

Developing user interfaces for interactive multiobjective optimization

Computational Science Seminar

Giovanni Misitano¹

¹University of Jyväskylä (Finland), The Multiobjective Optimization Group

August 23, 2022



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ



- 1 Interactive multiobjective optimization
- 2 DESDEO and its web API
- 3 Our frontend
- 4 A component library
- 5 Conclusions

Interactive multiobjective optimization

- 1 Interactive multiobjective optimization
- 2 DESDEO and its web API
- 3 Our frontend
- 4 A component library
- 5 Conclusions

- We can define a multiobjective optimization problem with¹ k objective functions $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i \in [1, k]$, and n decision variables in a decision vector $\mathbf{x} \ni x_j, j \in [1, n]$ as

Multiobjective optimization problem

$$\begin{aligned} \min F(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})) \\ \text{s.t. } \mathbf{x} &= (x_1, x_2, \dots, x_n) \in X \subseteq \mathbb{R}^n. \end{aligned} \tag{1}$$

- In (1), X is the *feasible set* of solutions. The feasible set is usually defined by constraints.
- $F(\mathbf{x})$, when computed, is called an objective vector, often denoted as $\mathbf{z} = F(\mathbf{x})$.

¹Kaisa Miettinen. *Nonlinear multiobjective optimization*. Boston: Kluwer Academic Publishers, 1999.

Pareto optimality I

- There are two important definitions for optimality in multiobjective optimization: *Pareto optimality* and *weak Pareto optimality*. These are defined as follows:

Pareto optimality

A solution \mathbf{x}^* is said to be Pareto optimal if, and only if, \mathbf{x}^* is feasible and there exists **no other** feasible solution \mathbf{x} such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*) \forall i \in [1, k]$ and $f_i(\mathbf{x}) < f_i(\mathbf{x}^*) \exists i \in [1, k]$.

Weak Pareto optimality

A solution \mathbf{x}^* is said to be weakly Pareto optimal if, and only if, \mathbf{x}^* is feasible and there exists **no other** feasible solution \mathbf{x} such that $f_i(\mathbf{x}) < f_i(\mathbf{x}^*) \forall i \in [1, k]$.

- The set of all Pareto optimal solutions to a multiobjective optimization problem can be defined as:

The Pareto optimal set and front

The set P consisting of all Pareto optimal solutions to a multiobjective optimization problem (1) is known as the *Pareto optimal set*. The image of the Pareto optimal set P is defined as the *Pareto optimal front* Z^{Pareto} . The Pareto optimal set is a subset of the feasible solutions S , i.e., $P \subseteq S$. Furthermore, if we define Z as the image of the feasible set, then the Pareto optimal front is a subset of this set, i.e., $Z^{Pareto} \subseteq Z$.

- The unique feature of Pareto optimal solutions is that they cannot be compared purely on basis of mathematics—we need some additional information to completely order these solutions!

Pareto optimality III

- Lower (best) and upper (worst) of the objective function values present in a Pareto optimal front are characterized by the *ideal* and *nadir* points, respectively. These are defined below.

Ideal point

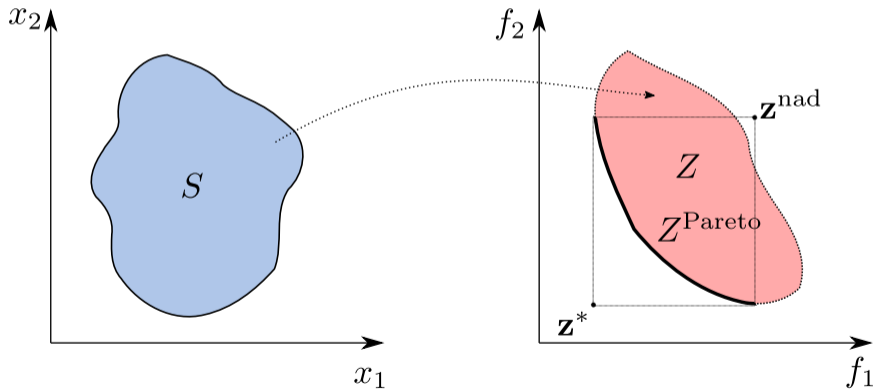
The ideal point \mathbf{z}^* of a Pareto optimal front for some problem (1) is defined as the vector consisting of the lowest individual values for each objective function when each objective function is minimized separately.

