

Perttu Valtonen

**Developing a User Interface for an Interactive Evolutionary
Multiobjective Optimization Method in DESDEO**

Master's Thesis in Information Technology

June 11, 2024

University of Jyväskylä

Faculty of Information Technology

Author: Perttu Valtonen

Contact information: `perttu.s.valtonen@student.jyu.fi`

Supervisors: Kaisa Miettinen, Juuso Pajasmaa, Bhupinder Singh Saini, and Giomara Lár-
raga Maldonado

Title: Developing a User Interface for an Interactive Evolutionary Multiobjective Optimiza-
tion Method in DESDEO

Työn nimi: DESDEO-käyttöliittymän kehittäminen interaktiiviselle evoluutiopohjaiselle
monitavoiteoptimointimenetelmälle

Project: Master's Thesis

Study line: Information Technology

Page count: 62+0

Abstract: Multiobjective optimization focuses on simultaneously optimizing multiple con-
flicting objectives, presenting a complex challenge where decision makers must balance with
inherent trade-offs. Interactivity plays a crucial role in effectively addressing these complex-
ities, enabling decision makers to dynamically influence the optimization process. This study
focuses on developing a user interface specifically for the interactive extension of the Ref-
erence Vector-guided Evolutionary Algorithm method within the DESDEO software frame-
work. By developing the UI, this study addresses the existing gap in tailored user interface
components for interactive evolutionary multiobjective optimization methods within DES-
DEO and details the challenges encountered during the development process. By detailing
these challenges and the implemented solutions, this study lays the groundwork for future
expansions that will accommodate for a wider array of interactive evolutionary multiobjec-
tive optimization methods in DESDEO. The ultimate goal is to create a user interface that
can be effectively used by real decision makers, regardless of their technical expertise with
using different UIs, thus enhancing the accessibility and applicability of DESDEO.

Keywords: Interactive methods, evolutionary methods, multiobjective optimization, user
interface, DESDEO

Suomenkielinen tiivistelmä: Monitavoiteoptimointi keskittyy useiden keskenään ristiriitaisten tavoitteiden samanaikaiseen optimointiin. Tämä luo monimutkaisen haasteen, jossa päätöksentekijöiden on tasapainoteltava erinäisten kompromissien välillä. Interaktiivisuudella on keskeinen rooli näiden kompromissien välillä tasapainottelemisessa, sillä se mahdollistaa päätöksentekijöiden dynaamisen vaikuttamisen optimointiprosessiin. Tämä tutkimus käsittelee DESDEO-ohjelmistokehyksen interaktiivisille evoluutiopohjaisille monitavoiteoptimointimenetelmille räätälöityjen käyttöliittymäkomponenttien puutetta. Kehittämällä käyttöliittymä DESDEO:n interaktiiviselle referenssivektoriohjatulle evoluutioalgoritmimenetelmälle pystytään täydentämään tämä puute ja ratkaisemaan kehitysprosessin aikana kohdattuja haasteita. Haasteiden ja toteutettujen ratkaisujen yksityiskohtainen kuvaus luo perustan tuleville käyttöliittymälaajennuksille, jotka voivat tukea laajempaa valikoimaa interaktiivisia evoluutiopohjaisia monitavoiteoptimointimenetelmiä DESDEO:ssa. Tavoitteena on luoda käyttöliittymä, jota todelliset päätöksentekijät pystyvät käyttämään tehokkaasti riippumatta heidän teknisestä osaamisestaan eri käyttöliittymien kanssa, parantaen näin DESDEO:n saavutettavuutta ja soveltuvuutta.

Avainsanat: Interaktiiviset menetelmät, evoluutiopohjaiset menetelmät, monitavoiteoptimointi, käyttöliittymä, DESDEO

List of Figures

Figure 1. An example of Pareto optimal solution z^*	7
Figure 2. Some examples of Liger visualization tools (Duro et al. 2021).	28
Figure 3. An example of a graphing tool which applies a parallel coordinate plot, where each line represents a solution found by the optimization (Tan, Lee, Khoo, and Khor 2001).	28
Figure 4. Solving a problem and showing the resulting non-dominated front in case of two objective functions in jMetal (Durillo and Nebro 2011).	29
Figure 5. The test module of PlatEMO UI (Tian, Cheng, Zhang, and Jin 2017).	30
Figure 6. A sketch of the UI for solving a multiobjective optimization problem (Lárraga and Miettinen 2023).	31
Figure 7. The UI for the Pareto Navigator method (Tarkkanen, Miettinen, Hakanen, and Isomäki 2013).	33
Figure 8. The UI for the reference point method in DESDEO, which served as the baseline for the new UI.	35
Figure 9. A sketch from the design documents about the outlook of the new UI.	36
Figure 10. The main content of the UI is divided to left and right side sections.	39
Figure 11. A closer view of the components on the upper left section of the UI.	40
Figure 12. A closer view of the button components on the lower right section of the UI.	41
Figure 13. A closer view of the components on the upper right section of the UI.	43
Figure 14. The modal window where the DM can adjust their UI settings.	43
Figure 15. A closer view of the button components on the lower left section of the UI.	44
Figure 16. The layout of the final solution page.	44
Figure 17. An illustration of the DM setting preferred ranges.	46
Figure 18. An illustration of the DM selecting preferred solutions.	47

Contents

1	INTRODUCTION	1
2	BACKGROUND	4
	2.1 Optimization Concepts	4
	2.2 Multiobjective Optimization	5
	2.3 Decision Maker and Preference Information	8
	2.4 Evolutionary Methods	9
	2.5 DESDEO	12
3	LITERATURE REVIEW	14
	3.1 Review Process	14
	3.2 Inclusion and Exclusion Criteria	16
	3.3 Citation Mining	17
	3.4 Search Results	18
	3.5 Review Categories	19
	3.5.1 Handling Preference Information	20
	3.5.2 Visualization and Design	23
	3.5.3 Example UIs and Components	27
4	DEVELOPING THE UI FOR IRVEA	34
	4.1 Tools and Technologies	34
	4.2 Development Process	37
	4.3 Description of the UI	39
5	CASE STUDY AND DEMONSTRATION	45
	5.1 Example Problem Definition	45
	5.2 Solving the Example Problem	46
6	DISCUSSION	48
7	CONCLUSIONS	50
	ACKNOWLEDGEMENTS	52
	BIBLIOGRAPHY	53

1 Introduction

In today's fast-paced world, decision makers (DMs) in various domains face increasingly complex problems that demand effective support tools for decision-making. The complexity is characterized by multiple interdependent factors and uncertain outcomes. These problems can also be multi-dimensional, meaning there are multiple objectives that compete or conflict with one another. Optimization is a fundamental concept in various fields, including engineering, finance, healthcare, environmental management, and more. In these domains, making informed decisions often entails navigating a vast solution space to find the best compromise among competing objectives. At its core, single-objective optimization involves the process of finding the best solution among a set of feasible alternatives to achieve a specific goal. The goal is typically to maximize or minimize a certain quantity, referred to as the objective function, while satisfying a set of constraints.

Unlike single-objective optimization methods, multiobjective optimization aims to find a solution or a set of solutions that represent different trade-offs among multiple conflicting objective functions (Miettinen 1999). If the objective functions are conflicting with one another, it means that there may not be just a single optimal solution for the problem. Instead, there may be a set of Pareto optimal solutions. They are solutions where no objective function value can be improved further without worsening some other. In multiobjective optimization, the primary aim is not just to find a single solution or a set of solutions, but to provide a comprehensive understanding of the trade-offs between competing objectives for the DM.

Multiobjective optimization methods can be categorized into four distinct types: *a priori*, *a posteriori*, interactive, and methods without DM involvement (Miettinen 1999). The difference in these types comes from the way they handle the preference information that the DM provides or does not provide in the case of no DM involvement methods. *A priori* methods require that the DM provides sufficient preference information before the optimization process begins. This allows the method to tailor the search toward solutions that align with these predefined preferences. *A posteriori* methods aim to provide a range of solutions without requiring the DM's preferences upfront. This allows the DM to apply their preferences directly to the range of solutions provided. Interactive methods actively involve the DM throughout

the optimization process. By incorporating the DM's preferences, feedback, or guidance iteratively, these methods adapt the search dynamically. This real-time involvement allows the DM to directly influence the progression of the optimization, adjusting their preferences as more information becomes available and as they gain a deeper understanding of the trade-offs involved. Lastly, the methods without DM involvement operate autonomously, without any input or guidance from a DM. These methods rely on some neutral criterion to select the solution.

In the realm of multiobjective optimization, evolutionary methods have emerged as one potential way for tackling complex decision-making challenges (Coello, Veldhuizen, and Lamont 2007). These methods draw inspiration from the principles of natural selection to navigate the solution spaces of multiobjective problems. These methods are heuristic in nature, meaning they employ practical approaches that prioritize speed and feasibility over exactness. Therefore, they cannot definitively guarantee finding Pareto optimal solutions. Instead, they provide sets of near Pareto optimal solutions, which help inform the DM about possible trade-offs and alternatives (Deb 2001). While the first evolutionary methods were of *a posteriori* type, they then extended to *a priori* types and later to interactive types by incorporating preference information interactively and iteratively. Evolutionary methods are useful in multiobjective optimization for their ability to handle complex problems with multiple conflicting objectives. They are adept at exploring a wide range of possible solutions, which helps the DM to understand potential trade-offs.

The importance of interactivity arises from the inherent complexity of real-world decision problems. The DM often faces trade-offs, uncertainty, or changes in their priorities. Interactive methods allow them to incorporate their personal domain knowledge and make real-time decisions based on their intuition and expertise. By involving the DM, interactive evolutionary methods, as described by (Deb 2001), increase the likelihood of discovering solutions that align with the DM's individual preferences and insights, compared to non-interactive methods.

DESDEO (Misitano, Saini, Afsar, Shavazipour, and Miettinen 2021) is an open-source software framework aiming to bring interactive multiobjective optimization methods closer to researchers and practitioners worldwide by providing them with implementations of various

interactive methods. While user interface (UI) components for several types of interactive multiobjective optimization methods have already been developed and implemented within DESDEO, it currently lacks UI components that cater specifically to interactive evolutionary multiobjective optimization methods. Given this gap, there is a need for a study on how implementing such UI components would differ from the already developed DESDEO UIs for other methods.

This study aims to bridge this gap by developing a UI and implementing it into the existing DESDEO UI framework, specifically for the interactive extension of the Reference Vector-guided Evolutionary Algorithm (iRVEA) (Hakanen, Chugh, Sindhya, Jin, and Miettinen 2016). This method was selected because it already has functional implementations in the backend of DESDEO, which refers to the part of the software framework that handles data processing, method execution, and other non-visual functions that operate behind the scenes. By focusing on the iRVEA method, which shares common features with other interactive evolutionary multiobjective optimization methods, the new UI for DESDEO has the potential to be extended to other similar methods in the future. This thereby enhances the usability of DESDEO but also provides critical insights into the challenges and specifications that need further attention for the ongoing development of the DESDEO UI framework.

The structure of the upcoming study is as follows: Chapter 2 introduces the fundamental concepts of optimization necessary for following this study, with a specific focus on interactive evolutionary multiobjective optimization. Chapter 3 presents a comprehensive literature review, detailing the inclusion criteria for relevant studies and categorizing the findings to inform the development of the UI. Chapter 4 describes the development process of the UI for the iRVEA method for DESDEO, including the tools, technologies, and specific design choices made along with illustrations. Chapter 5 showcases a case study and demonstration of the newly developed UI, illustrating its application in solving a multiobjective optimization problem. The thesis concludes with a discussion in Chapter 6 that evaluates the research findings and implications, followed by final conclusions in Chapter 7 that summarize the study and suggest the needs for future work.

2 Background

Optimization is a multidisciplinary approach that draws on mathematical, computational, and analytical methods to improve decision outcomes (Luenberger 1969). In the context of decision-making, optimization involves finding the best solution among a set of possible alternatives. This chapter goes over the fundamental concepts and terms in the field of optimization and multiobjective optimization, in particular, necessary for the context of this study.

2.1 Optimization Concepts

In the realm of optimization, several fundamental concepts form the basis for modeling and solving problems. The key concepts as described in (Luenberger 1969) and (Pierre 1986) are *objective functions*, *decision variables*, and *constraints*.

An objective function is typically a mathematical expression that quantifies the goal or goals one aims to optimize in optimization problems. In data-driven optimization, the objective function can also be a simulation model or a real-world experiment. An objective function represents a quantity, such as profit, utility, efficiency, or any other measurable property relevant to the problem, that needs to be maximized or minimized. Typically, with optimization methods, unless otherwise specified, objective functions are minimized. Mathematically, an objective function is typically represented as $f : \mathbb{R}^n \rightarrow \mathbb{R}$, where elements in \mathbb{R}^n represents vectors of the n decision variables of the problem.

Decision variables are the parameters within an optimization problem and are adjusted to optimize specific objective function values. For example, in a manufacturing context, decision variables might include production quantities, the allocation of labour hours, or the product mix. These variables, which are controlled within the manufacturing process, significantly impact the optimization of objective functions like cost, efficiency, or throughput.

In single-objective optimization, where there is only one objective function, the goal is to determine the values of the decision variables x that result in the optimal value of the func-

tion $f(x)$, either maximizing or minimizing it, depending on the problem's requirements. A specific example would be optimizing the volume of an aluminum can, where the objective function f for volume depends directly on the decision variables x_1 and x_2 , representing the can's height and radius, respectively. By adjusting these decision variables, x_1 and x_2 , the volume $f(x)$ changes, demonstrating how the values of decision variables directly influence the outcome of the objective functions. The specific values assigned to these decision variables in any given case represent a potential solution to the optimization problem, which the optimization method aims to refine towards the best possible outcome.

Constraints are conditions or limitations that the solution to the optimization problem must satisfy. They represent the boundaries or restrictions on the decision variables and reflect real-world limitations, such as resource availability, budget constraints, or physical constraints. Thus, by eliminating options that do not meet the specified requirements, constraints narrow down the *feasible solution space*, which is the set of all possible solutions that satisfy these constraints. While constraints reduce the feasible solution space, they can make finding a valid solution more challenging by limiting the number of valid solutions available. Additionally, constraints can sometimes divide the feasible region into isolated areas, further complicating the optimization process.

2.2 Multiobjective Optimization

In multiobjective optimization, there are two or more conflicting objective functions that need to be optimized simultaneously. These objective functions are often in competition with each other, as improving one may result in the degradation of another. This competition inherently involves the concept of *trade-offs* (Miettinen 1999), necessitating decisions on which objective functions to prioritize.

