DESDEO23 Forum: Tutorial 1—General structure of DESDEO

Giovanni Misitano[1]

[1]University of Jyväskylä (Faculty of Information Technology), The Multiobjective Optimization Group

June 19, 2023

JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

Giovanni Misitano

- I am Giovanni Misitano, a doctoral researcher from the Multiobjective Optimization Group here at the University of Jyväskylä.
- I am one of the main developers of DESDEO.
- For more info, visit my homepage: `http://giovanni.misitano.xyz`.
- You may also find me on LinkedIn, Twitter (`@misitano_g`), and GitHub (`gialmisi`).

Contents

Overview

Overview

- In this tutorial, we will first go through some basic background concepts needed to understand the rest of the tutorials and talks, and establish a common terminology.
- We will then discuss the structure of DESDEO, focusing on its core packages.
- Then, we will briefly discuss some packages beyond the core packages.
- Lastly, we will very briefly look into what the next tutorials will consist of.

Terminology and background concepts

Multiobjective optimization I

- We consider multiobjective optimization problems with the following form:

Multiobjective optimization problem definition

$$\min_{\mathbf{x}\in S} \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}) \dots, f_k(\mathbf{x})), \tag{1}$$

where $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is a vector of n *decision variables*; $S \subset \mathbb{R}^n$ is the *feasible set of decision variables*; and $\mathbf{f}$ is a vector valued function consisting of k *objective functions* $f_1(\mathbf{x}), f_2(\mathbf{x}) \dots, f_k(\mathbf{x})$.

- The objective functions f_i are conflicting.
- A solution $\mathbf{x}$ is *feasible* when $\mathbf{x} \in S$. S is defined by constraints.
- The image of a feasible solution, i.e., $\mathbf{f}(\mathbf{x}) = \mathbf{z}$, is known as an *objective vector*.

Multiobjective optimization II

We may have the following constraints that define S:

- box-constraints, which are lower and upper bounds for decision variables $x_i \in \mathbf{x}$;
- equality constraints of the form $h(\mathbf{x}) = C$, where C is some constant; or
- inequality constraints of the form $g(\mathbf{x}) \leq C$, or $g(\mathbf{x}) \geq C$.

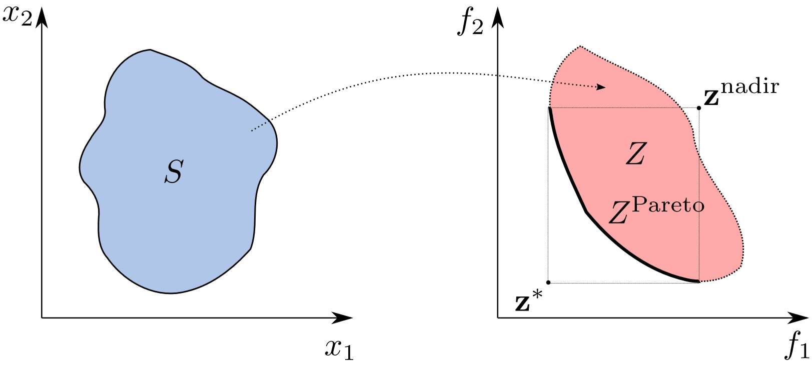
Multiobjective optimization III

- A feasible solution $\mathbf{x}'$ is *Pareto optimal* if, and only if, there exists no other feasible solution $\mathbf{x}$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}')$ for all $i = 1, \ldots, k$, and $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ for some $j = 1, \ldots, k$.
    - In other words, no other solution $\mathbf{x}$ *dominates* $\mathbf{x}'$.
- The set of all Pareto optimal solutions is known as the *Pareto optimal set* Z^{Pareto}.
- The set of all objective vectors is defined as Z, then $Z^{\text{Pareto}} \subset Z$.
- The lower and upper bounds of the Pareto optimal set are known as the *ideal point* $\mathbf{z}^*$ and the *nadir point* $\mathbf{z}^{\text{nadir}}$, respectively.

