JYVÄSKYLÄN YLIOPISTO
UNIVERSITY OF JYVÄSKYLÄ

DESDEO 2.0: The New and Improved Open-source Framework for Interactive Multiobjective Optimization

Giovanni Misitano[1], Bhupinder Singh Saini[1], Juho Roponen[1], Juuso Pajasmaa[1], Giomara Lárraga[1], Babooshka Shavazipour[1], Kaisa Miettinen[2]

25.6.2025 EURO 2025 LEEDS GIOVANNI MISITANO (GIOVANNI.A.MISITANO@JYU.FI)

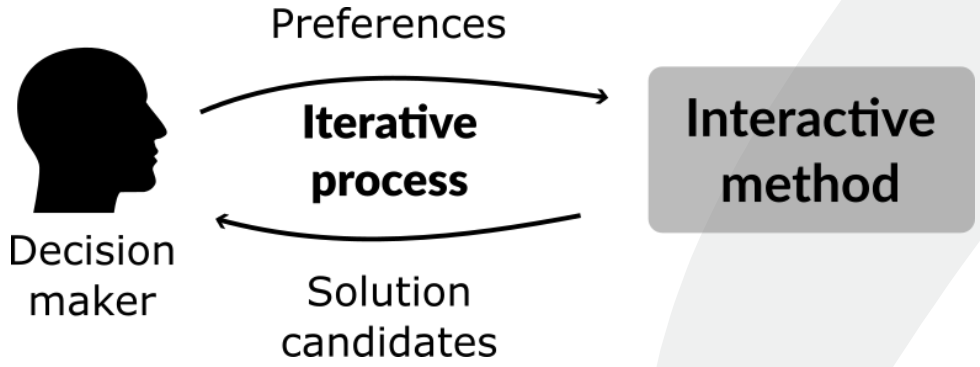
What is DESDEO?

- DESDEO is an **open-source** framework for **interactive multiobjective optimization**.
- Its main principles include **modularity, extensibility, and practical applicability**.
- Intended users of the framework are **researchers, students, and practitioners**.
- Developed mainly at the University of Jyväskylä in the Multiobjective Optimization Group.
- DESDEO development started in 2015 (**a decade ago!**).
  - Development of interactive software goes much farther back.

Misitano, G., Saini, B. S., Afsar, B., Shavazipour, B., & Miettinen, K. (2021). DESDEO: The modular and open source framework for interactive multiobjective optimization. *IEEE Access*, *9*, 148277-148295.

Why interactive multiobjective optimization?

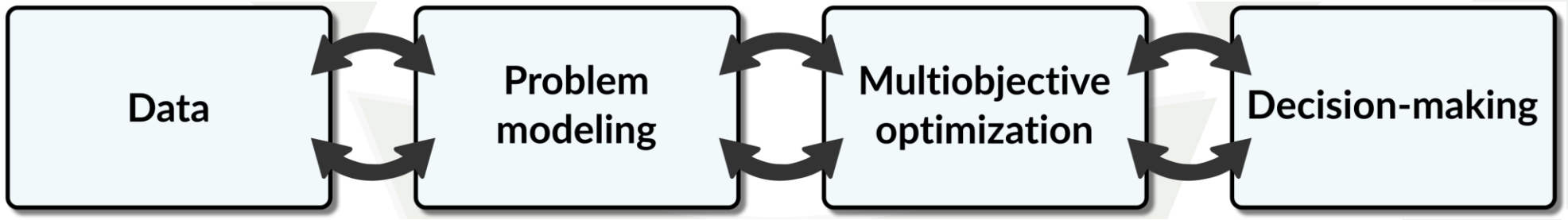
- Real-world problems involve **trade-offs between conflicting objectives**.
- Decision makers have the responsibility to choose the **best Pareto optimal solution**.
- By **gradually and iteratively** incorporating preferences, interactive methods allow decision makers to **explore and learn** about the problem being solved.
- Interactive methods are **human-centric** by nature.



Miettinen, K., Nonlinear Multiobjective Optimization (1999), Kluwer Academic Publishers, Boston.

Why software for interactive methods is needed?