A multiobjective optimization problem can be mathematically formulated as follows:

$$\begin{aligned} &\text{minimize } F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\ &\text{subject to } \mathbf{x} \in S, \end{aligned} \tag{2.1}$$

where $F(\mathbf{x})$ is a vector of $k(\geq 2)$ objective functions provided with the input \mathbf{x} , each function f_i representing a distinct goal of the optimization. The set S , also known as the feasible solution set, represents all points \mathbf{x} that satisfy the problem's constraints (Miettinen 1999).

Pareto optimality is a concept in multiobjective optimization, named after the Italian economist Vilfredo Pareto. In problems with multiple conflicting objective functions, a solution is considered Pareto optimal if it is not possible to improve in one objective function without degrading performance in at least one other objective function (Miettinen 1999).

It is also important to distinguish between the *objective space* and *solution space*. The objective space is where the values of the objective functions exist. With k objective functions, the objective space is \mathbb{R}^k . This is separate from the solution space \mathbb{R}^n , which refers to the range of all possible decision variable configurations that have not been narrowed down by constraints.

A *Pareto front* represents the set of all Pareto optimal solutions in the objective space. When represented graphically, the Pareto front typically appears as a curve or surface in the objective space. The Pareto front can be a curve even in problems with more than two objective functions, in which case it is known as a degenerate Pareto front. Each point on the Pareto front corresponds to a solution or solutions in the solution space, representing a set of decision variables. For an example of a Pareto optimal solution visualized, see Figure 1. In the figure, the example of a Pareto optimal solution z^* in the 2-dimensional objective space is represented on the bold Pareto front curve. The axes f_1 and f_2 represent the values of their respective objective functions, and the area denoted by Z represents the image of the feasible solution set in the objective space.

Directly related to the concept of Pareto optimality is the idea of *non-dominated solutions*. A solution is non-dominated if no other solution in the given solution set is better in at least one objective function and not worse in any other objective function when all objective functions are considered simultaneously (Miettinen 1999). Pareto optimal solutions differ in that they represent the broader concept of optimality in multiobjective optimization, encompassing all non-dominated solutions. In other words, solution s_1 dominates solution s_2 if s_1 is no worse than s_2 in all objective functions and better in at least one objective function.

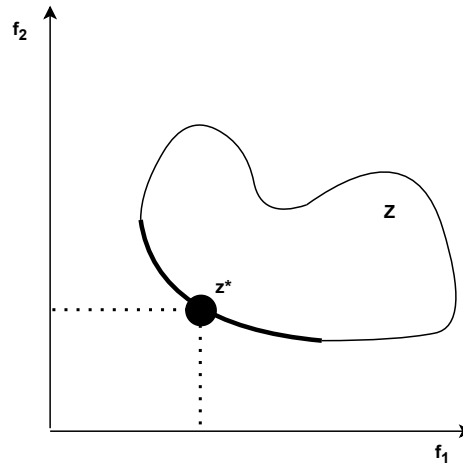


Figure 1. An example of Pareto optimal solution z^* .

Thus, non-dominated solutions represent the best trade-offs among the objective functions under consideration and are all equally valid from a purely mathematical perspective until further preferences are applied by the DM. These solutions constitute the Pareto front: where each solution is optimal in the sense that no other solutions in the set offer a better outcome without compromising another objective function. This set of solutions provides the DM with a variety of options to choose from, each reflecting a different balance of the objective functions based on their specific needs and priorities.

Building upon the understanding of the Pareto front, the concepts of *ideal* and *nadir vectors* are important for evaluating multiobjective optimization solutions (Miettinen 1999). The ideal vector consists of the best possible values for each objective function across all solutions, representing an optimal solution that is not practically achievable but serves as a target for optimization efforts. Conversely, the nadir vector represents the worst values for each objective function among the Pareto optimal solutions, essentially outlining the least desirable outcomes within this set. Together, these vectors define the boundaries of the objective space and help the DM to visualize the full range of achievable outcomes, enabling them to better understand the extents of trade-offs and to set realistic expectations for the optimization results. This is particularly useful in graphical representations of the Pareto front, providing a clearer perspective on the relative performance of solutions.

2.3 Decision Maker and Preference Information

A *decision maker* is the individual human or group responsible for making decisions in the optimization process and uses their preferences to navigate among solutions found via the optimization. The preferences and decisions of the DM are crucial because Pareto optimal solutions are mathematically incomparable due to trade-offs (Miettinen 1999). The DM plays a crucial role in defining the problem, setting goals, and ultimately choosing the final solution based on their preferences.

The involvement of the DM is critical in multiobjective optimization, as it incorporates human judgment, domain expertise, and subjective preferences into the decision-making process, ensuring that the final solution aligns with the DM's goals. Information provided by the DM is referred to simply as *preference information*. Typically, the DM is a domain expert of the problem being solved and is ultimately responsible for selecting the final solution.

As mentioned in the introduction, multiobjective optimization methods can be categorized into four main types based on when the DM provides preference information in the solution process: *a priori*, *a posteriori*, interactive, and without DM involvement (Miettinen 1999). The key differences among these methods lie in their approaches to incorporating and managing preference information.

In *a priori* methods, the DM's preferences are considered upfront in the optimization process. This requires the DM to have a good understanding of the problem in order to provide realistic preference information. After the DM has defined their preferences before the optimization method begins its exploration, the method aims to find solutions that align with these preferences from the outset.

In *a posteriori* methods, the DM's preferences are considered after the optimization process has found a representative set of possible solutions. These methods first present a comprehensive view of the potential solutions, allowing the DM to retrospectively apply their preferences and select the solution that best aligns with their priorities. *A posteriori* methods are effective in situations where initial preferences are not well-defined. However, a key shortcoming of these methods is that they can be overwhelming if the set of solutions is very large, potentially making it difficult for the DM to make a clear choice.

In interactive methods, the DM actively participates in the optimization process which consists of iterations. In these methods, a dynamic feedback loop allows the DM to receive information at the end of each iteration and to iteratively input their preferences. This real-time interaction enables the DM to steer the exploration of the solutions in a manner that reflects their evolving preferences. Through this iterative process, the DM gains insights into the trade-offs and interdependencies inherent in the problem, which may lead them to modify their initial preferences as they gain a better understanding of the complexities and potential outcomes of the optimization.

Lastly, methods without a DM are methods where explicit preferences by the DM are unavailable. These methods utilize a neutral criterion to autonomously select the final solution without external guidance. Such an approach is advantageous when preferences are difficult to articulate, ensuring objectivity since the method is relying solely on quantifiable data to make decisions. Because of this, these methods can be time-consuming as they often require extensive computation to evaluate all possible solutions thoroughly and determine the optimal one according to the neutral criterion, without any DM preferences to guide and potentially shorten the search process.

2.4 Evolutionary Methods

Evolutionary methods draw inspiration from the principles of natural selection and genetics to simulate an evolutionary process in the optimization domain (Branke, Deb, Miettinen, and Słowiński 2008). By emulating these principles, evolutionary methods in multiobjective optimization explore the solution space, providing a diverse set of trade-off solutions to the DM.

The evolutionary process involves selection, crossover, and mutation operations. In each generation, a new population is created by selecting the better-performing solutions from the current population and applying genetic operators to produce offspring. This process continues until certain termination criteria are met, such as reaching a maximum number of generations or achieving a satisfactory level of solution quality.

These methods are particularly effective in handling complex and high-dimensional optimization problems where “traditional” non-evolutionary optimization techniques may struggle. Because of this, evolutionary methods are used in multiobjective optimization to efficiently search for solutions in the solution space (Deb 2001).

The key strengths of these methods include their ability to explore a wide solution space and adaptively refine solutions based on problem-specific characteristics, and in the case of interactive methods, input from the DM (Branke, Deb, Miettinen, and Słowiński 2008). However, a notable drawback is the stochastic nature of these methods (Coello, Veldhuizen, and Lamont 2007), which can lead to variability in performance and will require multiple runs to achieve consistent results.

Evolutionary methods in optimization are heuristic in nature (Deb 2001). This means that they do not guarantee a Pareto optimal solution but often provide effective approximations. They operate by manipulating a set of solution candidates, known as a population. Throughout generations, which are iterative in nature, the populations evolve toward better solutions. These “better solutions” are determined using a fitness function that evaluates candidates based on their suitability. These fitness functions do not necessarily guarantee finding Pareto optimal solutions but rather approximated solutions (Coello, Veldhuizen, and Lamont 2007).

An example of an evolutionary method is the *Reference Vector-guided Evolutionary Algorithm* (RVEA) (Cheng, Jin, Olhofer, and Sendhoff 2016). The RVEA method uses a set of reference vectors to identify and choose the most promising solutions, thereby efficiently searching for solutions within the solution space.

Evolutionary methods in optimization have seen a progression first from *a posteriori* to *a priori* methods, and then to interactive methods (Deb 2001). This evolution reflects a shift from generating solutions first and involving the DM afterward (*a posteriori*) to incorporating DM’s preferences upfront (*a priori*), and finally to involve the DM during the optimization process (interactive).

The increasing emphasis on interactivity is particularly crucial when dealing with complex decision problems, where there may be intricacies and nuances that are not immediately apparent (Branke, Deb, Miettinen, and Słowiński 2008). Interactivity allows for a deeper

understanding, ensuring that the optimization solutions found are comprehensive in addressing all relevant aspects of the problem. Furthermore, interactivity enhances the flexibility, adaptability, and relevance of these solutions, making them more aligned with the nuanced preferences and evolving insights of the DM (Branke, Deb, Miettinen, and Słowiński 2008).

The iRVEA (Hakanen, Chugh, Sindhya, Jin, and Miettinen 2016) is an interactive extension of the RVEA method. This method begins by presenting a set of initial solutions to the DM, which provide a foundational understanding of the solution space. The DM can then iteratively provide from one of four distinct types of preference information to steer the optimization towards the most favorable outcomes. The four distinct types of preference information, as detailed in (Hakanen, Chugh, Sindhya, Jin, and Miettinen 2016), are:

- Reference point: the DM can specify a desired value for each objective function, indicating their aspiration levels for outcomes. The optimization method uses these values to focus the search around the specified preferences, aiming to find solutions that resemble and respect these aspiration levels as closely as possible.
- Preferred ranges: the DM can define the upper and lower bound values, which define a range of values, for each objective function. By doing so, the optimization seeks solutions that fall within these ranges, thereby filtering out those that do not meet the specified preferences.
- Preferred solutions: selecting this option allows the DM to select their preferred solutions from the ones displayed in the UI. This preference information is used to steer subsequent iterations toward similar solutions, enhancing the likelihood of satisfying the DM's specific needs.
- Non-preferred solutions: conversely, from the set of solutions that are shown in the UI, the DM can select solutions that they wish to steer away from. The optimization process then adjusts to exclude these characteristics in the next iteration, helping to refine the search towards more favorable outcomes.

Each type of preference information directly influences the subsequent iterations, helping to refine and direct the search towards solutions that best meet the DM's needs. Given that the method could potentially find hundreds or thousands of solutions per iteration, it also integrates an input from the DM regarding the desired number of solutions to be received.

This input is essential for ensuring the data remains manageable and effectively analyzable. This nuanced method of incorporating preference information requires a user interface that can not only display these multifaceted data but also enable the DM to interactively and iteratively provide and refine their preferences.

2.5 DESDEO

DESDEO (Misitano, Saini, Afsar, Shavazipour, and Miettinen 2021) is a software framework designed for multiobjective optimization and released under a permissive open-source license. It supports both non-interactive and interactive methods, such as RVEA and iRVEA, and features developed UIs for some of the methods including the reference point method (Wierzbicki 1999).

The framework features an architecture that facilitates the utilization and further development of interactive methods. Its modular components are reusable, which allows for the implementation of new methods using old components. DESDEO's aims include making interactive methods more easily available and applicable, thus contributing to the advancement of multiobjective optimization research and practice.

When discussing about software, the term 'backend' refers to the part of the software that houses the core computational logic and in the case of DESDEO, performs the optimization calculations. DESDEO's backend consists of packages, specifically designed for multi-criteria decision-making (MCDM) and evolutionary multiobjective optimization (EMO). The MCDM package includes non-evolutionary methods, such as the reference point method (Wierzbicki 1999), while the EMO package contains evolutionary methods, including iRVEA.

These packages are accessed from the UIs through the 'desdeo-webapi', which serves as DESDEO's application programming interface (API). This API facilitates communication between the frontend UIs and the backend computational logic, allowing the DM's input and preferences to be processed and the optimization results to be returned and displayed within the UI.

The typical workflow in DESDEO begins when the DM selects the problem they wish to optimize and the method with which to solve it. They are then directed to the dedicated UI for the chosen method. However, DESDEO currently lacks dedicated UI components specifically designed for interactive evolutionary methods. While it provides a foundation for multiobjective optimization, the absence of an intuitive and user-friendly UI for interactive evolutionary methods hinders its broader applicability.

This study aims to bridge this existing gap in DESDEO's UI components by developing a dedicated UI specifically tailored for the iRVEA method. As mentioned previously, this method was selected because it already has functional implementations in the backend of DESDEO. This UI will be designed to enhance user experience, providing the DM with a more intuitive and engaging platform for guiding the optimization process.

Effective UIs help simplify the complexity of multiobjective optimization problems by visually representing data, which can make it easier for the DM to understand the trade-offs involved and assess the feasibility of different solutions (Xin, Chen, Chen, Ishibuchi, Hirota, and Liu 2018). The DM can also adjust preferences in real-time and immediately see the impact of these changes on the optimization results, facilitating a more informed decision-making process. Lastly, a well-designed UI can make complex optimization tools more accessible to DMs who may not have a deep technical background in optimization theory or evolutionary methods. This access can expand the use of these methods across a wider range of disciplines and applications.

3 Literature Review

As has been established, the integration of effective UIs in interactive evolutionary multiobjective optimization methods is crucial for enhancing decision-making processes. This is also a particular necessity in the context of DESDEO, which currently lacks modular, reusable and user-friendly UI components specifically designed for such interactive methods.

Following this understanding, the goal of this literature review is two-fold: first, to establish a comprehensive understanding of the current state of UI design within the domain of interactive evolutionary multiobjective optimization. This includes exploring what prior examples of UIs have been developed in similar contexts, with a focus on identifying features and designs that promote effective decision-making and user engagement.