Multiobjective optimization IV

- To summarize:

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



Interactive methods

- We assume a DM to be interested only in Pareto optimal solutions (or non-dominated solutions) on the basis of *rationality*.
- Because the objective vectors of Pareto optimal solutions are incomparable, we need a *decision maker* (DM) to provide *preference information* that can be used to find the *most preferred solution* to a multiobjective optimization problem (1) being solved.
- To help a DM find their most preferred solution, we utilize *multiobjective optimization methods*.
- Based on *when* a DM provides preference information in a method, we can categorize methods into three categories[1]:
  - *a priori methods* (preferences given *before* optimization),
  - *a posteriori methods* (preferences given *after* optimization), and
  - **interactive methods** (preferences given **during** optimization).

[1] Method that make no use of preference information are known as *no-preference methods*. These methods do not interest us.

Interactive methods I

- Interactive methods, and multiobjective optimization methods in general, can be roughly divided into two categories:
  - *scalarization-based* methods and
  - *evolutionary* methods.
- In scalarization-based methods, the problem (1) being solved is transformed into a single-objective optimization problem, which can be readily solved for a Pareto optimal solution.
- Evolutionary methods make use of evolutionary operators that are suitable for multiobjective optimization problems.
- When implemented as interactive methods, in both types of method, preference information provided by the DM is utilized one way or another to find solutions.

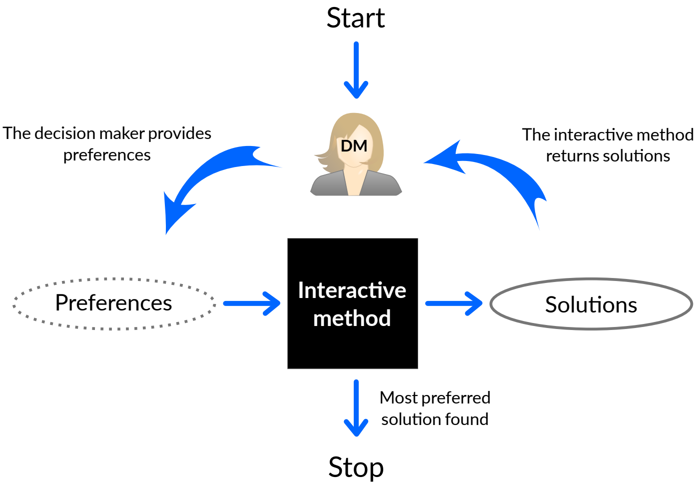
Interactive methods II

- Preferences may come in many different forms. To give a few examples:
  - *a reference point* $\bar{\mathbf{z}}$ consists of aspiration levels for each objective function being optimized,
  - *upper and lower limits*, or *reservation levels*, representing acceptable ranges can be defined for each objective function,
  - each objective function can be *classified* into different classes[2]; and
  - *preferred* and *non-preferred* solutions can be indicated.
- Different DMs may prefer to give different types of preference information.

[2]Objective functions that should: i. *improve*, ii. *improve until some limit is reached*, iii. *stay as it is*, iv. *impair until some limit is reached*, and v. *change freely*.

Interactive methods III

- To summarize:



DESDEO core packages

DESDEO I

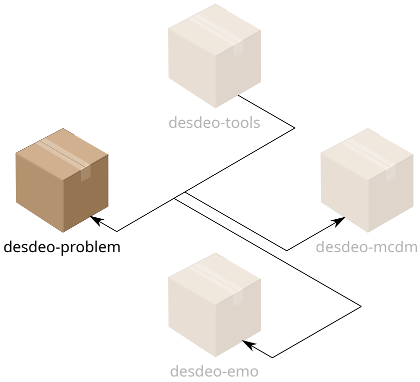
- DESDEO is a modular, open source software framework for interactive multiobjective optimization implemented mainly in Python[3].
- DESDEO is available and openly distributed on GitHub https://github.com/industrial-optimization-group/DESDEO
- The *core packages* in DESDEO contain the algorithms and logic needed to implement various interactive multiobjective optimization methods and to model problems.
- Each core package consists of specialized *modules*.

[3] G. Misitano, B. S. Saini, B. Afsar, B. Shavazipour, and K. Miettinen. "DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization". *IEEE Access* 9 (2021), pp. 148277–148295. DOI: 10.1109/ACCESS.2021.3123825.

Why DESDEO?

- To have multiple interactive methods of various types under the same framework.
- To ease switching between different interactive methods.
- To ease the comparison of interactive methods.
- To have a documented framework.
- To ease experimenting with and applying different interactive methods
- And more!

