

DESDEO: The framework for interactive multiobjective optimization

Giovanni Misitano

`giovanni.a.misitano@jyu.fi`

University of Jyväskylä
The multiobjective optimization group

November 8, 2021



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ



- 1 Background
 - Multiobjective optimization
 - Interactive multiobjective optimization
 - Scalarization based methods
 - Evolutionary methods
- 2 DESDEO: the framework
 - Motivation: why DESDEO?
 - Packages
 - Potential: three examples
- 3 DESDEO: interfaces
 - web API
 - Components
 - The frontend
 - Links and resources
- 4 Wrapping up
- 5 Bibliography

- 1 Background
 - Multiobjective optimization
 - Interactive multiobjective optimization
 - Scalarization based methods
 - Evolutionary methods
- 2 DESDEO: the framework
- 3 DESDEO: interfaces
- 4 Wrapping up
- 5 Bibliography

- When we make choices in real-life scenarios, we often have many goals in mind, which we wish to achieve.
 - E.g., when looking for a product to buy (a new smartphone, for instance), we try to strike a balance between quality (which we wish to be as high as possible) and price (which we wish to be as low as possible).
- From experience, we know that there are often multiple available options, of which some are better regarding one goal and some regarding another goal.
- Our goals are often in mutual conflict.



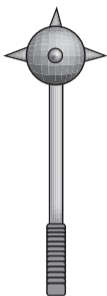
- In multiobjective optimization [1] we are interested with problems with multiple (continuous) objectives, which we wish to optimize simultaneously. Such a problem with $k \geq 2$ objectives can be defined as:

$$\begin{aligned} \min F &= [f_1(x), f_2(x), \dots, f_k(x)] \\ \text{s.t. } x &\in S, \end{aligned} \tag{1}$$

where each objective is to be minimized, F is an objective vector with k objective functions $f_i(x)$ as its components, $x \in \mathbb{R}^n$ is a decision variable vector, and $S \subset \mathbb{R}^n$ is the space of feasible decision variable vectors defined by constraints imposed on x .

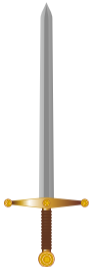
- In the non-trivial case, many solutions exist to (1).
- The set of *Pareto optimal* solutions is such a set where we cannot switch one solution to another without having to make a *trade-off* at least in one objective.
- Finding the *best* solution to (1) is mathematically an ill-defined problem.
- To find the best solution, we can use the preferences expressed by a domain expert known as the *decision maker* (DM).

Morning star



+8 str
+4 agi

2-handed sword



+6 str
+6 agi

Pareto optimal

Daggers



+4 str
+8 agi

There is no gun

Gun



+999 str
+999 str
"kills target instantly"

- In interactive multiobjective optimization the preferences of a DM are iteratively incorporated *during the optimization process*.
- Examples of preferences are aspiration levels for each objective, classifications, and desirable objective ranges, to name a few.
- Incorporating the preferences into the optimization process can help decrease computation costs and allows the DM to learn about the problem being solved (e.g., the trade-offs present), and allows them to feel confident in the solution found.

- There are two main classes of interactive multiobjective optimization methods.
- The first are scalarization based methods where the multiobjective optimization problem (1) is scalarized using, e.g., achievement scalarizing functions, into a single-objective optimization problem while accounting for the DM's preferences.
 - Weight coefficients are maybe the simplest example of this.
- Depending on the scalarization method, we can achieve exact and accurate solutions for (1) (e.g., *strongly* Pareto optimal solutions).

- NIMBUS [2]
- E-NAUTILUS [3]
- Pareto navigator [4]

- The other main class of methods are evolutionary methods.
- Utilizing mutation, cross-over and recombination, a *population* of solutions to (1) is evolved.
- The usual way for tackling computationally expensive problems.
- However, because these methods are heuristic in nature, they cannot guarantee the solutions found to be Pareto optimal.

- (interactive) RVEA [5]
- (interactive) NSGA-III [6]

- 1 Background
- 2 **DESDEO: the framework**
 - Motivation: why DESDEO?
 - Packages
 - Potential: three examples
- 3 DESDEO: interfaces
- 4 Wrapping up
- 5 Bibliography

- DESDEO [7] was developed as a framework to allow researchers and practitioners to use and apply interactive multiobjective optimization methods and to model multiobjective optimization problems.
- DESDEO is the only framework, which has both scalarization and evolutionary interactive methods implemented in the same environment.
- DESDEO is open source, modular, and implemented in Python.

DESDEO consists of four core packages:

- `desdeo-problem` – defining multiobjective optimization problems of various kinds.
- `desdeo-tools` – tools and utilities used in both scalarization based and evolutionary methods, such as achievement scalarizing functions.
- `desdeo-mcdm` – scalarization based interactive multiobjective optimization methods.
- `desdeo-emo` – evolutionary interactive (and non-interactive) multiobjective optimization methods.