Second, this review aims to uncover the possible needs and gaps in UI design, which should be taken into account as they pertain to DESDEO and its further development with UIs involving interactive evolutionary multiobjective optimization methods. Ultimately this review intends to aid in developing a UI for DESDEO, that enhances user experience and provides DMs with a more intuitive and engaging platform for guiding the optimization process.

3.1 Review Process

For this literature review, source material was systematically compiled using the academic databases *Scopus* and *Web of Science*. These databases were chosen because they are reputable and contain scientific, peer-reviewed literature. Furthermore, a crucial factor in their choice was the consistency and reliability of their search features. Unlike other databases that might use algorithms or approximations leading to fluctuating search results based on user behavior or other variables, these databases provide stable and repeatable search outcomes. This consistency is vital for this literature review, as it guarantees that searches can be replicated accurately, ensuring transparency and reproducibility in the research process (Aghaei Chadegani, Salehi, Yunus, Farhadi, Fooladi, Farhadi, and Ale Ebrahim 2013).

An advanced search feature was utilized, allowing for the application of various keyword combinations. The search strategy was iteratively adjusted to balance between comprehensiveness and relevance. It was structured around finding specific queries formed from keyword combinations that will yield literature regarding the subject matter. Due to the subject matter being quite specific, it was necessary to form two separate queries and thus perform two rounds of literature searches in both databases.

The query for the first round was focused on identifying literature that discusses UIs in some manner within the context of interactive evolutionary multiobjective optimization. Simultaneously it was necessary for the query to leave out literature which is in general more broadly focused on optimization or literature focusing simply on some specific optimization method.

The queries used in the first round of Scopus and Web of Science database searches, respectively, were:

- *TITLE-ABS-KEY (user AND interface AND interactive AND evolutionary AND (multi OR many) AND objective AND (optimization OR optimisation))*
- *ALL=(user AND interface AND interactive AND evolutionary AND (multi OR many) AND objective AND (optimization OR optimisation))*

The query for the second round was for finding literature regarding a broader and more general focus on UI design, particularly emphasizing the incorporation of user's preference information. Thus, the query aimed to find literature that could help identify best practices, design strategies, UI components or visualizations relevant to enhancing UIs in systems like DESDEO.

The queries used in the second round of Scopus and Web of Science database searches, respectively, were:

- *TITLE-ABS-KEY (user AND interface AND design AND interactive AND preference AND information)*
- *ALL=(user AND interface AND design AND interactive AND preference AND information)*

3.2 Inclusion and Exclusion Criteria

The primary inclusion criteria for this literature review were that the papers to be considered must address UI design or user engagement in the context of interactive evolutionary multi-objective optimization or otherwise be applicable in this context in some way. This included not only theoretical discussions but also practical examples of developed UIs in this field. Studies were selected if they provided insights into UI elements, user experience, user interaction, or demonstrable implementations within this specific area. Lastly, since the literature searches were conducted in English, it is natural that all of the included literature must be likewise in English.

Conversely like with the inclusion criteria, the exclusion criteria were stringent. Any literature that did not directly address UI or user engagement, or focused solely on optimization without considering interactivity from the user's perspective, was omitted. This approach ensured a focused and relevant collection of sources that could directly inform the development of UIs for DESDEO. Literature concerning mobile applications or virtual reality systems were also excluded since the design choices made within those could not be realized in DESDEO. Similarly any literature focused on using artificial intelligence or other such intelligent agents as well as highlighting specific tools in designing adaptive UIs were excluded as DESDEO does not currently utilize such design factors.

In terms of source material, the review predominantly included peer-reviewed articles and conference papers, which were chosen for their scientific credibility. The databases used, Scopus and Web of Science, are scientific literature databases, which ensured the quality and reliability of the found material. Given the specialized nature of the topic, emphasis was placed on selecting sources that contributed directly to the understanding of UI design in the context of interactive evolutionary multiobjective optimization.

Considering the relatively recent emergence and ongoing development in the field of UI design for interactive evolutionary multiobjective optimization, the time frame for when the literature was released was not strictly limited as a criterion. This decision was based on the understanding that the number of the relevant literature would most likely be quite small by itself. However, if necessary, priority would be given to more recent studies to ensure

the relevance and currency of the research. This approach ensured that the literature review captured the most up-to-date and relevant developments in the field, reflecting the current state of UI design in interactive evolutionary multiobjective optimization.

3.3 Citation Mining

After the initial batch of literature that passed through the qualification process, a thorough forward and backward search also known as 'citation mining' (Florida Atlantic University Libraries 2024) was conducted to refine our collection of relevant literature. This methodological approach is designed to ensure comprehensiveness and depth in our literature review.

In the backward search, the references cited within each selected paper are examined. This approach allows us to identify seminal works and relevant studies that might have been missed in the original database searches. Given the strict nature of the subject, some pivotal studies could have been overlooked due to variations in keyword usage or publication in sources not indexed in the chosen databases. By manually reviewing the references of the initial batch of literature, these potentially significant studies would be captured.

Conversely, the forward search involves looking at studies that cite the initially selected literature. This method helps in identifying more recent works that have built upon the foundational research which was initially identified. These backward and forward searches are able to be done easily by using citation-tracking tools available in the Scopus and Web of Science databases. Each piece of literature identified through forward and backward searches are subjected to the same qualification process as our initial batch.

3.4 Search Results

The initial phase of the first round of the literature search yielded, as expected, quite a small but promising set of results across the two databases used. From Scopus, our query resulted with 34 hits, whereas Web of Science provided 9 hits. These results showed preliminary adherence to our predefined search criteria focused on UI design in interactive evolutionary multiobjective optimization.

Upon a preliminary review of these hits, the qualification process was conducted. This process involved examining each article to determine its relevance to our research objectives, particularly focusing on whether the studies addressed UI design, user engagement, or provided practical examples of UI implementation in our field of interest. As a result of this scrutiny, a total of 12 articles were deemed qualified and suitable for inclusion in our literature review.

From Scopus, the supplementary backward search through the references of the initially selected literature found 351 pieces of literature. Out of these, 5 new articles were found that adhered to the inclusion criteria. The backward search in Web of Science, involving 227 references from the initially selected literature, similarly yielded 3 new pieces of literature that passed the qualification criteria. However, all three were also identified in the backward search conducted in Scopus. In the forward searches from Scopus, 231 citing works were found, and the Web of Science literature had 65 citings. However, none of these articles contributed any new supplementary literature that met our qualification criteria.

As expected, a majority of these supplementary search results involved literature simply concerning optimization or its methods and different frameworks used in it instead of specifically concerning UIs and their implementations in this context. Many of the already selected literature was also found in these supplementary searches, which indicated the strength and accuracy of the initial search queries.

The second round of searches for the literature review was done in a similar manner to the first one, albeit now with the searches done for literature that focus on the UI design when it comes to handling preference information. From Scopus, our query resulted in 202 hits and Web of Science provided 76 hits. These results were then put through the same qualification

process as with the results from the first round and afterwards 6 suitable pieces of literature from the Scopus results and 1 from the Web of Science results were left. Also similarly with these results all of the ones found in the Web of Science were also found in the Scopus results, confirming that at least when it comes to this subject matter the Scopus database seems to offer a better range of viable results.

After performing the backward and forward searches for the search result from the second round of literature searches, we found 3 additional qualified pieces of literature from Scopus and an additional one from Web of Science. Supplementing our initial qualified search results with these we ended up with a total of 11 qualified literature works to be included in our results from the second round of literature searches.

3.5 Review Categories

After compiling and going over the selected articles in the review, there was a need for dividing them into their respective categories for improved comprehensiveness and clarity. While the results from this literature review overlap in places, they will be summarized as three distinct categories, *Handling Preference Information*, *Visualization and Design*, and *Example UIs and Components*, according to their contents:

- **Handling Preference Information:** This category includes studies focusing on integrating decision-maker's preferences into multiobjective optimization processes. These works demonstrate the importance of incorporating human input in guiding the optimization towards preferred solutions.
- **Visualization and Design:** Literature under this category emphasizes the role of advanced UIs in multiobjective optimization. They highlight how effective visualization aids are in navigating complex solution sets and in enhancing decision-making processes or what design choices can be made with the UI to enhance the user experience.
- **Example UIs and Components:** This includes studies that present specific UIs designed for multiobjective optimization. These studies are crucial as they provide practical examples of how UIs can be structured to support user interaction in complex optimization tasks.

Dividing the appropriate articles by their focus on handling preference information was crucial because it is a core aspect of interactive optimization. It is fundamental since the essence of interactive multiobjective optimization is to find solutions that best match the DM's goals, which can vary widely and is guided by the preference information they provide. By focusing on preference information handling, this category highlights approaches that make optimization tools more adaptive and directly impacting the relevance and accuracy of the found solutions.

The role of UIs, particularly in terms of visualization and design, is central to making complex data and solution sets understandable and interactable for users. This category is justified by the need to present multi-dimensional data in a way that is intuitive to the user, enhancing the decision-making process by providing an understanding of trade-offs between competing objectives. It encompasses studies that explore how UI design and visual representation techniques can improve user experience and interaction.

Lastly, focusing on specific UI implementations offers concrete examples of how UI design and preference handling can be applied in practice. This category is essential for illustrating the practical application of theoretical principles and methodologies discussed in the other two categories as well as providing additional reference points on ways of implementing the required components for the DESDEO UI. By showcasing actual UIs developed for multi-objective optimization tasks, these papers provide insights into the challenges and solutions in UI development.

3.5.1 Handling Preference Information

The selected articles in this section collectively emphasize the critical role of DM's preference information in the optimization process. These papers highlight approaches to integrating preference information, utilizing both interactive methods and advanced visualization tools to steer the search towards solutions that align with the DM's goals. This summary aims to find useful and viable methods in making the optimization process more accessible and aligned with DM's needs, thereby enhancing the decision-making and user experience in interactive optimization.

The articles (Sekizaki, Nishizaki, and Hayashida 2019) and (Hettenhausen, Lewis, and Mostaghim 2010) emphasize the importance of effectively incorporating the DM's preferences into multiobjective evolutionary methods to guide the search process towards more relevant and satisfactory solutions for complex optimization problems. The article (Hettenhausen, Lewis, and Mostaghim 2010) also introduces a heatmap-based UI, to efficiently incorporate the DM's preferences and domain-specific knowledge into the optimization process, demonstrating improved solution quality and focus on areas of interest compared to non-interactive methods.

When utilizing visualization tools to enhance multiobjective optimization processes, the article (Sato, Tomita, and Miyakawa 2015) showcases the effective use of a parallel coordinate plot within the UI, to effectively display the representative solutions approximating the Pareto front and facilitating the specification of the DM's preferences. Additionally, it focuses on handling preference information by allowing the DM to specify preferred regions in the objective space rather than single points, guiding the optimization process towards areas of interest.

The paper (Deb and Chaudhuri 2006) discusses the development and application of an interactive multiobjective evolutionary optimization procedure designed to address the challenge of selecting a single, optimal solution from a set of Pareto optimal solutions. The software guides the DM through the process of exploring the Pareto front, focusing on preferred regions and ultimately selecting a single solution for implementation. Simultaneously the software is aiming to enhance the DM's confidence in their final choice by offering various checks and balances. One such example is the ability to investigate the robustness of solutions under variable perturbations, ensuring that chosen solutions remain viable under slight changes in the problem's parameters.

In the paper (Gong, Liu, Wei, Jiao, and Zhang 2011), an interactive multiobjective evolutionary method is introduced, which facilitates direct interaction between the DM and the optimization process. The method is decomposition-based, which means that it first decomposes a multiobjective optimization problem into several single-objective sub-problems. It then efficiently incorporates preference information by presenting a set of current best solutions from each sub-problem to the DM and allows them to choose their most preferred one.

The method then identifies the weight vector associated with the selected solution as the center of a new preferred region, thus guiding the search towards the DM's favored regions.

IK-EMOViz (Ghosh, Deb, Averill, and Goodman 2023) is a software tool designed to incorporate the knowledge and intuition of experienced DMs into the optimization process. It enables interactive engagement by extracting and visualizing key information, such as emergent patterns, trends, and relationships among variables, from ongoing optimization processes. The DM can then provide feedback directly through a UI, where they can interactively select, modify, or prioritize specific rules or insights based on their domain expertise. The framework supports both synchronous and asynchronous interactions: where the synchronous interaction enables real-time feedback and immediate adjustments to the optimization process, while asynchronous interaction allows the DM to interact with the system at their convenience, utilizing background computational resources to further the optimization. This dual approach facilitates a more adaptive and efficient pathway to discovering solutions.

The paper (Jaimes, Montaña, and Coello 2011) investigates the application of an interactive optimization method on problems with 2, 3 and 6 objectives. The method, which incorporates DM's preferences and focuses searches on areas of interest, demonstrated improved performance in most cases over methods without preferences. The study suggests the potential for enhanced optimization strategies by carefully balancing the region of interest size and exploring alternative evolutionary operators.

The NAUTILUS Navigator (Ruiz, Ruiz, Miettinen, Delgado-Antequera, and Ojalehto 2019) and O-NAUTILUS (Saini, Emmerich, Mazumdar, Afsar, Shavazipour, and Miettinen 2022) methods present approaches to multiobjective optimization, emphasizing interactive decision-making processes that efficiently handle the preference information without necessitating trade-offs. They focus on understanding and learning about the problem's intricacies through UIs that display the impact of the DM's preferences in real-time.

Some ways to enhance the DESDEO's UIs in handling preference information are the integration of interactive and visual feedback mechanisms that allow the DM to directly influence the optimization process and facilitate an intuitive understanding of trade-offs and the impact of preferences in real-time. This is highlighted in the aforementioned papers, where

incorporating the DM's preferences effectively with interactive methods and advanced visualization tools has improved the optimization process. Additionally, there is a need for implementing UI elements to effectively guide the search towards areas of high interest and relevance, as well as providing comprehensive information on the solutions to enhance the DM's confidence in their final choice.

3.5.2 Visualization and Design

The articles reviewed in this section underline the landscape of UI design within the realm of multiobjective optimization. The research highlights the potential of adaptive and customizable UIs to address challenges such as information overload, user fatigue and the need for transparency in interactive systems. This summary showcases approaches to UI design and interaction in optimization, pointing towards methods that facilitate more engaging and efficient decision-making process but also enhance the accessibility and effectiveness of optimization tools.

The paper (Hettenhausen, Lewis, and Kipouros 2014) discusses advancements in user interaction and UIs for interactive multiobjective optimization, focusing on a particle swarm optimization method. It highlights the importance of efficient UIs in interactive optimization, especially for engineering applications, using an aeronautics engineering problem as a case study. The research introduces improvements to both the UI and the interaction process between the user and the optimization method.