- Because of its human-centric nature, interactive methods are **not just algorithms.**
- There is a lot of information that needs to be conveyed to decision makers to support them in decision-making. **Not just text on a screen.**
- They are means for a decision-support process, which requires interactive **visualizations**, user **interfaces**, and **services to enable interaction**.
- Interactive methods cannot live in a vacuum **but must be applied and studied in a holistic way**.
- This poses **significant challenges** when developing interactive methods: **method implementations need to be available!**

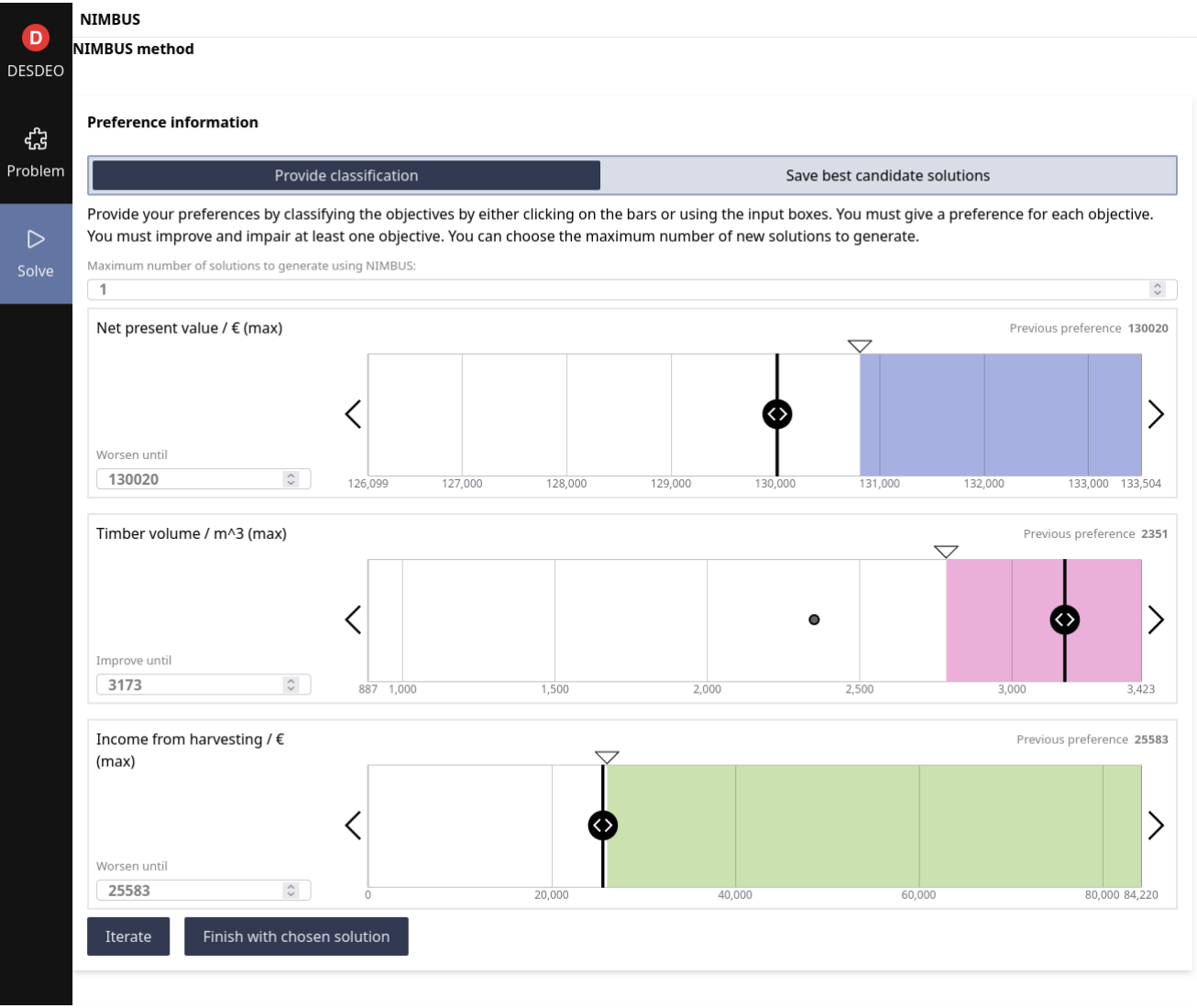


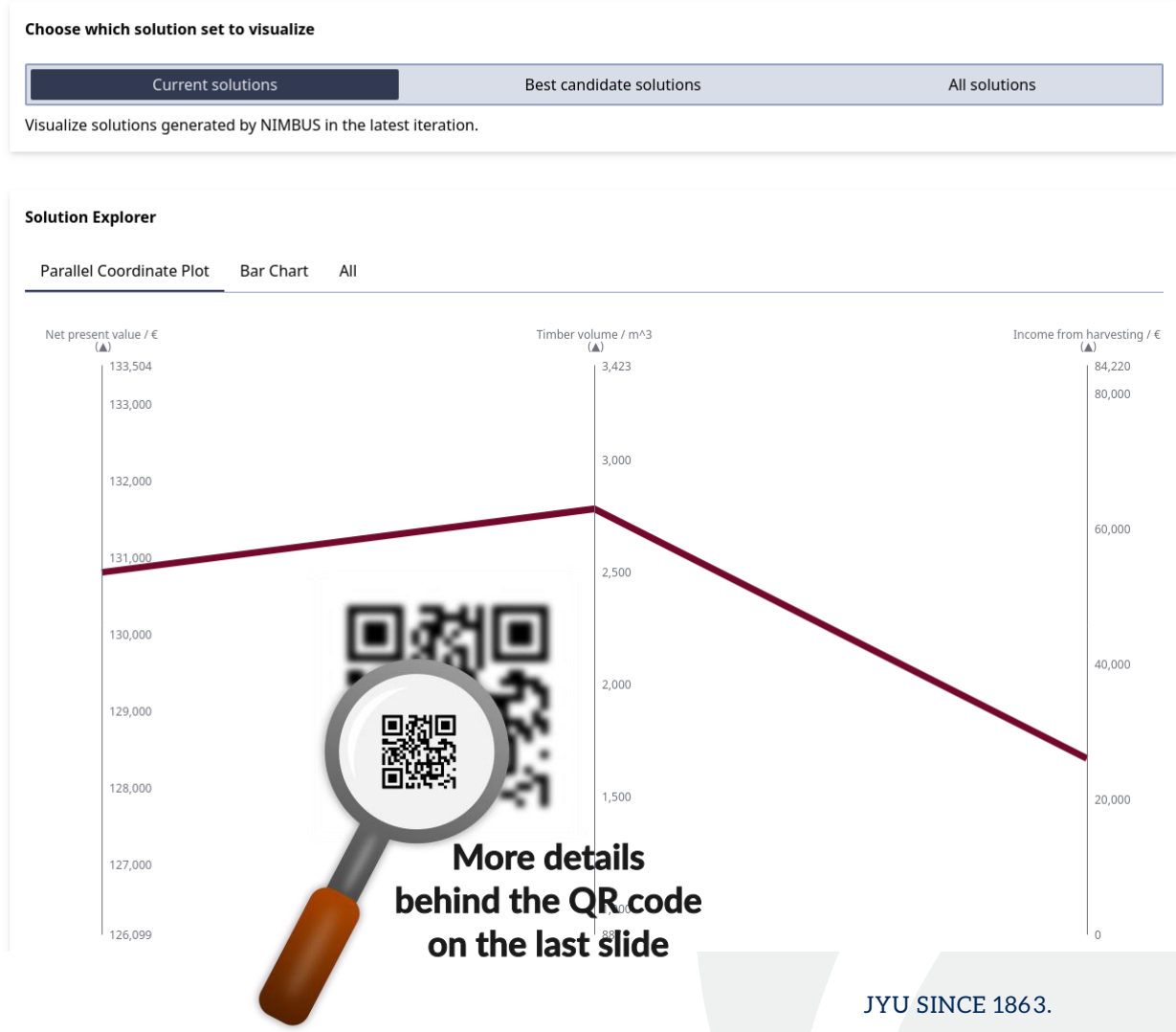
Afsar, B., Miettinen, K., & Ruiz, F. (2021). Assessing the performance of interactive multiobjective optimization methods: A survey. *ACM Computing Surveys (CSUR)*, *54*(4), 1-27.

Why DESDEO 2.0?

- Because the previous version of **DESDEO could not address all the needs** of researchers and decision makers.
  - Feature creep, issues rooted in fundamental design choices, lessons learned…
- DESDEO has been **completely restructured** and designed to address the **needs emerging from applying interactive methods in real-life in version 2.0.**
- Many of the choices made in the restructuring are motivated by our refined understanding of the **needs and requirements of researchers, practitioners (analysts), and decision makers**.
- The restructuring is a culmination of **years of experience**.
- Despite its shortcomings, the previous version of DESDEO has been used in a variety of research and teaching.