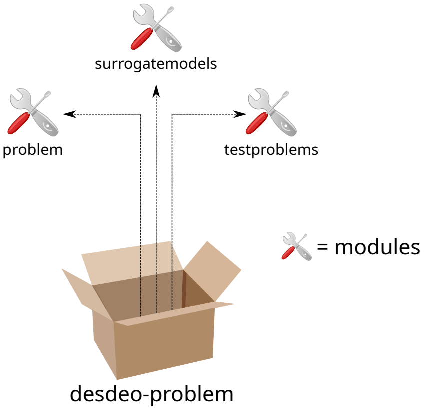
desdeo-problem I



desdeo-problem II

`desdeo-problem` contains:

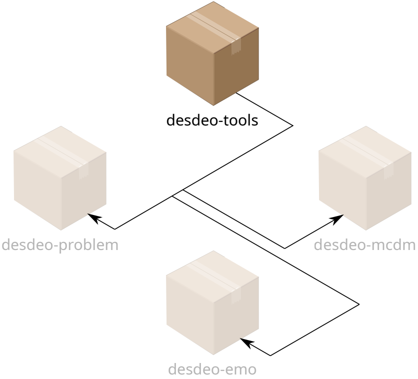
- tools and utilities to model multiobjective optimization problems;
- various pre-defined test problems; and
- tools to model problems based on data, i.e., surrogate models.



desdeo-problem III

- Types of problem supported:
  - analytical problems,
  - data-based problems, and
  - simulation-based problems.
- Currently, only continuous variables are supported (but we are working on it!).

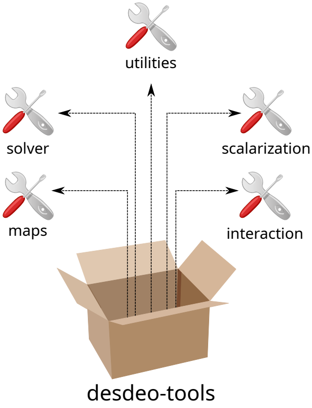
desdeo-tools I

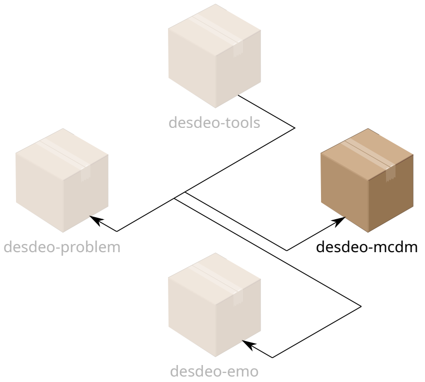


desdeo-tools II

desdeo-tools contains:

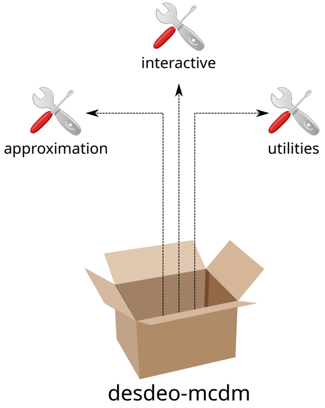
- (interfaces to) single-objective optimization solvers;
- tools to scalarize multiobjective optimization problems;
- tools to facilitate interaction between a DM and an interactive method;
- various mappings to transform multiobjective optimization problems; and
- other miscellaneous tools, e.g., routines to compute ideal and nadir points.




desdeo-tools
desdeo-problem
desdeo-mcdm
desdeo-emo

desdeo-mcdm contains:

- scalarization-based interactive methods, including trade-off free methods and navigation methods;
- methods to approximate the Pareto optimal front; and
- various utilities needed in scalarization-based methods.



Some examples of interactive methods implemented in `desdeo-mcdm` include:

- Synchronous NIMBUS[4],
- E-NAUTILUS[5],
- NAUTILUS Navigator[6], and
- The Reference Point Method[7].