The paper (Hakanen, Radoš, Misitano, Saini, Miettinen, and Matković 2022) presents an “interactivized” approach that integrates visual analytics with interactive multiobjective optimization to aid DMs in navigating through conflicting objectives by visually steering the iterative solution process. The interactivized method improves upon the capabilities of existing optimization software by allowing for more dynamic interaction through advanced visualization tools. This facilitates a more comprehensive and efficient decision-making process, particularly for complex, computationally demanding problems. The study underscores the potential of visual analytics to enhance decision support in interactive multiobjective op-

timization, offering a more intuitive, control-oriented and informative solution exploration experience for DMs.

The study (Kollat and Reed 2007) introduces a tool designed to facilitate the visual exploration and decision-making process for large multiobjective solution sets, which aims to enhance users' understanding of trade-offs among competing objectives. The framework is specifically engineered to enable users to navigate through Pareto optimal solution sets for problems with three or more objectives. It does this by employing dynamic visualization methods, which allow users to instantly see how changes in decision variables affect outcomes spatially across a given domain and provide an immediate graphical representation of how chosen solutions perform against each objective.

The paper (Tan, Lee, Khoo, Khor, and Kannan 2000) introduces a UI toolbox designed to streamline multiobjective optimization tasks. It features a user-friendly UI requiring minimal programming knowledge and advanced functions like dynamic population size and interactive fine-tuning capability, which allows users to adjust optimization parameters and constraints in real-time, based on evolving design requirements or insights gained from initial results. The toolbox facilitates easy setup, monitoring, and analysis of optimization processes, aimed at enhancing decision support in design through active human-computer interaction.

Exploring the integration of visual interaction techniques from visual analytics into interactive multiobjective optimization methods, the paper (Hakanen, Miettinen, and Matković 2020) aims to enhance the decision-making process by involving the DM iteratively. The study underscores the importance of visual analytics in creating an effective UI between the DM and the optimization method.

Addressing digital reading challenges, the paper (Jin and Fu 2023) proposes an optimization method based on scene theory to handle the challenges of information overload, inconvenient interaction and user fatigue in digital reading. The study focuses on identifying key design factors and refining design strategies with UI color as the main focus to better meet users' needs for reading from a UI.

The paper (Hearst and Tory 2019) explores user preferences for integrating visualizations, such as charts and graphs, within UIs. The research highlights the importance of personal preference in the design of data-oriented UIs, suggesting that while charts with contextual information may be widely preferred, offering personalization options could enhance user experience.

Examining adaptive UIs, the study (Peissner and Edlin-White 2013) explores the design and effectiveness of UI patterns, aimed at enhancing transparency and controllability of adaptations to improve user accessibility in interactive systems. The potential drawbacks of adaptive UIs are disorientation and loss of control, although the study proposes specific design strategies to mitigate these issues. The findings indicate that the implemented patterns successfully increase the system's transparency and controllability, allowing users to better manage and optimize the system's adaptive behavior.

The study (De Angeli, Sutcliffe, and Hartmann 2006) evaluated two websites with identical content but different UI styles: traditional menu-based and interactive metaphors. Despite the metaphor-based design being perceived as more aesthetically expressive, it scored lower in both objective and perceived usability compared to the traditional menu-based design. This contradiction suggests that while expressive aesthetics enhance engagement and are preferred for less serious contexts or younger audiences, they may not guarantee better usability. The findings highlight the importance of context in designing and evaluating UIs, suggesting that usability and aesthetics have context-dependent effects on user perceptions and preferences.

Introducing a theoretical framework to assess the attractiveness of websites, the study (Hartmann, Sutcliffe, and De Angeli 2007) discusses how framing and halo effects influence users' judgments on aesthetics, showing a complex interplay between perceived usability, aesthetics, content and overall preference. It suggests that users' impressions can significantly influence satisfaction and acceptability, even overriding poor usability experiences. The study further highlights the importance of knowing the audience's preferences and expectations, indicating that design priorities should align with user profiles and application domains. It concludes that while aesthetics are crucial, perception can be influenced by users' backgrounds and tasks. Good design can, however, mitigate some usability issues.

The paper (Reinecke, Yeh, Miratrix, Mardiko, Zhao, Liu, and Gajos 2013) explores how users form instant judgments on a website's attractiveness based on its visual complexity and colorfulness, significantly affecting their views on its usability and trustworthiness. The research underscores the impact of first impressions on website evaluation, providing quantitative tools for predicting appeal and guiding design improvements to enhance user experience.

The study (Findlater and McGrenere 2004) investigates the efficiency and user preference between static, adaptable (user-controlled customization), and adaptive (system-controlled customization) menu UIs through a controlled experiment. Results showed that static menus were significantly faster than adaptive ones, and adaptable menus were faster than adaptive under certain conditions, with a notable preference for the adaptable approach among users. The majority of participants preferred adaptable menus, indicating a strong support for UIs that allow user input in customization.

Insights from the reviewed studies emphasize the importance of integrating UI designs and interaction processes tailored specifically for complex optimization tasks. Other findings suggest utilizing emotional design principles and optimized UI colors, which enhance engagement and reduce cognitive fatigue of the user. Moreover, personalization and visual feedback mechanisms, along with considering the first-look aspect and adaptability, are highlighted as possibilities for catering to individual preferences, improving data interpretability, and enhancing user experience. Finally, integrating visual analytics techniques is advised to facilitate a more intuitive and informative decision-making process, allowing the DM to iteratively explore and refine solutions, thereby making the system more accessible and effective for the DM.

3.5.3 Example UIs and Components

The collection of articles in this section showcase the development of user-friendly optimization platforms and interactive methods that cater to researchers and non-specialists alike. Key aspects to be summarized include ease of use, decision-making enhancements and the integration of preference information by the user, with a particular interest in how these features improve the overall user experience and facilitate decision-making in interactive multiobjective optimization.

Liger (Duro et al. 2021) is a user-friendly optimization software designed for non-specialists in industry, enabling easy construction of optimization workflows through a visual programming language and a novel library for composing multiobjective evolutionary methods. Liger supports a wide variety of problem types, including those with uncertainties and mixed variable types, and extends functionality through a plugin system for customization. See Figure 2 for examples of Liger visualization tools, which facilitate interactive decision-making by providing the DM with the capability to actively participate in the optimization process. Liger allows for the real-time adjustment of preferences during the optimization run, enabling the DM to guide the search towards their most preferred solutions based on the articulated preferences. This interactive capability is supported by advanced visualization tools that help the DM study and select from solutions found by evolutionary methods.

The paper (Tan, Lee, Khoo, and Khor 2001) introduces a UI-based toolbox for evolutionary multiobjective optimization methods, which utilizes the Pareto optimality concept to find a diverse set of non-dominated solutions. Showcasing an example of a graphing tool included in the toolbox, see Figure 3. The toolbox is equipped with various features for enhanced decision-making, including dynamic population size adjustment, soft/hard goal settings, multiple goal specifications, and convergence representation. It is designed for ease of use, requiring minimal evolutionary computing knowledge, and supports extensive decision-making and optimization analysis tools.

The jMetal framework (Durillo and Nebro 2011) is a Java-based tool designed to assist in the research and application of metaheuristics to multiobjective optimization problems. It aims to provide an easy-to-use, flexible, and extensible platform for method implementation,

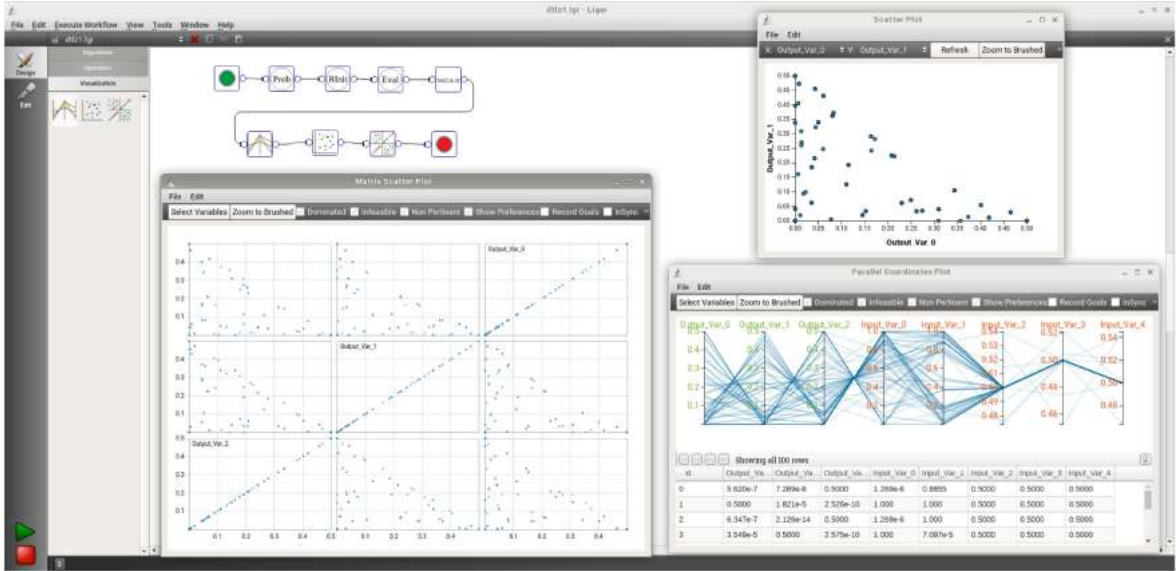


Figure 2. Some examples of Liger visualization tools (Duro et al. 2021).

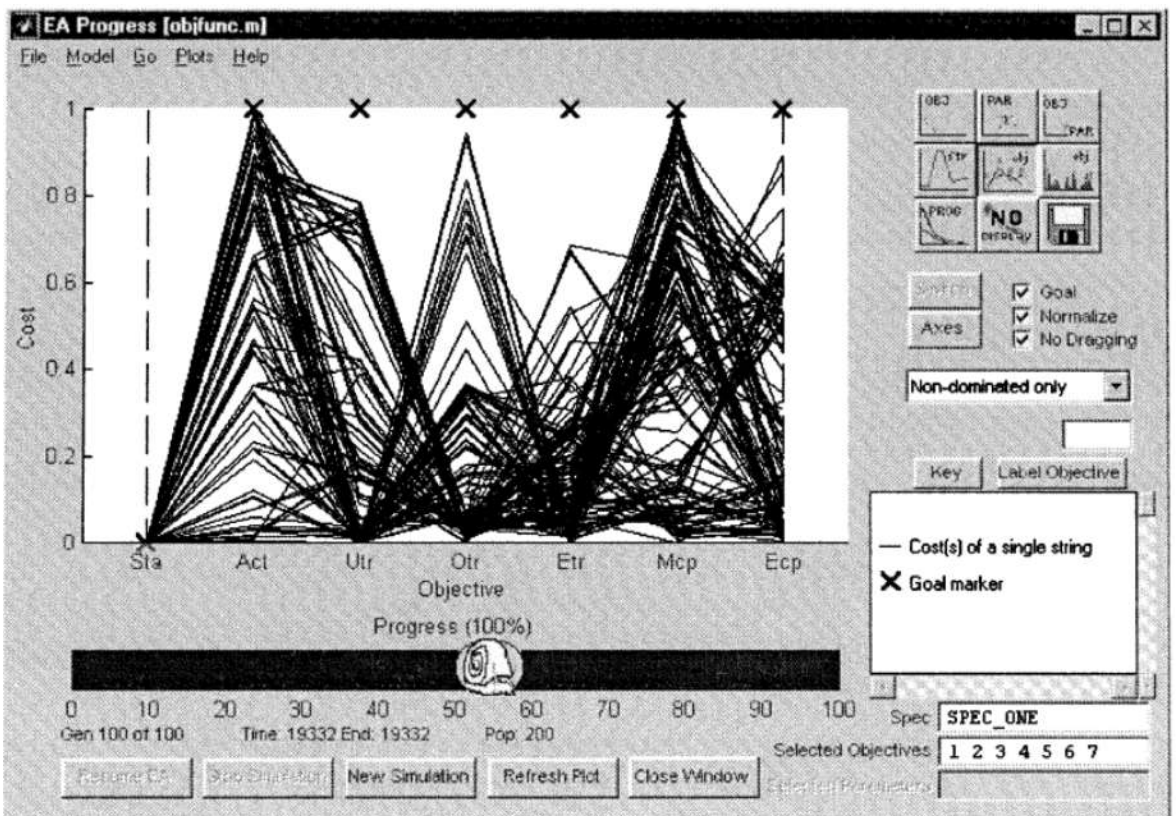


Figure 3. An example of a graphing tool which applies a parallel coordinate plot, where each line represents a solution found by the optimization (Tan, Lee, Khoo, and Khor 2001).

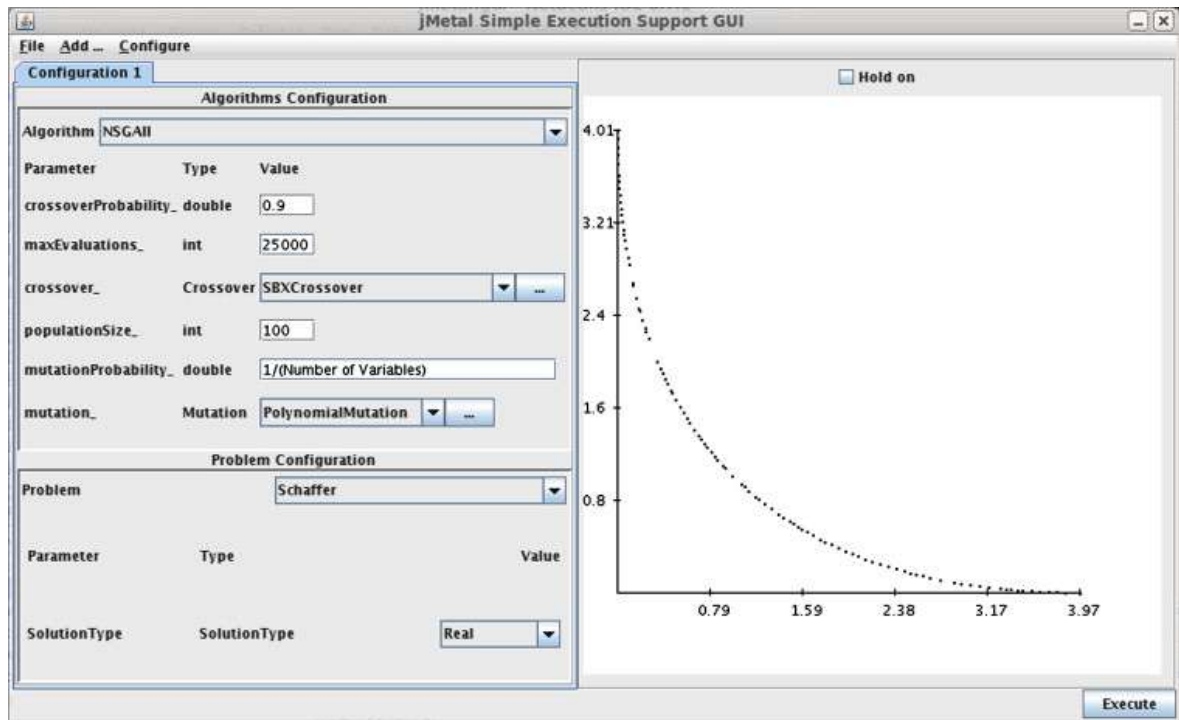


Figure 4. Solving a problem and showing the resulting non-dominated front in case of two objective functions in jMetal (Durillo and Nebro 2011).

problem definition, and performance assessment. jMetal supports a broad range of applications and includes features like graphical tools and decision-making support. See Figure 4 for an illustration of how jMetal operates in action. Notably, jMetal is equipped to handle multiobjective optimization problems involving more than two conflicting objective functions, for which it uses advanced algorithms that are capable of managing these conflictions.