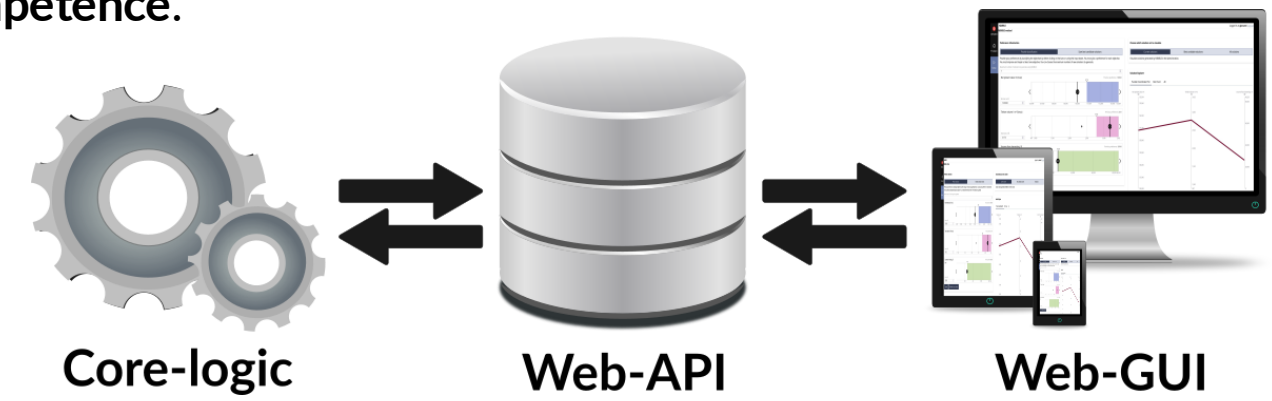
Goal: Accessible interfaces for interactive methods





DESDEO 2.0's three main components

- **The core-logic** is implemented in Python and contains **means to model multiobjective optimization problems** and **solve them using interactive methods**.
- **The web-API exposes the functionalities of the core-logic**, allowing external application to interface into DESDEO, and utilize it. And it provides the needed services to build applications for interactive multiobjective optimization.
- **The web-GUI provides a web-based graphical user interface** for modeling and solving multiobjective optimization problems using interactive methods. It also provides a multitude of interactive visualizations and components.
- DESDEO 2.0 has been designed as a **full-stack application**. These systems are rare in out field, and they **require versatile competence**.



The core-logic

- Designed from the beginning to **meet the needs of the web-API and web-GUI**.
- **JSON-based problem modeling language**, which supports many kinds of problems, including data- and simulation-based problems.
- **A problem needs to be modeled only once**, after which it can be solved using any suitable method available in DESDEO.
- Both **scalarization (MCDM) and population (EMO) -based** interactive methods are available.
- Interactive methods have been implemented so that **reusing their components is easy**.
- We provide many **interfaces to existing optimization algorithms**. We do not wish to reinvent the wheel.