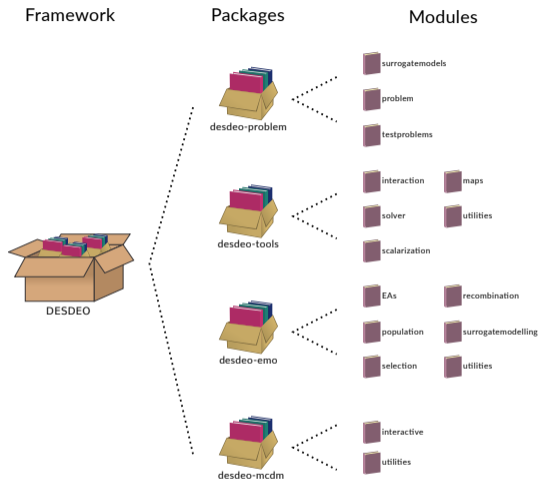


Figure: The packages found in DESDEO and their modules.

In what follows, three examples of the potential DESDEO offers are given.

Example 1: Hybridization

- Having both scalarization based interactive methods and evolutionary methods in the same framework allows for the *hybridization* of the two classes of method.
- For instance, it is possible to change methods between iterations. This allows to combine the two kind of interactive methods in exciting new ways.
- For a recent example on hybridization, see [8].

Example 2: Interactive method comparison

- Comparison of interactive multiobjective optimization methods is a great challenge because of the human aspect(s) involved [9].
- No indicators exist for comparing interactive methods.
- Having different methods available in the same framework is a step in the right direction when it comes to comparing interactive methods.
- Our group has been part of a research project where interactive methods have been compared in an in-subject study with real human participants.

Example 3: A central hub for interactive methods

- As mentioned, no other frameworks exist for interactive multiobjective optimization methods like DESDEO.
- DESDEO offers an avenue for other researchers and practitioners to:
 - Re-use existing implementation of interactive methods.
 - Implement their own methods.
 - Modify and re-use parts of existing methods, combining them into new methods.
- We have had some outside contributions to DESDEO, but we expect a lot more to come in the future.
- **You are also welcome to check DESDEO out and contribute!**

- Homepage of DESDEO – <https://desdeo.it.jyu.fi>
- Documentation – <https://desdeo.readthedocs.io/en/latest/>
- desdeo-problem –
<https://github.com/industrial-optimization-group/desdeo-problem>
- desdeo-tools –
<https://github.com/industrial-optimization-group/desdeo-tools>
- desdeo-mcdm –
<https://github.com/industrial-optimization-group/desdeo-mcdm>
- desdeo-emo –
<https://github.com/industrial-optimization-group/desdeo-emo>
- Notebook with four cases showcasing how DESDEO can be used –
https://desdeo.readthedocs.io/en/latest/notebooks/four_simple_use_cases.html

- 1 Background
- 2 DESDEO: the framework
- 3 DESDEO: interfaces**
 - web API
 - Components
 - The frontend
 - Links and resources
- 4 Wrapping up
- 5 Bibliography

- The DESDEO *framework* itself does not contain any implementations of interfaces.
- Instead, it is expected that users implement their own. This is because the requirements of the interfaces vary wildly depending on the application area, problem type, preferences of DMs', and so forth...
- That being said, in the DAEMON project we have started to develop interfaces, which are also open, modular, and easy to extend to meet one's needs.
- Utilizing the web API discussed next, we have build a web stack on top of DESDEO.

- Since DESDEO is implemented in Python, utilizing it directly with languages and libraries better suited for building modern interfaces is not easy.
- We want DESDEO to be usable in any sort of graphical application, independent of the environment.
- This is why we build a web API, which exposes some of DESDEO's functionalities to be utilized through HTTP endpoints.
- This means, for instance, that one server implementing the web API can serve multiple applications, which are able to handle HTTP requests (in practice any modern application).