PlatEMO (Tian, Cheng, Zhang, and Jin 2017) is a MATLAB-based open-source platform designed to support the field of evolutionary multiobjective optimization. It offers over 50 evolutionary methods and more than 100 test problems, along with widely used performance indicators. Displaying the test module capabilities of PlatEMO, Figure 5 showcases its utility in helping the DM analyze and choose among different optimization strategies. While primarily aimed at research purposes for method benchmarking and real-world problem applications, PlatEMO also supports the DM by providing user-friendly UIs for easy method comparison and result extraction to Excel or LaTeX files. This enables both the academic

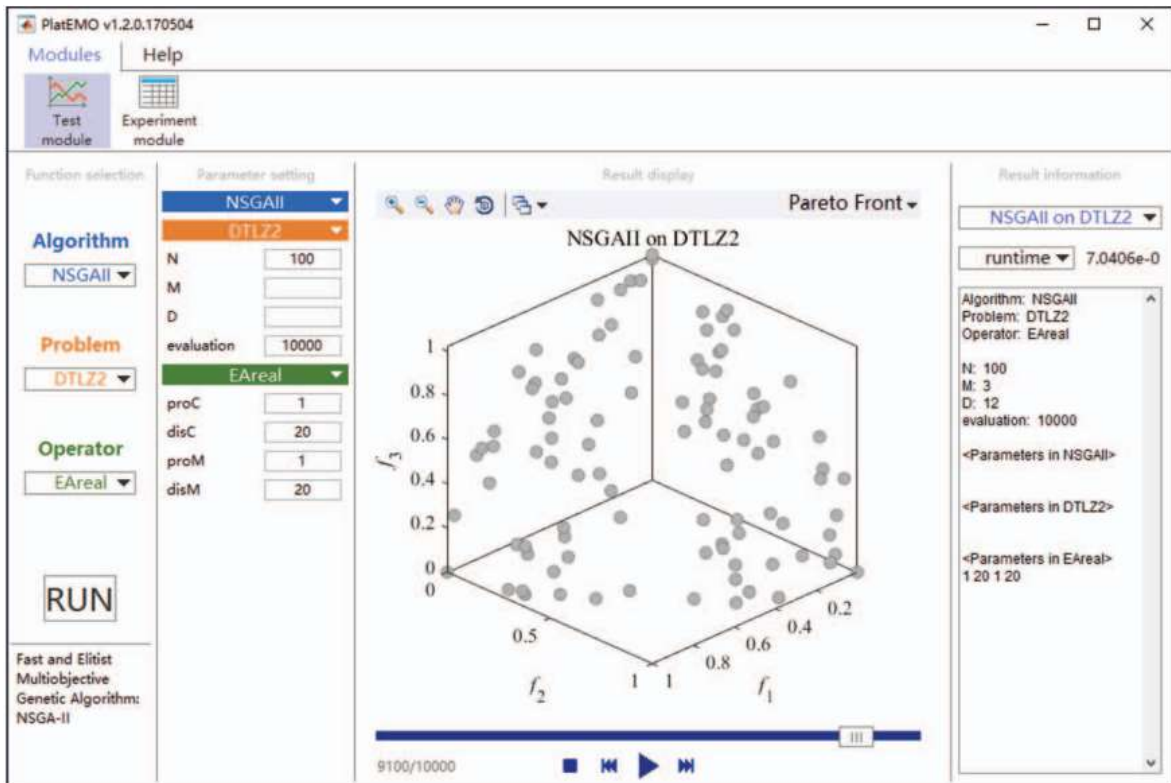


Figure 5. The test module of PlatEMO UI (Tian, Cheng, Zhang, and Jin 2017).

and practical application of the platform, facilitating the assessment and visualization of trade-offs among various solutions.

The paper (Lárraga and Miettinen 2023) introduces a framework aimed at improving interactive evolutionary multiobjective optimization methods by incorporating the DM's preferences, thereby focusing on more relevant solutions in optimization problems with multiple conflicting objectives. For a sketch of a UI aimed at solving multiobjective optimization problems, see Figure 6. This framework outlines a strategy for identifying common algorithmic components and the DM's interaction needs. It suggests combining these components in novel ways to create or refine methods and includes a practical implementation example featuring a UI, database, and algorithmic components.

The article (Tarkkanen, Miettinen, Hakanen, and Isomäki 2013) showcases the development and empirical validation of a UI for Pareto Navigator, an interactive multiobjective optimization method (Eskelinen, Miettinen, Klamroth, and Hakanen 2010). The prototype

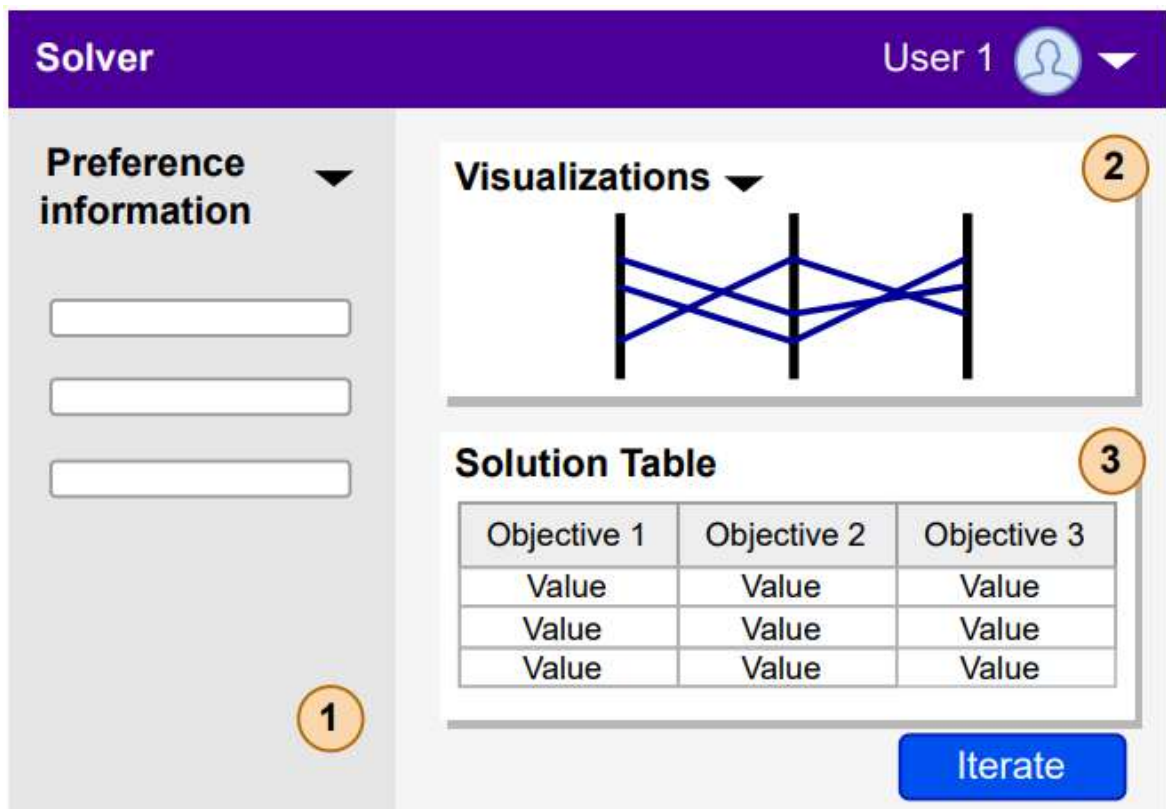


Figure 6. A sketch of the UI for solving a multiobjective optimization problem (Lárraga and Miettinen 2023).

implementation of the UI for Pareto Navigator is shown in Figure 7. The research emphasizes the importance of design science and usability testing. Key findings include the identification of suitable interaction and visualization techniques for various aspects of interactive multiobjective optimization, such as selecting a starting point, preference elicitation, and navigating through solutions. The usability tests confirmed the effectiveness of these techniques in making the decision-making process more intuitive and efficient.

Some ways to enhance the DESDEO's UIs include effectively utilizing user-friendly visualization and interactive decision-making tools. This involves focusing on constructing easy and intuitive workflows, expanding with UIs to support the implementation of a further variety of methods, and integrating dynamic preference information. These enhancements are specifically designed to accommodate DMs who may not be technically adept when interacting with different UIs, ensuring the platform is accessible and useful for DMs with multiple levels of technical proficiency. Another way is to utilize design science and empirical validation approaches, as demonstrated in the Pareto Navigator research, to identify effective interaction and visualization techniques that facilitate an intuitive understanding of trade-offs and preference impacts, thereby improving decision-making efficiency and real-world applicability.

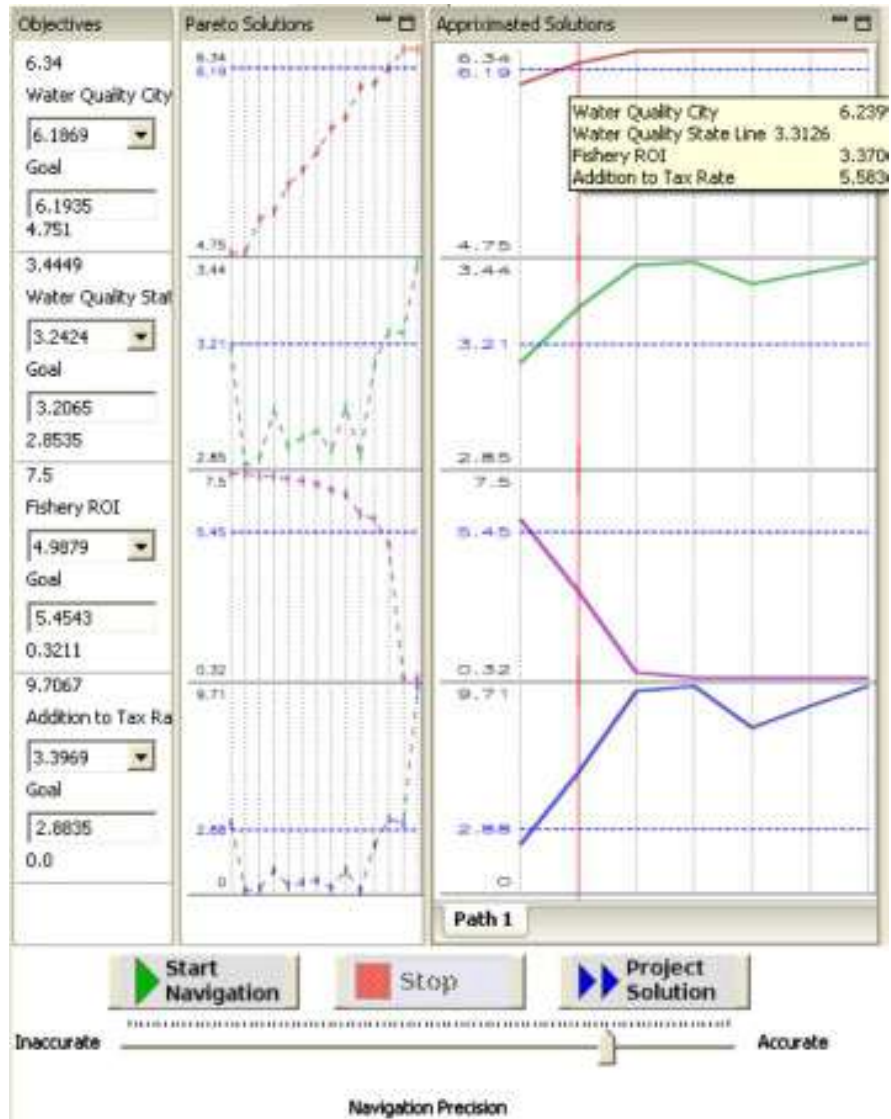


Figure 7. The UI for the Pareto Navigator method (Tarkkanen, Miettinen, Hakanen, and Isomäki 2013).

4 Developing the UI for iRVEA

This chapter goes over the process of developing a user-centric and modular UI for DESDEO for the interactive evolutionary multiobjective optimization method iRVEA, detailing the selection of development tools, design principles aimed at enhancing user experience, and technical challenges encountered. Special attention is given to user-friendly design decisions and innovative solutions drawn from the literature review. By thoroughly explaining and visually presenting the process, the chapter highlights how the UI not only facilitates but also enhances the decision-making experience in interactive evolutionary multiobjective optimization.

4.1 Tools and Technologies

The source code for DESDEO is available on GitHub and can be accessed at the DESDEO GitHub repository (<https://github.com/industrial-optimization-group/DESDEO>). For comprehensive documentation on how to utilize and extend the functionalities of DESDEO, the DESDEO documentation is available at (<https://desdeo.readthedocs.io/en/latest/>).

The development of the new UI for DESDEO began by taking the existing UI (as of October 4, 2023) for another interactive method, namely the reference point method (Wierzbicki 1999) as the base for the new UI development, see Figure 8. It was chosen as a natural starting point because it was already correctly integrated into DESDEO's navigation hierarchy and features a layout and some components for providing preference information by the DM.

