduate degree in 1994.