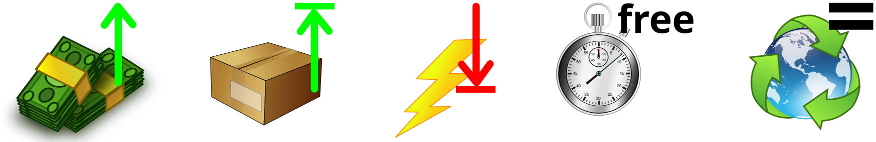
[4] Kaisa Miettinen and Marko M. Mäkelä. "Synchronous approach in interactive multiobjective optimization". *European Journal of Operational Research* 170.3 (2006), pp. 909–922. DOI: 10.1016/j.ejor.2004.07.052.

[5] Ana B. Ruiz, Karthik Sindhya, Kaisa Miettinen, Francisco Ruiz, and Mariano Luque. "E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method". *European Journal of Operational Research* 246.1 (2015), pp. 218–231. DOI: 10.1016/j.ejor.2015.04.027.

[6] Ana B. Ruiz, Francisco Ruiz, Kaisa Miettinen, Laura Delgado-Antequera, and Vesa Ojalehto. "NAUTILUS Navigator: free search interactive multiobjective optimization without trading-off". *Journal of Global Optimization* 74.2 (2019), pp. 213–231. DOI: 10.1007/s10898-019-00765-2. URL: https://doi.org/10.1007/s10898-019-00765-2.

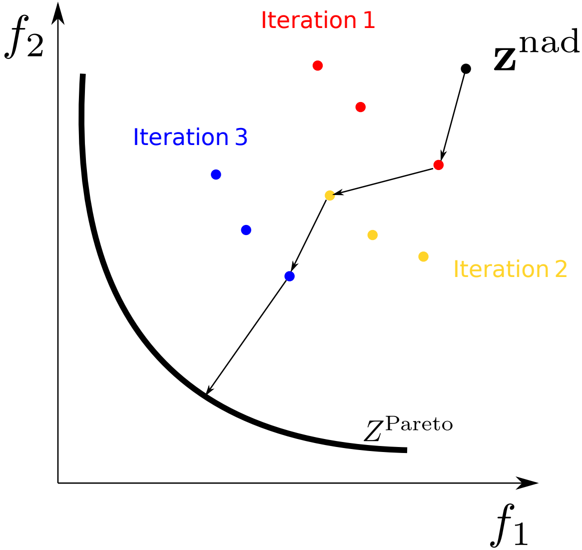
[7] Andrzej P Wierzbicki. "A mathematical basis for satisficing decision making". *Mathematical Modelling* 3 (1982), pp. 391–405.

NIMBUS

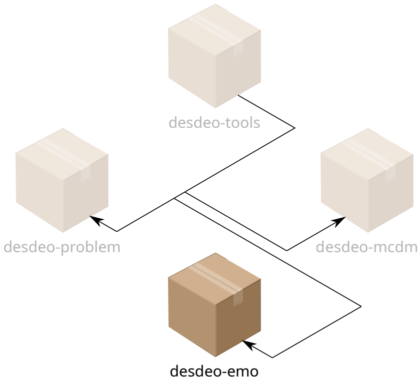


500 units   200 €/kwh   50000kg (CO2)

Trade-off free methods



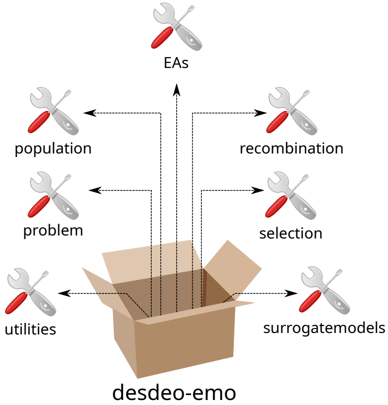
desdeo-emo I



desdeo-emo II

desdeo-emo contains:

- evolutionary algorithms and operators to implement interactive methods;
- tools to model populations for evolutionary algorithms;
- surrogate models specific to evolutionary methods; and
- various utilities needed in evolutionary interactive methods.



desdeo-emo III

Some examples of interactive methods implemented in `desdeo-emo` include:

- PBEA[8], O-NAUTILUS[9], interactive RVEA[10], interactive NSGA-III[11], and an implementation of the PIS[12] paradigm.

For how RVEA and NSGA-III have been made interactive, see[13].