Problem modeling

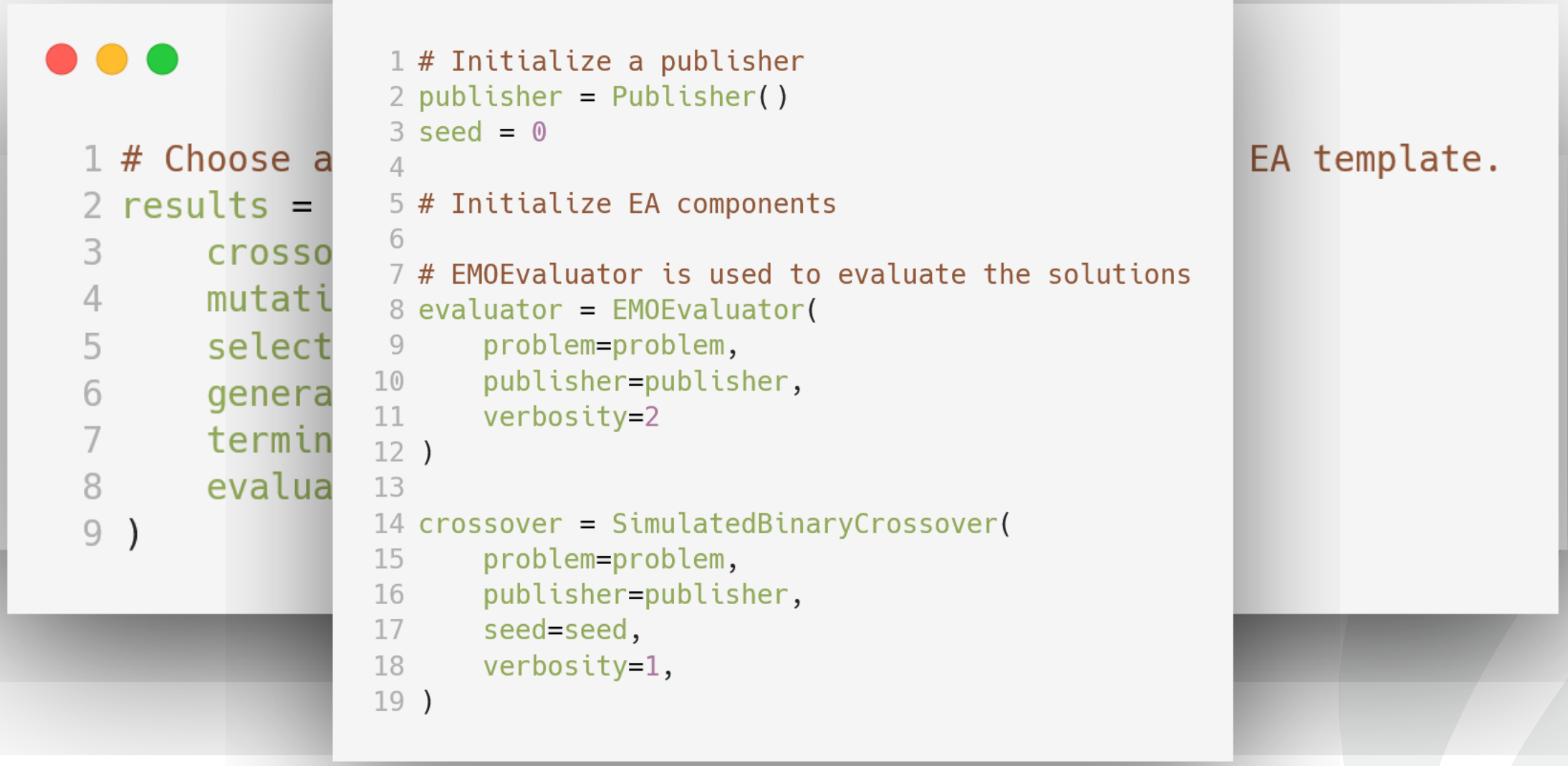
```
from desdeo.problem import P

vehicle_design_problem = Pro
    name="Vehicle design pro
    description="Vehicle des
constraints.",
    variables=[x_1, x_2, y],
    constants=[ideal_n_cylin
    objectives=[cost, emissi
    constraints=[fuel_effici
)
```

```
{
  "name": "objective example",
  "symbol": "f",
  "unit": null,
  "func": [
    "Add",
    "x",
    1
  ],
  "simulator_path": null,
  "surrogates": null,
  "maximize": false,
  "ideal": null,
  "nadir": null,
  "objective_type": "analytical",
  "is_linear": true,
  "is_convex": false,
  "is_twice_differentiable": false,
  "scenario_keys": null
}
```

Interactive methods

- MCDM methods are implemented in a **functional fashion**.
  - For each method, multiple functions are defined, which fulfill its various steps.
  - These functions are independent and can be combined freely.
- EMO methods are implemented based on **EMO components**:
  - Components follow a publish-subscribe pattern, which allows combining them in a multitude of ways.
  - Some templates are also available, which define a receive on how evolutionary components should interact, leaving the user the choice of choosing the specific components.
- Hybridization of methods is readily possible. **Combine MCDM and EMO methods** in a modular fashion!