Nadir point

The nadir point $\mathbf{z}^{\text{nadir}}$ is defined as the highest individual values found for each objective function in the Pareto optimal front for some problem (1).

Pareto optimality IV

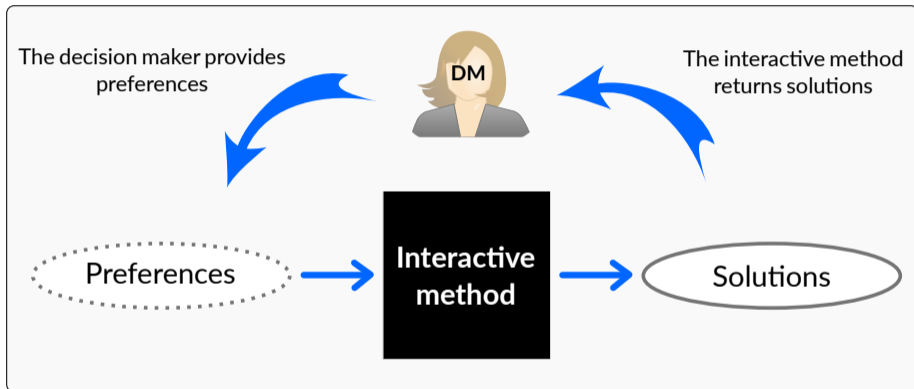
$$\min_{\mathbf{x} \in S} \{f_1(\mathbf{x}), f_2(\mathbf{x})\}$$



Interactive multiobjective optimization I

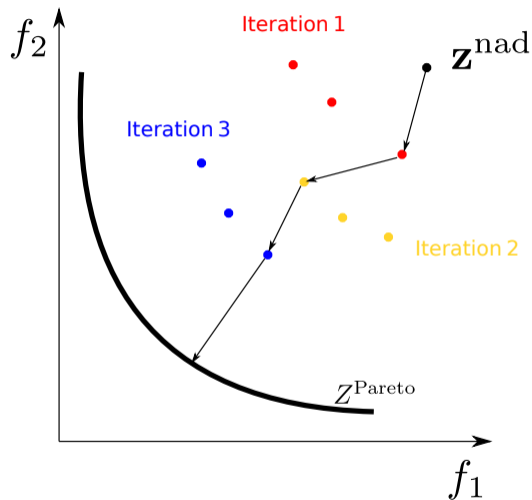
- Multiobjective optimization problems have multiple Pareto optimal solutions with various trade-offs.
- To select the *best* solution is up to a domain expert, the *decision maker* or DM.
- The DM can express their preferences, which are then used to find the most satisfying solution(s) to the problem at hand.
- Preferences can be expressed before (*a priori*) the optimization, after (*a posteriori*) the optimization, or interactively during the optimization.
- Interactive multiobjective optimization allows the DM to explore the problem, learn about it, and learn about one's own preferences as well.

Interactive multiobjective optimization II



- Preferences can be expressed in many ways:
 - As aspiration levels, which are desired values for each objective function. These form a reference point \bar{z} .
 - As upper and lower limits for each objective function.
 - By classifying each objective function into various classes. E.g., should increase from its current value, should decrease from its current value, should stay the same, etc. . .
 - By selecting preferred/non-preferred solution from a set of solutions.
 - Many more. . .

Example: E-NAUTILUS²



²Ana B. Ruiz et al. "E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method". In: *European Journal of Operational Research* 246.1 (2015), pp. 218–231. doi: 10.1016/j.ejor.2015.04.027.

DESDEO and its web API

- 1 Interactive multiobjective optimization
- 2 DESDEO and its web API**
- 3 Our frontend
- 4 A component library
- 5 Conclusions