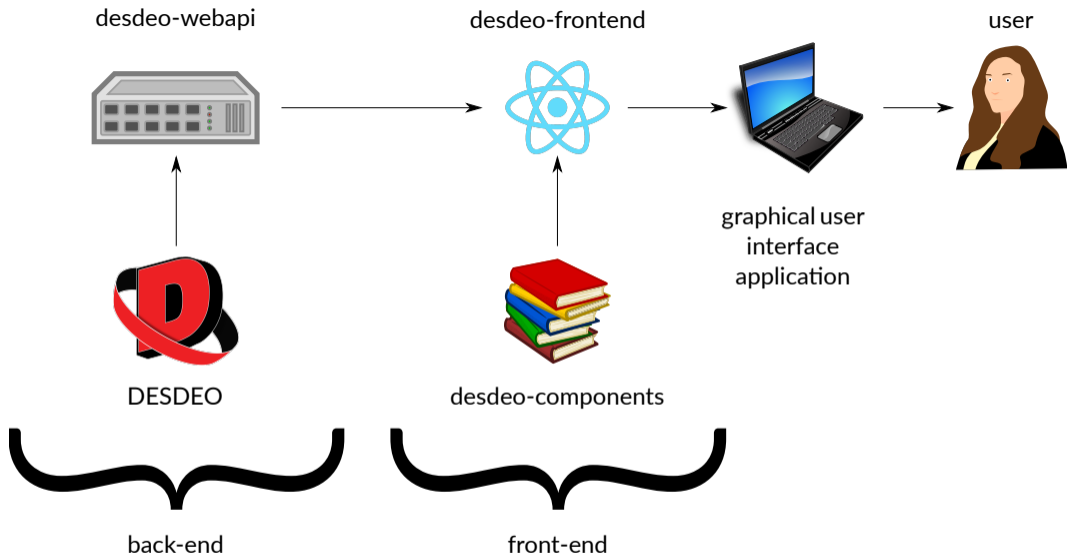


Figure: The webstack for DESDEO.

- When it comes to implementing interfaces for interactive multiobjective optimization methods, challenges arise with the visualization and interaction components required.
- Interactive methods require quite unique ways of interacting with visualizations so that a DM can express their preferences.
- For instance, a DM might want to select individual solutions from a parallel axes plot, or set aspiration levels or ranges using horizontal bar charts.
- Turns out, no ready-made solutions exist that *fully* meet our needs, which is why we decided to start building our own as React components utilizing D3.js.

- Utilizing the web API and the components library, we implemented a web-based application for DESDEO, where we have some interactive methods available to be used to solve multiobjective optimization problems.
- We chose to go with web-based interfaces because those are accessible by anyone with a relative modern web browser.
- This way the device or operating system does not limit the use of the interface.
- The frontend has been implemented in TypeScript and React.

Cue demonstration

- `desdeo-webapi` – <https://github.com/industrial-optimization-group/desdeo-webapi>
- `desdeo-components` – <https://github.com/industrial-optimization-group/desdeo-components>
- `desdeo-frontend` – <https://github.com/industrial-optimization-group/desdeo-frontend>
- TypeScript – <https://www.typescriptlang.org/>
- React – <https://reactjs.org/>
- D3.js – <https://d3js.org/>

- 1 Background
- 2 DESDEO: the framework
- 3 DESDEO: interfaces
- 4 Wrapping up**
- 5 Bibliography

In summary, we have build a framework, DESDEO, for interactive multiobjective optimization methods, which contains both scalarization and evolutionary methods. The framework offers exciting new possibilities to combine the two kinds of methods. In addition, we have started to build new web-based interfaces for the methods contained in the framework.

- 1 Background
- 2 DESDEO: the framework
- 3 DESDEO: interfaces
- 4 Wrapping up
- 5 Bibliography**

- [1] K. Miettinen. *Nonlinear Multiobjective Optimization*. Boston: Kluwer Academic Publishers, 1999.
- [2] Kaisa Miettinen and Marko M Mäkelä. “Synchronous approach in interactive multiobjective optimization”. In: *European Journal of Operational Research* 170.3 (2006), pp. 909–922.
- [3] Ana B. Ruiz et al. “E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method”. en. In: *European Journal of Operational Research* 246.1 (Oct. 2015), pp. 218–231. ISSN: 03772217. DOI: 10.1016/j.ejor.2015.04.027.
- [4] Petri Eskelinen et al. “Pareto navigator for interactive nonlinear multiobjective optimization”. In: *OR spectrum* 32.1 (2010), pp. 211–227.

- [5] Pouya Aghaei Pour et al. “Surrogate assisted interactive multiobjective optimization in energy system design of buildings”. In: (Jan. 2021). DOI: [10.1007/s11081-020-09587-8](https://doi.org/10.1007/s11081-020-09587-8). URL: <https://doi.org/10.1007/s11081-020-09587-8>.
- [6] Julian Blank, Kalyanmoy Deb, and Proteek Chandan Roy. “Investigating the normalization procedure of NSGA-III”. In: *International Conference on Evolutionary Multi-Criterion Optimization*. Springer. 2019, pp. 229–240.
- [7] G. Misitano et al. “DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization”. In: *IEEE Access* (2021), pp. 1–1. DOI: [10.1109/ACCESS.2021.3123825](https://doi.org/10.1109/ACCESS.2021.3123825).
- [8] Bhupinder Singh Saini, Jussi Hakanen, and Kaisa Miettinen. “A New Paradigm in Interactive Evolutionary Multiobjective Optimization”. In: *Parallel Problem Solving from Nature – PPSN XVI, Proceedings*. Ed. by Thomas Bäck et al. Cham: Springer, 2020, pp. 243–256. ISBN: 978-3-030-58115-2.

- [9] Bekir Afsar, Kaisa Miettinen, and Francisco Ruiz. “Assessing the Performance of Interactive Multiobjective Optimization Methods”. In: 54.4 (May 2021), pp. 1–27. DOI: [10.1145/3448301](https://doi.org/10.1145/3448301).

Questions?