Interactive methods



```
# Choose a
results =
    crosso
    mutati
    select
    genera
    termin
    evalua
)
```

```
# Initialize a publisher
publisher = Publisher()
seed = 0

# Initialize EA components

# EMOEvaluator is used to evaluate the solutions
evaluator = EMOEvaluator(
    problem=problem,
    publisher=publisher,
    verbosity=2
)

crossover = SimulatedBinaryCrossover(
    problem=problem,
    publisher=publisher,
    seed=seed,
    verbosity=1,
)
```

```
EA template.
```

Other features of the core-logic

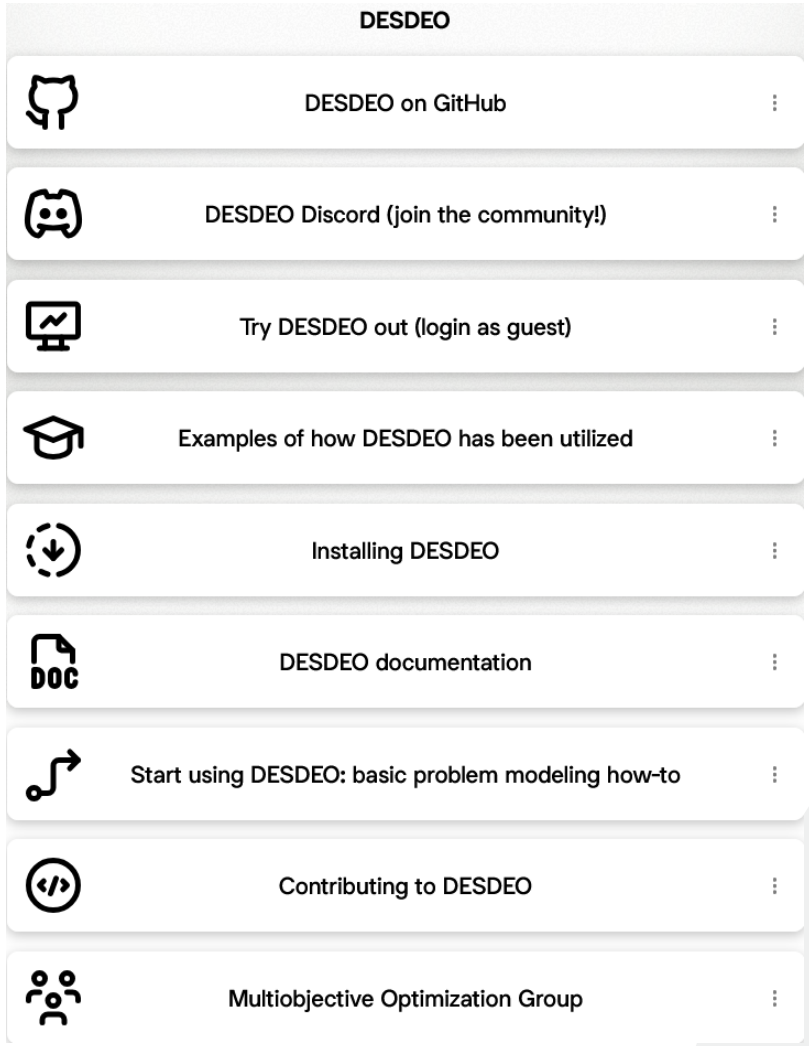
- The **core-logic is a fully independent** component of DESDEO, and can be utilized like any Python library, e.g., in a **Jupyter notebook**.
- DESDEO enables comparison of methods, switching between methods, changing the preference information style, **all in one environment**.
- Novelties! Interactive **group decision-making methods, explainability, agents, artificial decision makers**...
- **Model once, solve multiple times: once you have a Problem modeled, you can start solving and manipulating it to your liking.**

Conclusions

- DESDEO 2.0 is currently **the only open-source project specializing in both MCDM and EMO interactive multiobjective optimization methods**.
- DESDEO 2.0 is designed from the ground-up to **address the needs of an actual decision-support application**. We are aware that interactive methods are much more than just an algorithm.
- DESDEO 2.0 **is for researchers, practitioners, and students.** The web-GUI offers and will offer a way for anybody to use interactive methods through their web browser.
- Do not like our GUI? **Implement your own GUI** or integrate DESDEO into your existing application using the web-API.
- We want to build an active community around DESDEO. Anybody is welcome to contribute. **Not just academy!**

Q&A

https://linktr.ee/gialmisi



Research Council of Finland

This work is supported by the DESIDES project (grant number 355346) funded by the Research Council of Finland. This work is part of the thematic research area Decision Analytics Utilizing Causal Models and Multiobjective Optimization (DEMO, jyu.fi/demo) at the University of Jyväskylä.