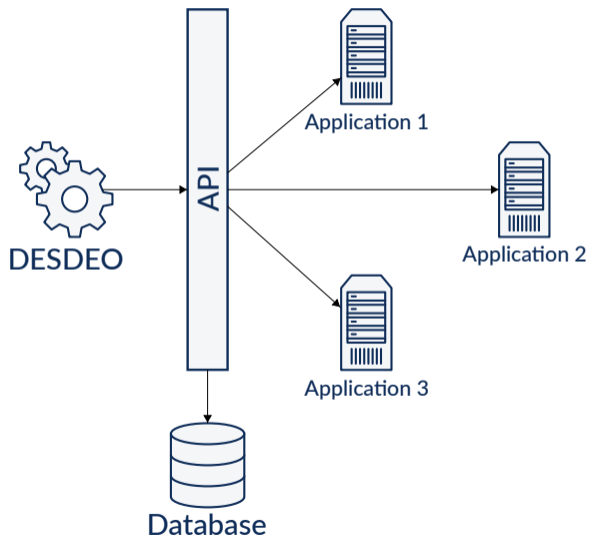
- DESDEO is a modular and open source Python software framework for interactive multiobjective optimization.
- Contains many interactive methods (scalarization- and evolutionary-based methods), support for modelling multiobjective optimization problem, and much more³.
- DESDEO's goal is to bring together many different methods in a modular environment to ease the application of and experimentation with these methods.
- Webpage: <https://desdeo.it.jyu.fi>; on GitHub: <https://github.com/industrial-optimization-group>

³G. Misitano et al. "DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization". In: *IEEE Access* 9 (2021), pp. 148277–148295. DOI: 10.1109/ACCESS.2021.3123825.

- Because DESDEO is implemented in Python, it is mostly limited to the packages and software infrastructure that support Python.
- Interactive multiobjective optimization methods DMs can be very demanding in what they require from a user interface.
- Web-based interfaces are very popular and accessible. There are very powerful software libraries to implement rich and very interactive interfaces.
- However, these powerful libraries are often implemented in JavaScript. . .

- Solution: build a web API (application programming interface) for DESDEO allowing any program with basic support for HTTP calls to interface into DESDEO and what it has to offer!
- The API (*backend*) is build using the Python library Flask (<https://flask.palletsprojects.com/en/2.2.x/>).
- Access to a database, allows user accounts to store problems, solutions, methods state, etc. . .
- Only requirements is that the API is hosted somewhere and that the end application (*frontend*) can handle basic HTTP requests.

The web API III



Our frontend

- 1 Interactive multiobjective optimization
- 2 DESDEO and its web API
- 3 Our frontend**
- 4 A component library
- 5 Conclusions

Why the hassle? I

- Different DMs dealing with different multiobjective optimization problems have different needs.
- This means that we need different user interfaces (UI) depending on these needs.
- To complicate things, different interactive multiobjective optimization methods also have their own needs...
- DESDEO has the implementation of different methods and the modelled problem, but makes no assumption of the UI.



No ring to rule them all in this case :(

A React application I

- We want a powerful and flexible library to build just the kind of interfaces we want/need.
- React (<https://reactjs.org/>) is a very well established JavaScript library to build just this kind of interfaces.
- And because everybody hates JavaScript (**fact!**), we decided to use TypeScript (<https://www.typescriptlang.org/>) instead!

A React application II

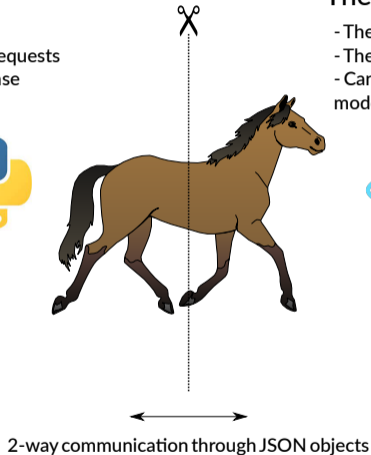
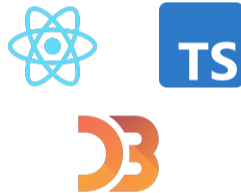
The back-end

- The side hiding from the user
- A web server handling HTTP requests
- Information stored in a database



The front-end

- The side facing the user
- The thing rendered in your web browser
- Can be accessed by any device with a modern web browser



The hassle was worth it!

- One challenge with interactive multiobjective optimization methods, is their comparison. How to compare them?
- What is good and what is bad, is subjective—it depends on the DM. . .
- Subject studies are needed. We have designed and conducted such experiments [4].
- We were able to implement just the kind of interface we wanted and to integrate a questionnaire in the interface. We were also able to record multiple other metrics, like time spent in a method, time spend answering questions, etc. . .

DEMO time!

The future is beautiful: Beautification

- We are currently developing a general interface for our methods to bring multiobjective optimization closer to DMs from different fields and regular people.
- We are putting a lot of effort into designing a sound interface with common elements and a unified style.
- We are collaborating with a cognitive scientists.