---

[8] Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. "A preference-based evolutionary algorithm for multi-objective optimization". *Evolutionary computation* 17.3 (2009), pp. 411–436.

[9] Bhupinder Singh Saini, Michael Emmerich, Atanu Mazumdar, Bekir Afsar, Babooshka Shavazipour, and Kaisa Miettinen. "Optimistic NAUTILUS navigator for multiobjective optimization with costly function evaluations". *Journal of Global Optimization* (2022), pp. 1–25.

[10] Ran Cheng, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization". *IEEE Transactions on Evolutionary Computation* 20.5 (2016), pp. 773–791. DOI: `10.1109/TEVC.2016.2519378`.

[11] Kalyanmoy Deb and Himanshu Jain. "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints". *IEEE Transactions on Evolutionary Computation* 18.4 (2014), pp. 577–601. DOI: `10.1109/TEVC.2013.2281535`.

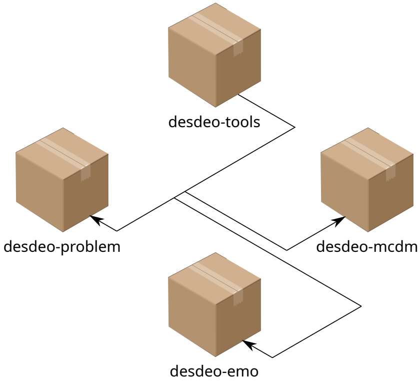
[12] Bhupinder Singh Saini, Jussi Hakanen, and Kaisa Miettinen. "A New Paradigm in Interactive Evolutionary Multiobjective Optimization". *Parallel Problem Solving from Nature – PPSN XVI*. ed. by Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann. Cham: Springer International Publishing, 2020, pp. 243–256.

[13] Jussi Hakanen, Tinkle Chugh, Karthik Sindhya, Yaochu Jin, and Kaisa Miettinen. "Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms". *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. Athens, Greece: IEEE, 2016, pp. 1–8. DOI: `10.1109/SSCI.2016.7850220`.

Modularity

"All together now!"

- Together, the core packages and the contained modules can be used selectively together to experiment with existing and new interactive methods.
- Having both scalarization-based and evolutionary methods in the same framework eases the *hybridization* of different kinds of methods.
- No need to reinvent the wheel...

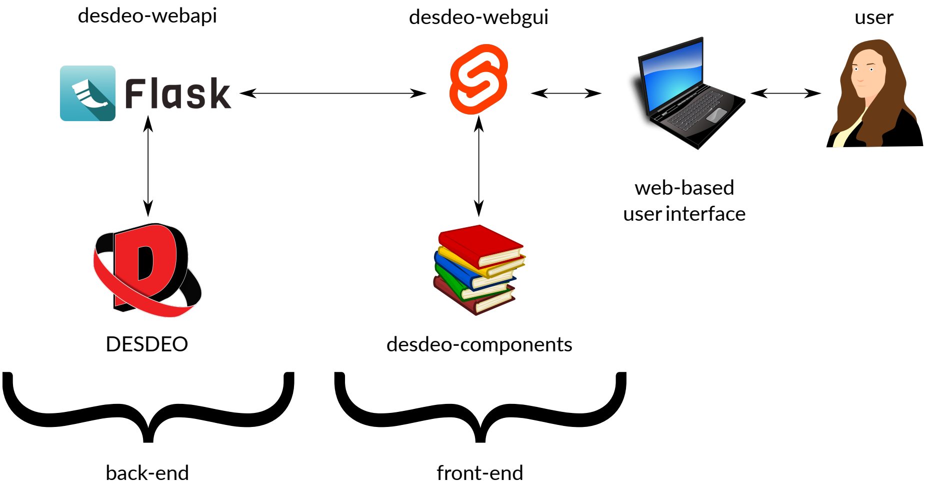


The “DESDEO ecosystem”

The “DESDEO ecosystem” I

- In addition to the core packages of DESDEO, we have also been developing other packages and tools that facilitate using the interactive methods found in DESDEO in other applications, especially graphical user interfaces.
- We have been developing a web API (application programming interface) and a web-based user interface.

