

DEMO Tutorial: DESDEO – From Multiobjective Optimization Problems to Solutions

Giovanni Misitano

`giovanni.a.misitano@jyu.fi`

University of Jyväskylä
The multiobjective optimization group

December 1, 2021



JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ



- 1 Introduction
- 2 Defining Multiobjective Optimization Problems
- 3 Solving Multiobjective Optimization Problems
 - Scalarization
 - Evolving Populations
- 4 Interacting With Interactive Methods
- 5 Summary

- 1 Introduction
- 2 Defining Multiobjective Optimization Problems
- 3 Solving Multiobjective Optimization Problems
- 4 Interacting With Interactive Methods
- 5 Summary

This will be an example driven tutorial on the basic usage of DESDEO. As the name of the tutorial implies, we will cover how to define multiobjective optimization problems, how to solve them, and how to interact with interactive methods. The examples shown during this tutorial are available online¹.

¹https://github.com/industrial-optimization-group/desdeo-tutorials/blob/78a0d4ca2bb443531fc7d91745500237e4a7a86a/DEMO_tutorial_01122021/Tutorial.ipynb  4/18

Defining Multiobjective Optimization Problems

- 1 Introduction
- 2 Defining Multiobjective Optimization Problems**
- 3 Solving Multiobjective Optimization Problems
- 4 Interacting With Interactive Methods
- 5 Summary

Multiobjective optimization problem

A multiobjective optimization problem can be defined as

$$\min F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})), \quad (1)$$

where $f_1 \dots f_i$, $i \in [1, k]$ are objective functions and \mathbf{x} is a decision variable vector. The vectors \mathbf{x} can be subject to both **box-constraints** and **function constraints**. Feasible \mathbf{x} belong to the *feasible variable space* S or $\mathbf{x} \in S$.

Box-constraints

$$x_i^{\text{low}} \leq x_i \leq x_i^{\text{high}}, x_i \in \mathbf{x} \quad (2)$$

Function constraints

$$\begin{aligned} g(\mathbf{x}) - \delta_g &> 0 \\ h(\mathbf{x}) - \delta_h &= 0 \\ \delta_g, \delta_h &\in \mathbb{R} \end{aligned} \quad (3)$$

- In (2) x_i^{low} and x_i^{high} are the lower and higher limits for the i th element in \mathbf{x} , respectively.
- In (3) δ_g and δ_h are scalar values which should be exceeded or be exactly matched by $g(\mathbf{x})$ and $h(\mathbf{x})$, respectively.

Notebook example

Solving Multiobjective Optimization Problems

- 1 Introduction
- 2 Defining Multiobjective Optimization Problems
- 3 Solving Multiobjective Optimization Problems**
 - Scalarization
 - Evolving Populations
- 4 Interacting With Interactive Methods
- 5 Summary

Scalarization function

$$s(F(\mathbf{x}); \mathbf{p}) : \mathbb{R}^k \rightarrow \mathbb{R} \quad (4)$$

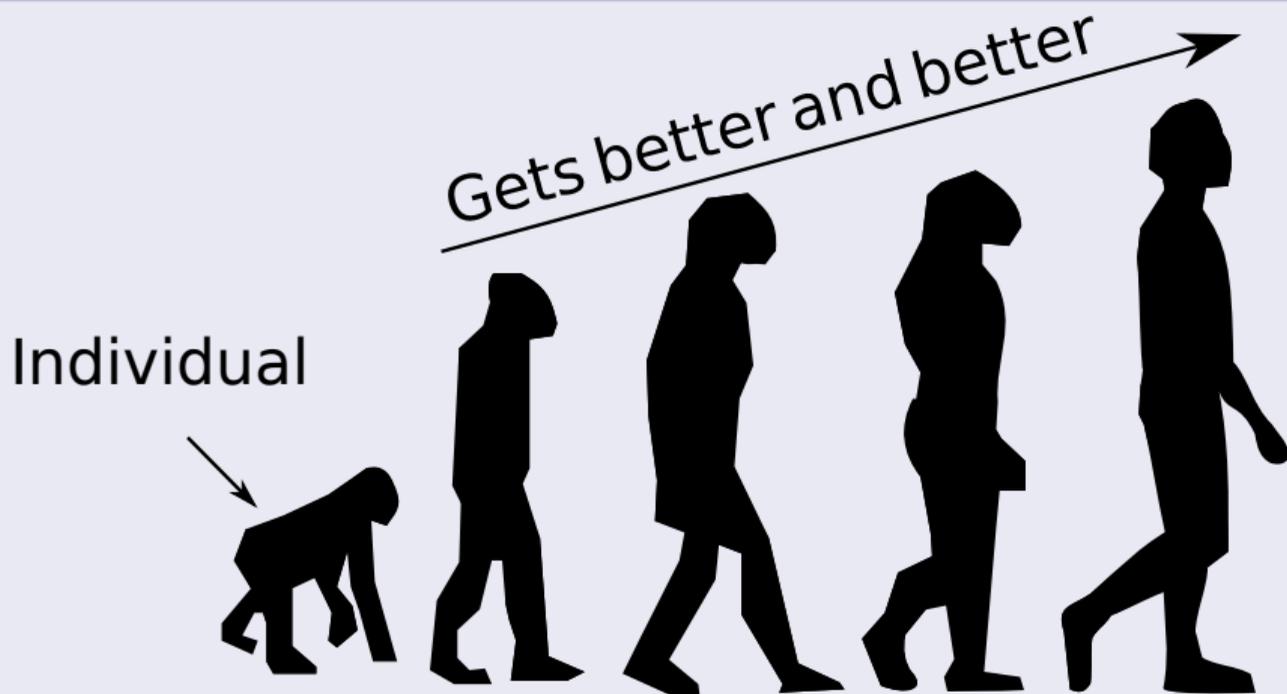
Scalarized problem

$$\begin{aligned} \min s(F(\mathbf{x}); \mathbf{p}) \\ \text{s.t. } \mathbf{x} \in S \end{aligned} \quad (5)$$

- In (4) and (5), \mathbf{p} is a set of additional parameters passed to the scalarization function. I.e., **preferences**.

Notebook example

Evolution



Notebook example

Interacting With Interactive Methods

- 1 Introduction
- 2 Defining Multiobjective Optimization Problems
- 3 Solving Multiobjective Optimization Problems
- 4 Interacting With Interactive Methods**
- 5 Summary

`request`

Returned by interactive methods when their `iterate()` or `start()` methods are invoked. It is a Python dictionary with two important entries: `content` and `response`.

`request.content`

A Python dictionary with data and information necessary to continue iterating an interactive method. Has at least the entry `message`.

`request.response`

An often empty Python dictionary. Used to define a response to the `request` so that iteration of an interactive method can be continued.

Notebook example

- 1 Introduction
- 2 Defining Multiobjective Optimization Problems
- 3 Solving Multiobjective Optimization Problems
- 4 Interacting With Interactive Methods
- 5 Summary**

Through the examples, we have seen how to define, scalarize, solve and evolve problems. We also saw the basic idea of interactive with interactive multiobjective optimization methods. These same ideas have also been showcased in our recently published DESDEO article [1].

- [1] G. Mispitano 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.