A component library

- 1 Interactive multiobjective optimization
- 2 DESDEO and its web API
- 3 Our frontend
- 4 A component library**
- 5 Conclusions

- In addition to just develop our own interface, we would like to provide a library of interactive visualizations to be used in other web-based (React) interfaces.
- We need to build our custom interactive visualizations simply for the reason that the kind of visualization that we need do not exist!
- We build these visualization components pretty much from the ground up utilizing D3.js (<https://d3js.org/>).
- The component library is also undergoing a beautification overhaul...

Another DEMO (kinda)!

Conclusions

- 1 Interactive multiobjective optimization
- 2 DESDEO and its web API
- 3 Our frontend
- 4 A component library
- 5 **Conclusions**

- Interactive multiobjective optimization methods have quite specific needs that can vary a lot depending on the needs of the DM or the method itself.
- DESDEO has many implementations of interactive methods, but has no (graphical) UI.
- We wanted to use the best available tools to build flexible and feature rich interfaces—these are not available natively in Python.
- A web API allows us to interface into DESDEO from almost an application (and software library).
- We have build our own interfaces for our own needs, developed a library of interactive visual components, and we are developing a new and modern interface to bring multiobjective optimization closer to the masses.

- DESDEO on GitHub:
<https://github.com/industrial-optimization-group/DESDEO>
- DESDEO's homepage: <https://desdeo.it.jyu.fi>
- DESDEO web API:
<https://github.com/industrial-optimization-group/desdeo-webapi>
- DESDEO components library:
<https://github.com/industrial-optimization-group/desdeo-webapi>
- Our frontend for DESDEO:
<https://github.com/industrial-optimization-group/desdeo-frontend>
- Lecture about DESDEO and more:
https://www.youtube.com/channel/UC6qFfAgD8_aa28pcBmKDXTw.

- Multiobjective Optimization (research) Group¹
- Follow me on LinkedIn²

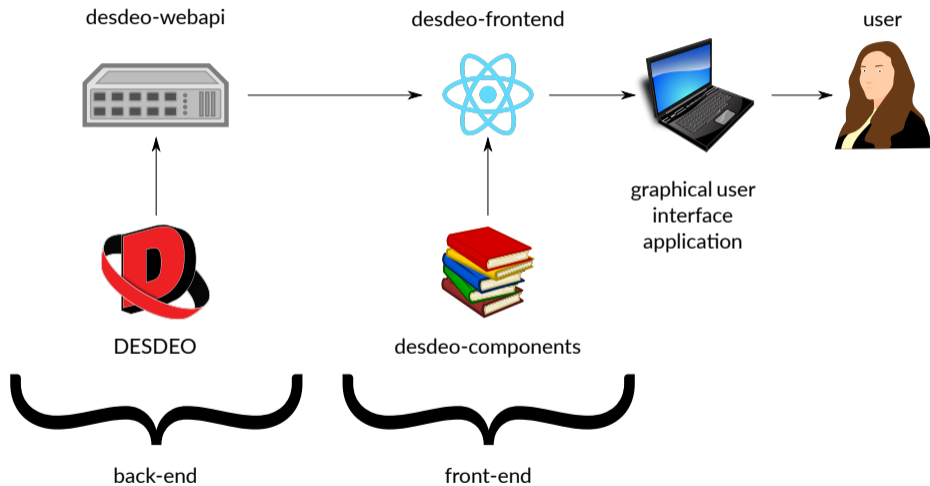
Slides online



¹<http://www.mit.jyu.fi/optgroup/>

²<https://linkedin.com/in/misitano>

The End



- [1] Kaisa Miettinen. *Nonlinear multiobjective optimization*. Boston: Kluwer Academic Publishers, 1999.
- [2] Ana B. Ruiz et al. “E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method”. In: *European Journal of Operational Research* 246.1 (2015), pp. 218–231. DOI: [10.1016/j.ejor.2015.04.027](https://doi.org/10.1016/j.ejor.2015.04.027).
- [3] G. Misitano et al. “DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization”. In: *IEEE Access* 9 (2021), pp. 148277–148295. DOI: [10.1109/ACCESS.2021.3123825](https://doi.org/10.1109/ACCESS.2021.3123825).
- [4] Bekir Afsar et al. “Designing Empirical Experiments to Compare Interactive Multiobjective Optimization Methods”. In: *Journal of the Operational Research Society* (2022). Under review.