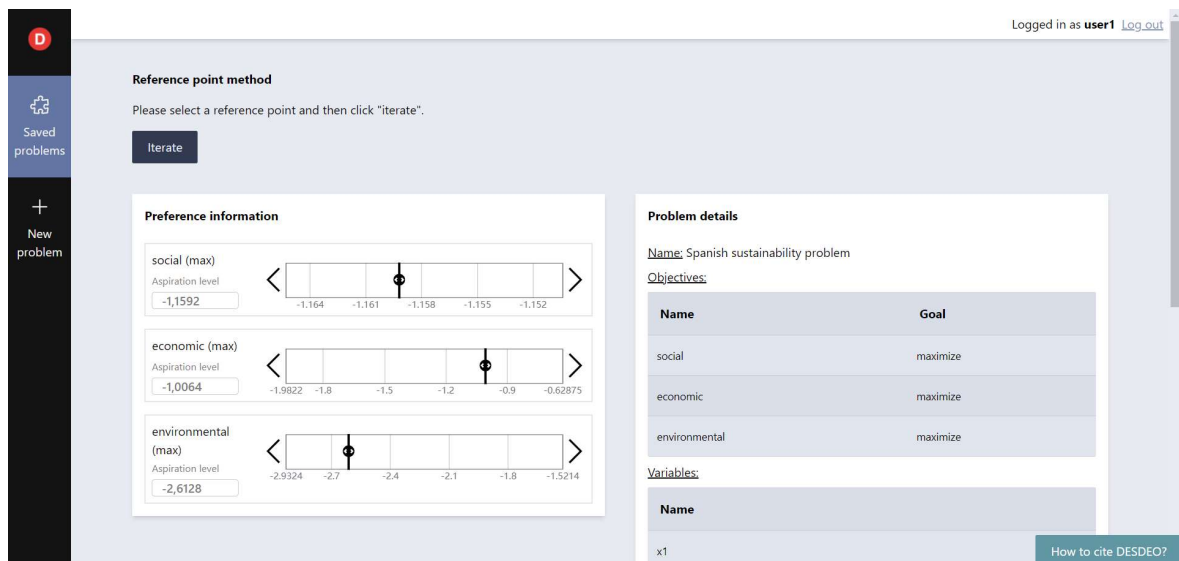


Figure 8. The UI for the reference point method in DESDEO, which served as the baseline for the new UI.

The UI was developed using technologies that were already a part of DESDEO and therefore used in the development process by default for compatibility with the existing DESDEO components. These technologies used included the following:

- TypeScript – a typed superset of JavaScript that compiles to plain JavaScript.
- Svelte – a TypeScript web framework for building UIs.
- ECharts – an interactive charting and visualization library for the browser.
- Skeleton – a lightweight CSS (Cascading Style Sheets) framework for developing fast and responsive websites.
- Git – a distributed version control system to track changes in the source code during software development.

The conducted literature review was used to justify and demonstrate the choices made within implementing the UI design. Key findings from the literature review emphasized the importance of intuitively handling DM's preference information, clear visualization of data, and interactive elements that guide and enhance the decision-making process.

The DESDEO development team provided design documents to also assist in developing the UI during the development process. These documents included detailed sketches that

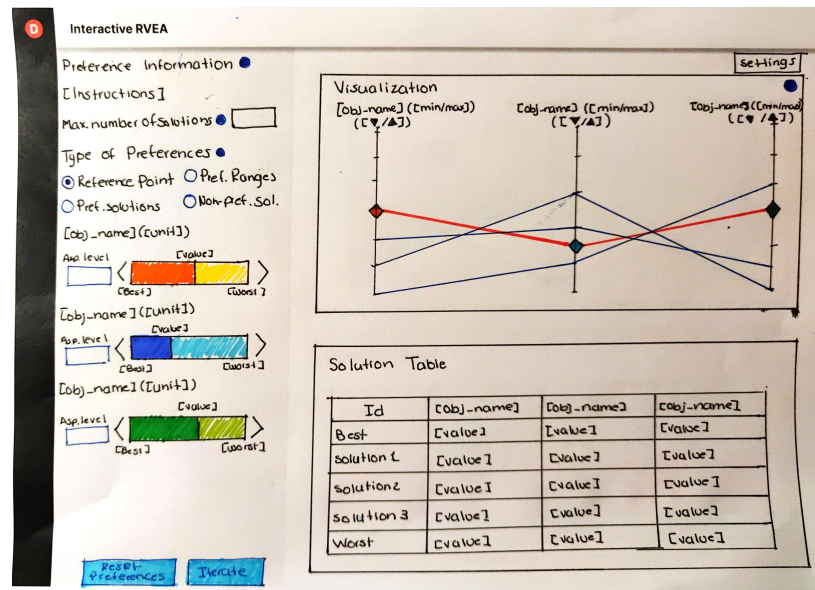


Figure 9. A sketch from the design documents about the outlook of the new UI.

outlined the general layout of the UI and essential components for the required functionalities, see Figure 9. Additionally, the documents outlined the aesthetic elements such as color schemes and text content to be displayed.

Regular meetings with members of the DESDEO development team were integral to aligning the development progression with the strategic objectives for the UI and aligned with the desired outcomes. These meetings served as a platform for continuous feedback, allowing iterative refinements to the UI based on real-time insights from the development team.

It was imperative to ensure that the developed UI conformed both visually and structurally to the rest of DESDEO. This conformity involved adhering to established design patterns and styles already employed in DESDEO to maintain a uniform user experience across the software framework.

Given that DESDEO is modular by nature, it was important that the UI components were also specifically designed to be user-friendly and reusable. This modularity allows for easy integration into different contexts within DESDEO, supporting current and future developments. Such design ensures that all UI components can be adjusted and reused as needed, facilitating flexibility and expansion as the framework evolves.

4.2 Development Process

In describing the development process, the focus here is on detailing the functionalities and components that were either developed or significantly modified, in accordance with the goals set within this study. As such, the details primarily revolve around these developments, with less emphasis placed on other aspects of the UI that were not significantly altered.

The initial step in developing the new UI was to start upon the existing UI designed for the reference point method, due to the reasons previously mentioned. This approach allowed for an iterative process where incremental adjustments and feature additions could be methodically integrated and tested in the UI simultaneously while developing the UI. According to the design documents, the UI was to be divided into two sections: the left side displays the preference information and handles its management, while the right side presents the current set of solutions found via the optimization with both visualizations and in a table format. This division is meant to facilitate further visual analysis by the DM and aid them in navigating through conflicting objective functions by visually steering the iterative solution process (Hakanen, Radoš, Misitano, Saini, Miettinen, and Matković 2022). Both sections represent the state of the current iteration, but are designed to operate such that interacting with either side will not affect the state of the other, until the DM explicitly decides so.

The first necessary addition to the UI was the inclusion of radio button components on the left side of the UI. The radio buttons needed to accommodate the selection from the four distinct types of preference information of the iRVEA method detailed earlier in the background chapter. This selection and the corresponding preference information provided by the DM, would then directly influence the optimization process by tailoring the solutions found via the optimization to better align with the DM's preferences. Along with the preference information the DM needed the ability for indicating the amount of solutions they would like to see displayed at a time.

Following this, the next critical step was to enable the DM to input the appropriate preference information after selecting the preferred type. The existing reference point functionality from the pre-existing UI for the reference point method was utilized as such for providing reference points. For accommodating the DM to provide the preference information as pre-

ferred ranges, there was a need for a similar functionality, where both upper and lower bound values would be stored simultaneously. The lower bound represents the minimum acceptable value and the upper bound the maximum acceptable value for a solution. Whether minimizing or maximizing an objective function, these bounds help define the range within which solutions are considered optimal, aligning the optimization process with the DM's specific preferences.

Implementing options for the DM to categorize solutions as preferred or non-preferred was another development step. This functionality was integrated into the right side of the UI where the visualizations and the table are located. In this way, the DM is allowed to select the preferred or the non-preferred solutions directly from either the visualizations or the table. However, since it was decided that interactions on the right side of the UI would not influence the state of the left side, the confirmation of these selections required additional interaction. Specifically, the DM must actively transfer these selections on the right side of the UI as preference information to the left side of the UI through a dedicated action of pressing a button.

When the DM identifies a satisfactory solution from the available solutions presented on the right side of the UI, they need the ability to select it as the final solution and conclude the optimization process. Once the DM has decided to conclude the optimization process, the UI needs to change accordingly to display the solution data and appropriate visualizations for the final solution. This final solution page view needs to be modular and universal so it could be used with other optimization methods in DESDEO as well. This is because many other methods will similarly need the ability to get to this final solution page upon completion, each with their own final solution through the respective optimization processes. After this point, the option to return to the prior optimization stage is no longer available, and the only option for the DM is to start a new optimization.

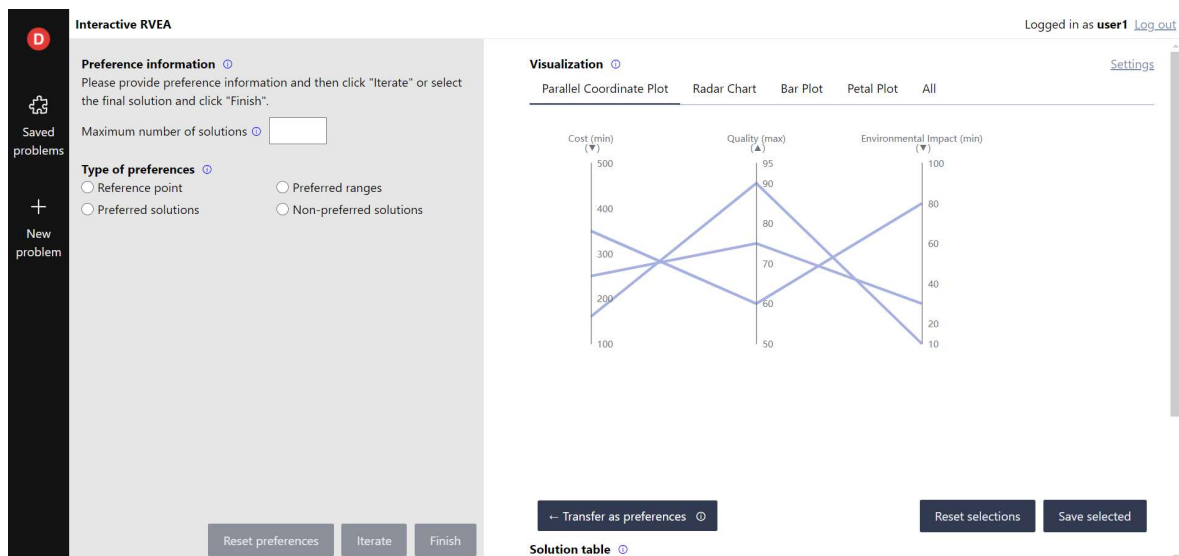


Figure 10. The main content of the UI is divided to left and right side sections.

4.3 Description of the UI

The developed UI reflects on both the provided design documents as well as the supporting findings from the conducted literature review. As mentioned, the UI layout is divided into two sections. The left side manages the preference information for the optimization process, while the right side dynamically displays the optimization results, which reflect the preferences provided as well as possible, through both visualizations and in a table format, see Figure 10.

Since the optimization method could potentially find hundreds or thousands of solutions in each iteration, to lower the cognitive burden on the DM there is a need for indicating the maximum number of solutions to be displayed. This is also important for the visualizations, as the visualizations must be able to represent the differing solutions distinctly enough for the DM to analyze them and make necessary follow-up choices. Therefore, there was a need to implement a way for the DM to indicate the maximum amount of solutions, which was done with a text box for entering the maximum amount to be shown. For an illustration, see component A in Figure 11, which provides a closer view of the upper left section from the prior Figure 10. This amount is then passed to the backend along with the previously described preference information.

Preference information ⓘ

Please provide preference information and then click "Iterate" or select the final solution and click "Finish".

Maximum number of solutions ⓘ

3

A

Type of preferences ⓘ

Reference point

Preferred ranges

Preferred solutions

Non-preferred solutions

Cost (min)

Lower bound

100



Upper bound

277,966

Quality (max)

Lower bound

69,6398



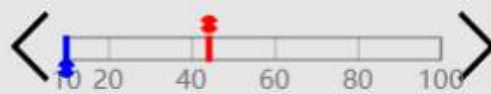
Upper bound

95

Environmental
Impact (min)

Lower bound

10



Upper bound

44,322

Figure 11. A closer view of the components on the upper left section of the UI.



Figure 12. A closer view of the button components on the lower right section of the UI.

Choosing the type of preference information by using the radio button components, as shown in component B in Figure 11, alters the appearance of the left side, allowing the DM to provide the selected preference information accordingly. The component specifically developed for setting preferred ranges, see component C in Figure 11, visualizes two adjustable points on horizontal bars, representing the upper and lower bounds of the ranges. These horizontal bars support the DM by displaying both the ideal and nadir values, providing a clear visual boundary within which the DM can operate. This design allows the DM to easily input and adjust preferred ranges of objective function values through intuitive interactions with the UI. The DM can either enter desired values manually into input boxes or adjust by dragging the value indicators along the horizontal bars.

A transfer button component needed to be implemented for enabling the DM to allow the state of information on the right side of the UI to affect the state of the left side. For an illustration of the transfer button, see component D in Figure 12, which provides a closer view of the buttons located on the lower right section of the UI illustrated in Figure 10. The main need for this function was to allow the DM to transfer the selected solutions from the visualizations and the table as preference information as either preferred or non-preferred solutions. For consistency, the transfer button should also work with the reference point and preferred ranges types of preference information. For the reference point preference information, the functionality was implemented so that selecting a single solution on the right side would set that solution's objective function values as the preference information on the left side when chosen to do so with the transfer button.

Since some of the solutions found via the optimization iterations may interest the DM for reasons other than being the final chosen solution, there was a recognized need to save these solutions for later examination. To accommodate this, the corresponding functionality was

implemented that allows the DM to save the currently selected solutions using a dedicated save button, as shown in component E of Figure 12. The selected solutions are then stored in the backend through a specific endpoint call. This functionality facilitates a more thorough analysis by preserving any results that may offer valuable insights during the decision-making process.