The “DESDEO ecosystem” II



The "DESDEO ecosystem" II

- The web API and the user interface are currently under active development and change often. Therefore, they are quite volatile.
- That being said, the interested audience member is welcome to explore what we currently have on GitHub:
  - Web API: `https://github.com/industrial-optimization-group/desdeo-webapi`
  - Web-based user interface: `https://github.com/industrial-optimization-group/desdeo-webui`
  - Older, React based user interface: `https://github.com/industrial-optimization-group/desdeo-frontend` (deprecated)

Time to act now!

- DESDEO is under active development *right now*!
- We are building towards a *minimum viable product* to be delivered at the end of August 2023.

Next

Next

- In the upcoming tutorials, we will see how one can contribute to DESDEO, and how DESDEO can be used in practice to model and solve data-driven multiobjective optimization problems.
- We will also see talks about the latest advancements in the graphical user interface being developed for DESDEO, and other novel applications utilizing DESDEO.
- For news and resources related to desdeo, visit: https://desdeo.it.jyu.fi/

Bibliography I

[1] G. Misitano, B. S. Saini, B. Afsar, B. Shavazipour, and K. Miettinen. "DESDEO: The Modular and Open Source Framework for Interactive Multiobjective Optimization". *IEEE Access* 9 (2021), pp. 148277–148295. DOI: 10.1109/ACCESS.2021.3123825.

[2] Kaisa Miettinen and Marko M. Mäkelä. "Synchronous approach in interactive multiobjective optimization". *European Journal of Operational Research* 170.3 (2006), pp. 909–922. DOI: 10.1016/j.ejor.2004.07.052.

[3] Ana B. Ruiz, Karthik Sindhya, Kaisa Miettinen, Francisco Ruiz, and Mariano Luque. "E-NAUTILUS: A decision support system for complex multiobjective optimization problems based on the NAUTILUS method". *European Journal of Operational Research* 246.1 (2015), pp. 218–231. DOI: 10.1016/j.ejor.2015.04.027.

[4] Ana B. Ruiz, Francisco Ruiz, Kaisa Miettinen, Laura Delgado-Antequera, and Vesa Ojalehto. "NAUTILUS Navigator: free search interactive multiobjective optimization without trading-off". *Journal of Global Optimization* 74.2 (2019), pp. 213–231. DOI: 10.1007/s10898-019-00765-2. URL: https://doi.org/10.1007/s10898-019-00765-2.

[5] Andrzej P Wierzbicki. "A mathematical basis for satisficing decision making". *Mathematical Modelling* 3 (1982), pp. 391–405.

[6] Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. "A preference-based evolutionary algorithm for multi-objective optimization". *Evolutionary computation* 17.3 (2009), pp. 411–436.

[7] Bhupinder Singh Saini, Michael Emmerich, Atanu Mazumdar, Bekir Afsar, Babooshka Shavazipour, and Kaisa Miettinen. "Optimistic NAUTILUS navigator for multiobjective optimization with costly function evaluations". *Journal of Global Optimization* (2022), pp. 1–25.

[8] Ran Cheng, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. "A Reference Vector Guided Evolutionary Algorithm for Many-Objective Optimization". *IEEE Transactions on Evolutionary Computation* 20.5 (2016), pp. 773–791. DOI: 10.1109/TEVC.2016.2519378.

[9] Kalyanmoy Deb and Himanshu Jain. "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints". *IEEE Transactions on Evolutionary Computation* 18.4 (2014), pp. 577–601. DOI: 10.1109/TEVC.2013.2281535.

Bibliography IV

[10] Bhupinder Singh Saini, Jussi Hakanen, and Kaisa Miettinen. “A New Paradigm in Interactive Evolutionary Multiobjective Optimization”. *Parallel Problem Solving from Nature – PPSN XVI*. Ed. by Thomas Bäck, Mike Preuss, André Deutz, Hao Wang, Carola Doerr, Michael Emmerich, and Heike Trautmann. Cham: Springer International Publishing, 2020, pp. 243–256.

[11] Jussi Hakanen, Tinkle Chugh, Karthik Sindhya, Yaochu Jin, and Kaisa Miettinen. “Connections of reference vectors and different types of preference information in interactive multiobjective evolutionary algorithms”. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. Athens, Greece: IEEE, 2016, pp. 1–8. DOI: 10.1109/SSCI.2016.7850220.