Implementing the transfer to work with the preferred ranges preference information was more complicated. A natural step for implementing this was to modify the already existing brushing functionality on the parallel coordinate plot visualization. This brushing functionality allows the DM to brush the desired areas of the objective functions on the plot, see component F in Figure 13, which provides a closer view of the upper right section of the UI illustrated in Figure 10. The original brushing functionality simply selected the solutions that were situated in the brushed areas of the objective functions. Modifying this brushing functionality to be used in the transfer of the preferred ranges of objective function values necessitated modifications to the parallel coordinate plot component. These changes were that the endpoint values of the brushed area are stored and can be accessed elsewhere in the UI. By using the transfer button when preferred ranges are selected as the preference information type, the stored values from the brushing can then be accessed and therefore set as the preferred ranges of objective function values. Additionally, it is not necessary for the DM to provide brushings for all objective functions simultaneously, as only the ranges for the brushed objective functions are transferred.

Some aspects of how relevant information is displayed in the UI can vary to suit the taste of the DM, such as the decimal precision used for displaying numbers in the UI. For this reason, it was important to implement a functionality that allows the DM to customize these settings. This was done by creating a settings button, which opens a modal window where the DM can change the UI settings according to their taste, as shown in component G of Figure 13 and Figure 14. While in the context of this UI development the different settings options were not a major focus and thus the developed settings are not exhaustive, they primarily demonstrate the capability for the DM to modify their settings through this modal window. Further settings options were left for future development considerations.

Visualization ⓘ

G [Settings](#)

Parallel Coordinate Plot Radar Chart Bar Plot Petal Plot All

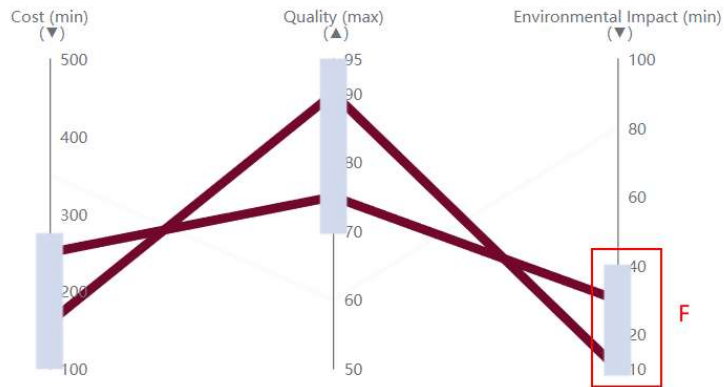


Figure 13. A closer view of the components on the upper right section of the UI.

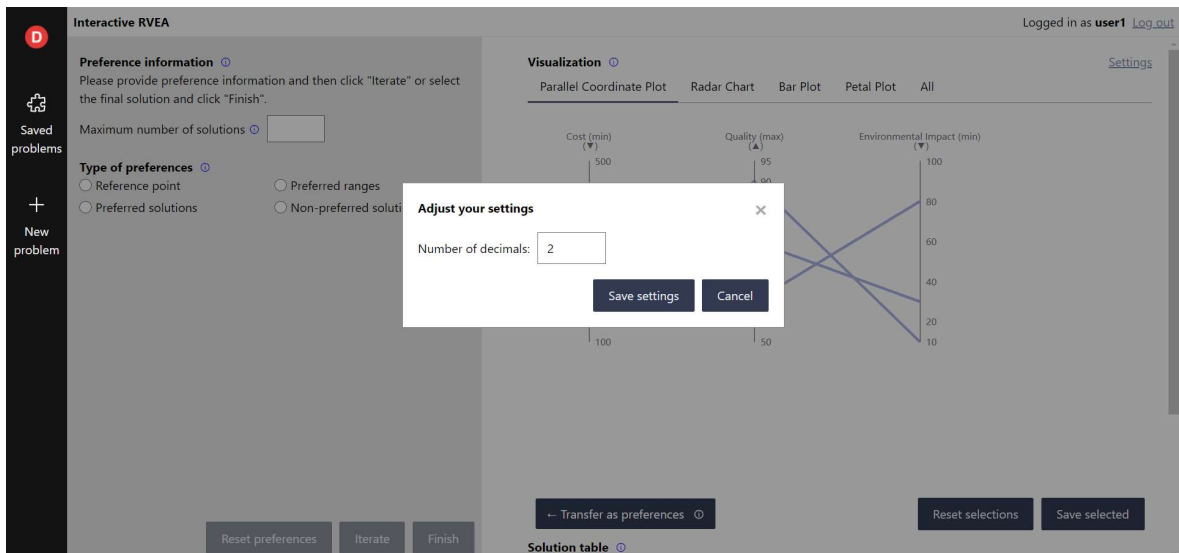


Figure 14. The modal window where the DM can adjust their UI settings.

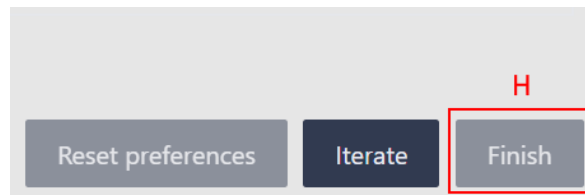


Figure 15. A closer view of the button components on the lower left section of the UI.

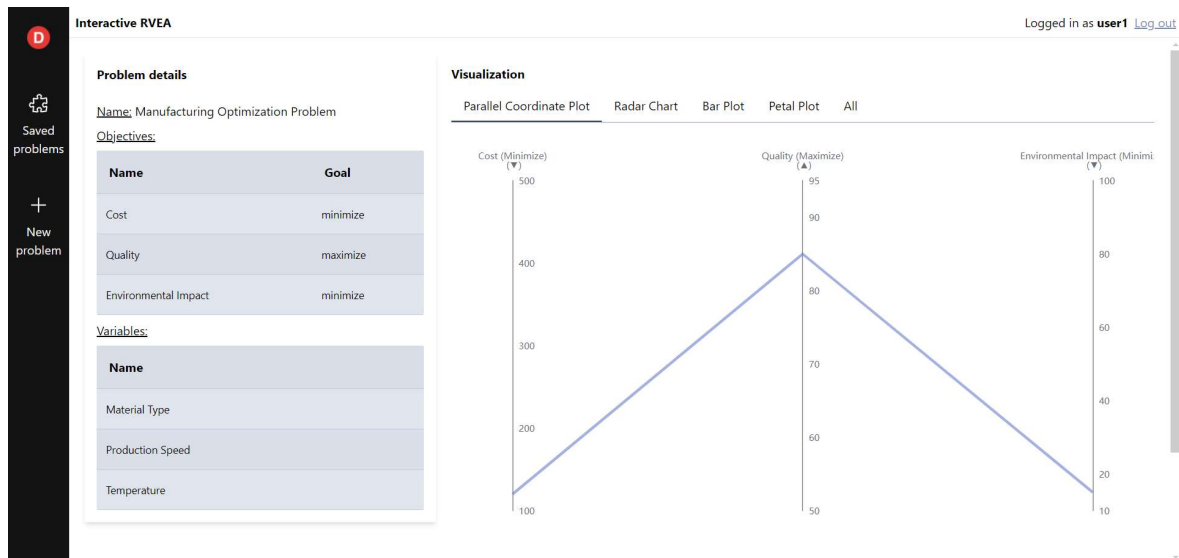


Figure 16. The layout of the final solution page.

The final solution page was implemented so that the DM is directed there after selecting their final solution from the solutions on the right-hand side, and then uses the finish button while only a single solution is selected. The finish button is depicted in component H of Figure 15, which provides a closer view of the buttons located on the lower left section of the UI illustrated in Figure 10. This action will then trigger a dedicated call to the backend to save the solution as the final solution. Simultaneously, the UI will then take the DM to a separate final solution page view. The final solution page appears as a summary view of the solution, with visualizations of the solution, see Figure 16. Throughout the UI, the DM is also supported in following the proper workflow by disabling certain selections when they are not applicable, such as disabling the finish button when more than one solution is selected.

5 Case Study and Demonstration

This chapter presents a detailed case study and demonstration to illustrate the practical application and effectiveness of the newly developed UI for DESDEO. It begins with an overview of an example optimization problem and then provides a detailed walkthrough of how the UI is utilized to solve this optimization problem, emphasizing how the DM will interact with its various components to input preferences and make decisions.

5.1 Example Problem Definition

The data presented here, including all solutions, and decision-making contexts, is entirely created for these demonstration purposes. It is designed to provide a realistic but simplified simulation of how the UI would perform with actual data in real-world contexts. The reason for using dummy data in the demonstration is due to the fact that parallel to this demonstration the DESDEO's backend is undergoing overhaul to a new version. This meant that utilizing the old version of the backend for the demonstration would have been unnecessary work endeavor. This dummy data facilitates a clear demonstration of how the DM would typically interact with the UI to solve a problem and arrive at the final chosen solution.

For this demonstration, the following dummy problem definition is used: the problem is set in a manufacturing context, focusing on optimizing the production of a product with specific considerations for cost, quality, and environmental impact. The objective functions involve minimizing cost and environmental impact and maximizing production quality, which are often relevant in manufacturing contexts.

The goal is to demonstrate how the UI can help the DM in balancing between these conflicting objective functions by effectively managing their preferences and exploring various solution contexts. This balance is crucial in manufacturing, where efficient resource management often directly impacts both cost and sustainability. The use of realistic dummy data simulates a context where the DM is concerned with variables such as material type, production speed, and temperature settings to see their effects on the overall production process.

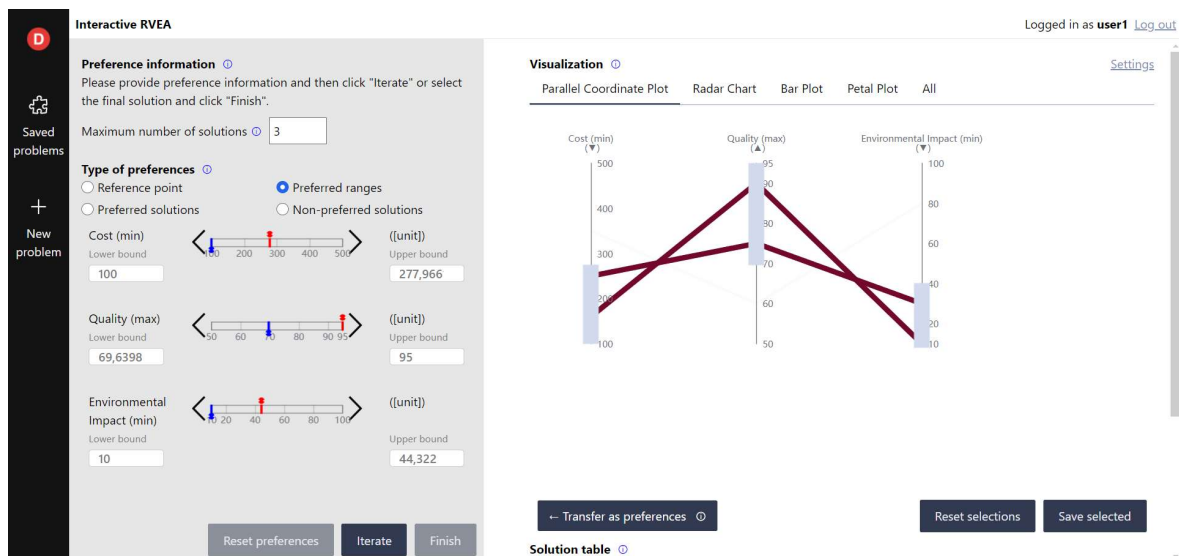


Figure 17. An illustration of the DM setting preferred ranges.

5.2 Solving the Example Problem

Once the DM has chosen the problem they wish to optimize and the method to do so, in this case the manufacturing optimization problem and the iRVEA method, they are shown the newly developed UI. In order to support the DM in decision-making, the method provides some initial solutions at the beginning of the process. These initial solutions are then already displayed in the UI before the DM has made any decisions. The DM can select the type of preference information they would like to convey through the UI to steer the optimization process in their preferred direction.

In this example, the DM wishes to set preferred range values for the objective functions. They begin by selecting 'preferred ranges' from the radio buttons in the preference information section of the UI. To specify these ranges, instead of directly inputting values or manually adjusting the upper and lower bounds with a cursor, the DM uses the parallel coordinate plot visualization. With the visualization the DM can intuitively brush the preferred ranges directly on the plot for each objective function. After defining these ranges, the DM can press the transfer button to apply the endpoints of the brushed areas as the upper and lower bound values for the objective functions, see Figure 17.

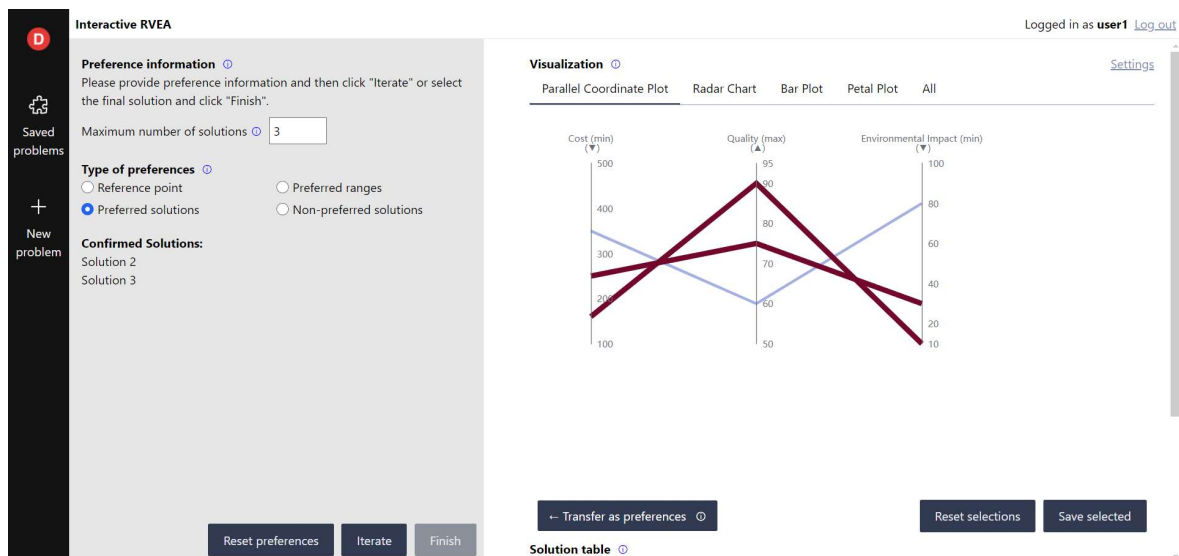


Figure 18. An illustration of the DM selecting preferred solutions.

After providing the preference information, the DM specifies the maximum number of solutions to be shown and then uses the iterate button to search for solutions following their preferences. After iterating, the UI displays the indicated number of solutions found for the DM. The DM selects from the new set of displayed solutions some promising ones and saves them for later inspection. None of them are quite satisfactory as choices for the final solution though, so while still having the promising solutions selected, they change the preference information type to preferred solutions and then use the transfer button to confirm these solutions as their preference information, see Figure 18. Once again, the DM iterates the solutions with the preferred solutions now guiding the optimization process toward their desired outcome.

Now the iteration has found a solution that the DM is confident about. It features low cost and environmental impact, while simultaneously offering high quality. The DM selects this solution from the UI and decides to conclude the optimization run, choosing it as the final solution. Upon clicking the finish button, the DM is taken to a dedicated page that summarizes their optimization run, see Figure 16. On the left of the UI, they see details of their chosen problem, and on the right, the final chosen solution and the objective function values are visualized. With the optimization run concluded and the final solution selected, the DM cannot continue with this finished process but may initiate a new optimization run if desired.

6 Discussion

The primary goal of this study was to develop a UI for a selected interactive evolutionary multiobjective optimization method within the DESDEO software framework. The newly developed UI allows the DM to effectively input their preference information through interactive components for each respective preference information type. The UI also utilizes dynamic visualizations for facilitating simultaneous visual analysis, which aims to help the DM in understanding the impact of their preferences on the optimization method and better guide the optimization process towards their preferred areas of interest or importance.

The UI was successfully integrated into the existing DESDEO software framework, maintaining consistency with its overall solution process flow, visual style, and design patterns. By successfully implementing this UI for a single interactive evolutionary multiobjective optimization method, it sets a foundation for further expansion and reusability in developing additional UIs for other methods. The modular design of the UI is crucial for the ongoing development of DESDEO, as it facilitates the addition of new features and the reusability of the new components with relative ease.

The UI was developed through an iterative process, improved by invaluable feedback from the members of the DESDEO development team. This collaborative approach led to significant improvements in the UI's design, usability, and functionality, ensuring that the final result closely matched the initial vision for the UI while also accommodating lessons learned along the way. Preliminary testing and demonstrations with the UI were conducted, but more extensive user testing is needed to further refine and validate it.

The literature review conducted as part of this study underscored the importance of effectively incorporating the DM's preferences directly into the optimization process to refine and target the search more effectively. It also highlighted the value of visual aids in complex decision-making, emphasizing how these tools can significantly enhance the understanding and navigation of multiobjective optimization challenges.

However, the development of the UI was not without its challenges and limitations. One significant issue was maintaining the modularity of the components. Despite the aim for a modular design, deviations from this principle occasionally occurred, disrupting the styling and reusability of the components initially. These deviations were primarily due to prior inexperience with the DESDEO software framework and some of the development technologies, such as Svelte.

As a result, it was often more natural to develop and test new components and functionalities directly tied to the existing code. This approach, while expedient in the short term, necessitated occasional and somewhat clumsy refactoring of these functionalities into their own separate components to restore and maintain the required modularity. This process underscored the need for more structured development practices and a deeper familiarity with the technology stack used.

Additionally, some challenges arose from the integration of existing DESDEO components into the new UI, particularly with scaling issues that affected the components' sizes and layouts. The core of these issues was that some components did not correctly fit or scale to their allocated spaces and containers in the new UI. Consequently, resolving these issues required modifications to the components themselves rather than adjustments to the new UI. However, it is unknown whether these modifications might have unexpected results in other contexts where the modular components are used, necessitating close inspection of the changes required in those components.

Understanding these challenges and limitations is crucial for setting the stage for future improvements and ensuring the continued evolution of the DESDEO software framework. This insight underscores the importance of revisiting component design to enhance compatibility and responsiveness, thereby supporting more seamless integration in future developments.

7 Conclusions

This study contributed to the DESDEO software framework by bridging the gap for the need of a UI for an interactive evolutionary multiobjective optimization method. The developed UI was designed to facilitate user-centric interaction in the iRVEA method, thereby enhancing decision-making effectiveness for the DM. These goals were met, as evidenced by the positive feedback regarding the UI and the performance of the UI in demonstration contexts.

Nevertheless, there are also several areas where further research and development are necessary. One apparent future development is on connecting and testing the new UI, with the upcoming version of the DESDEO backend. This integration is crucial to ensure that the UI functions seamlessly with the updated backend systems and can handle real-world data effectively. The UI developed in this study serves as a solid baseline for implementing similar UIs for other interactive evolutionary multiobjective optimization methods. Future developments can leverage the modular design of this UI to create additional UIs for various other optimization methods.

Currently, the final solution page serves a limited role due to initially undefined needs and requirements. To address this, there is a need for detailed specifications and further development aimed at refining its functionality and interaction capabilities. Further developments include integration with other methods and improved interactive elements to ensure that this page not only displays the final solution but also supports capabilities for deeper analysis and further actions with the solution. This would enable a more dynamic and informative final solution page that better assists the DM in validating their final decision.

Another aspect requiring further attention is the settings modal window. As described earlier, this modal window allows the DM to adjust UI settings to suit their preferences. However, this functionality is currently implemented with only the option for setting decimal precision in the UI. To refine its purpose and ensure it can effectively support a broader range of settings, this modal window requires more detailed specifications and development. Future enhancements should focus on expanding the range of customizable settings, such as color themes, font sizes, and interface layouts, which are vital for accommodating diverse

user needs and preferences. By doing so, the settings can offer a more personalized user experience.

These areas of future work highlight the need for ongoing research and development to increasing the potential of the DESDEO software framework. By addressing these issues, the UI and other aspects of the DESDEO software framework can be refined to better serve the needs of its users and to further enhance the field of interactive evolutionary multiobjective optimization.

Acknowledgements

This study was conducted for and in collaboration with members of the *Multiobjective Optimization Group* at the University of Jyväskylä of whom DESDEO is mainly developed by. While the group did not directly conduct the study, the nature of the study and its topic necessitated regular discussions, information exchange, and resource assistance. This assistance included providing the UI design documents, which served as the basis for the new UI implementation.

While conducting this thesis, generative AI ChatGPT, was utilized at times in accordance with the ethical guidelines set by the University of Jyväskylä. The AI was employed to assist in forming the structure and flow of the thesis, providing recommendations for additions, and ensuring the use of proper grammar. Additionally, during the technical development of the UI, the AI was used to aid with syntax and troubleshoot technical difficulties.

Bibliography

- Aghaei Chadegani, A., H. Salehi, M. Yunus, H. Farhadi, M. Fooladi, M. Farhadi, and N. Ale Ebrahim. 2013. "A Comparison Between Two Main Academic Literature Collections: Web of Science and Scopus Databases". *Asian Social Science* 9:18–26.
- Branke, J., K. Deb, K. Miettinen, and R. Słowiński, editors. 2008. *Multiobjective Optimization: Interactive and Evolutionary Approaches*. Springer.
- Cheng, R., Y. Jin, M. Olhofer, and B. Sendhoff. 2016. "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization". *IEEE Transactions on Evolutionary Computation* 20:773–791.
- Coello, C., D. Veldhuizen, and G. Lamont. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer.
- De Angeli, A., A. Sutcliffe, and J. Hartmann. 2006. "Interaction, Usability and Aesthetics: What Influences Users' Preferences?" In *Proceedings of the 6th Conference on Designing Interactive Systems*, 271–280.
- Deb, K. 2001. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons.
- Deb, K., and S. Chaudhuri. 2006. "I-MODE: An Interactive Multi-objective Optimization and Decision-Making Using Evolutionary Methods". In *Evolutionary Multi-Criterion Optimization*, 788–802. Springer.
- Durillo, J.J., and A.J. Nebro. 2011. "JMetal: A Java Framework for Multi-Objective Optimization". *Advances in Engineering Software* 42:760–771.
- Duro, J.A., Y. Yan, I. Giagkiozis, S. Giagkiozis, S. Salomon, D.C. Oara, A.K. Sriwastava, et al. 2021. "Liger: A Cross-Platform Open-Source Integrated Optimization and Decision-Making Environment". *Applied Soft Computing* 98:106851.
- Eskelinen, P., K. Miettinen, K. Klamroth, and J. Hakanen. 2010. "Pareto Navigator for Interactive Nonlinear Multiobjective Optimization". *OR Spectrum* 32:211–227.

- Findlater, L., and J. McGrenere. 2004. "A Comparison of Static, Adaptive, and Adaptable Menus". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 89–96.
- Florida Atlantic University Libraries. 2024. "Nursing - Literature Reviews & Systematic Reviews". Visited on January 23, 2024. <https://libguides.fau.edu/c.php?g=966176&p=7013756>.
- Ghosh, A., K. Deb, R. Averill, and E. Goodman. 2023. "IK-EMOViz: An Interactive Knowledge-Based Evolutionary Multi-Objective Optimization Framework". In *Proceedings of the Evolutionary Multi-Criterion Optimization: 12th International Conference (EMO 2023)*, 606–619.
- Gong, M., F. Liu, Z. Wei, L. Jiao, and Q. Zhang. 2011. "Interactive MOEA/D for Multi-Objective Decision Making". In *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, 721–728.
- Hakanen, J., T. Chugh, K. Sindhya, Y. Jin, and K. Miettinen. 2016. "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 (SSCI)*, 1–8.
- Hakanen, J., K. Miettinen, and K. Matković. 2020. "Task-Based Visual Analytics for Interactive Multiobjective Optimization". *Journal of the Operational Research Society* 72:1–18.
- Hakanen, J., S. Radoš, G. Misitano, B.S. Saini, K. Miettinen, and K. Matković. 2022. "Interactivized: Visual Interaction for Better Decisions with Interactive Multiobjective Optimization". *IEEE Access* 10:33661–33678.
- Hartmann, J., A. Sutcliffe, and A. De Angeli. 2007. "Investigating Attractiveness in Web User Interfaces". In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 387–396.
- Hearst, M., and M. Tory. 2019. "Would You Like a Chart with That? Incorporating Visualizations into Conversational Interfaces". In *2019 IEEE Visualization Conference (VIS)*, 1–5.

- Hettenhausen, J., A. Lewis, and T. Kipouros. 2014. “A Web-Based System for Visualisation-Driven Interactive Multi-Objective Optimisation”. *Procedia Computer Science* 29:1915–1925.
- Hettenhausen, J., A. Lewis, and S. Mostaghim. 2010. “Interactive Multi-Objective Particle Swarm Optimization with Heatmap-Visualization-Based User Interface”. *Engineering Optimization* 42:119–139.
- Jaimes, A.L., A.A. Montaña, and C.A. Coello. 2011. “Preference Incorporation to Solve Many-Objective Airfoil Design Problems”. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, 1605–1612.
- Jin, X., and R. Fu. 2023. “Study on the Optimization of Digital Content Reading Experience Based on Scene Theory”. In *HCI International 2023 – Late Breaking Papers: 25th International Conference on Human-Computer Interaction*, 47–62.
- Kollat, J.B., and P. Reed. 2007. “A Framework for Visually Interactive Decision-Making and Design Using Evolutionary Multi-Objective Optimization (VIDEO)”. *Environmental Modelling and Software* 22:1691–1704.
- Lárraga, G., and K. Miettinen. 2023. “Component-Based Thinking in Designing Interactive Multiobjective Evolutionary Methods”. In *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 1693–1702.
- Luenberger, D.G. 1969. *Optimization by Vector Space Methods*. John Wiley & Sons.
- Miettinen, K. 1999. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers.
- Misitano, G., B.S. Saini, B. Afsar, B. Shavazipour, and K. Miettinen. 2021. “DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization”. *IEEE Access* 9:148277–148295.
- Peissner, M., and R. Edlin-White. 2013. “User Control in Adaptive User Interfaces for Accessibility”. *Human-Computer Interaction* 8117:623–640.
- Pierre, D.A. 1986. *Optimization Theory with Applications*. Courier Corporation.

- Reinecke, K., T. Yeh, L. Miratrix, R. Mardiko, Y. Zhao, J. Liu, and K. Gajos. 2013. “Predicting Users’ First Impressions of Website Aesthetics with a Quantification of Perceived Visual Complexity and Colorfulness”. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2049–2058.
- Ruiz, A.B., F. Ruiz, K. Miettinen, L. Delgado-Antequera, and V. Ojalehto. 2019. “NAUTILUS Navigator: Free Search Interactive Multiobjective Optimization Without Trading-Off”. *Journal of Global Optimization* 74:213–231.
- Saini, B.S., M. Emmerich, A. Mazumdar, B. Afsar, B. Shavazipour, and K. Miettinen. 2022. “Optimistic NAUTILUS Navigator for Multiobjective Optimization with Costly Function Evaluations”. *Journal of Global Optimization* 83:865–889.
- Sato, H., K. Tomita, and M. Miyakawa. 2015. “Preferred Region Based Evolutionary Multi-Objective Optimization Using Parallel Coordinates Interface”. In *Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)*, 33–38.
- Sekizaki, S., I. Nishizaki, and T. Hayashida. 2019. “A Many-Objective Evolutionary Algorithm Incorporating Decision Maker’s Preference and Its Application to Management of the Electricity Distribution Network”. *Journal of Multi-Criteria Decision Analysis* 26:165–186.
- Tan, K., T. Lee, D. Khoo, and E. Khor. 2001. “A Multi-Objective Evolutionary Algorithm Toolbox for Computer-Aided Multi-Objective Optimization”. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics* 31:537–556.
- Tan, K.C., T.H. Lee, D. Khoo, E.F. Khor, and R.S. Kannan. 2000. “MOEA Toolbox for computer aided multi-objective optimization”. In *Proceedings of the 2000 Congress on Evolutionary Computation*, 38–45.
- Tarhkanen, S., K. Miettinen, J. Hakanen, and H. Isomäki. 2013. “Incremental User-Interface Development for Interactive Multiobjective Optimization”. *Expert Systems with Applications* 40:3220–3232.
- Tian, Y., R. Cheng, X. Zhang, and Y. Jin. 2017. “PlatEMO: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]”. *IEEE Computational Intelligence Magazine* 12:73–87.

Wierzbicki, A.P. 1999. *Multicriteria Decision Making: Advances in MCDM Models, Algorithms, Theory, and Applications*. Edited by T. Gal, T.J. Stewart, and T. Hanne. 237–275. Springer.

Xin, B., L. Chen, J. Chen, H. Ishibuchi, K. Hirota, and B. Liu. 2018. “Interactive Multiobjective Optimization: A Review of the State-of-the-Art”. *IEEE Access* 6:41256–41279.