Interactive tool for visualizing the comprehensive performance of evolutionary multi-objective algorithms applied to problems with two or three objectives

Michał Tomczyk **Miłosz Kadziński**

Institute of Computing Science
Poznan University of Technology, Poland

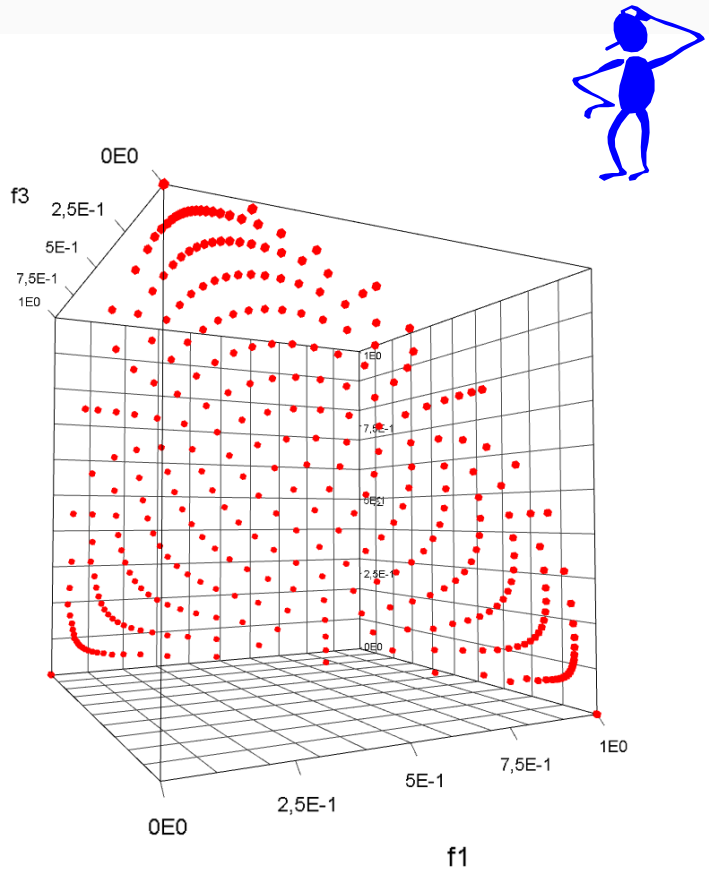
michal.tomczyk@cs.put.poznan.pl
www.cs.put.poznan.pl/mtomczyk/

Introduction

Visualization is crucial in Evolutionary Multi-objective Optimization (EMO) due to the inherent complexity of dealing with conflicting objectives.

However, the visualization can be deemed underutilized in this research field. Existing papers often just:

- show just a final population generated (no progression),
- this population is often derived from one execution of the method (arbitrary selection),
- the visualization often concerns just the population (limited scope).



Introduction

In this study, we try to simultaneously address all these three issues:

- show just a final population generated (no progression),
- ✓ all solutions generated throughout optimization can contribute to the final result (comprehensiveness)

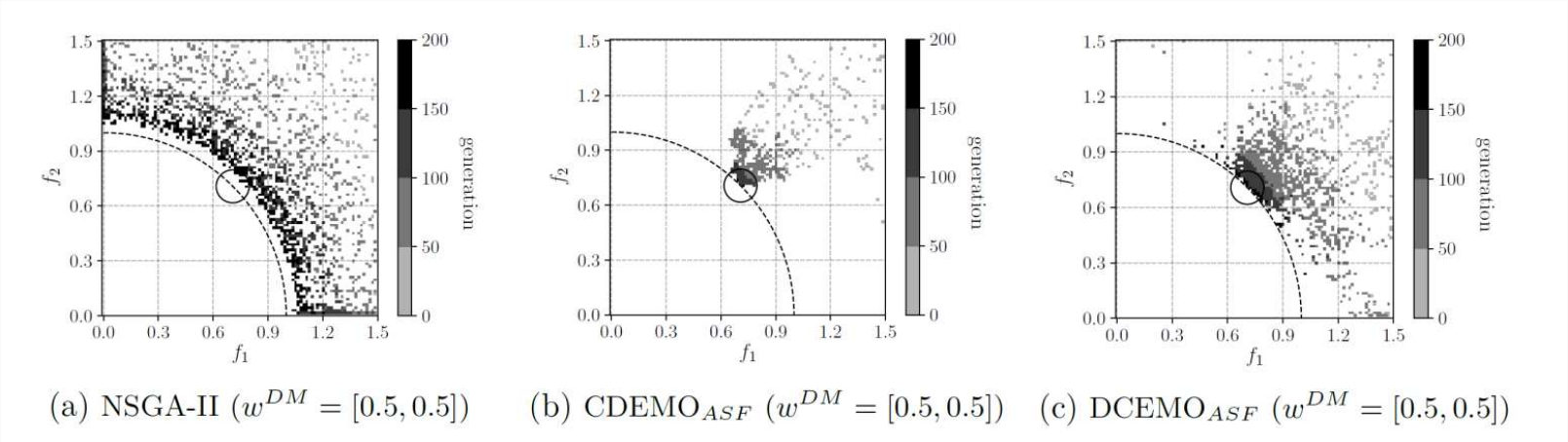
- this population is often derived from one execution of the method (arbitrary selection),
- ✓ the result is a product of many independent executions of the method (robustness)

- the visualization often concerns just the population (limited scope).
- ✓ the visualization can be easily adjusted to exhibit the method's performance from different views (versatility)

Comprehensive, robust, flexible visualization

This paper is rooted in one of our previous works, in which we presented **the ViPEMO technique primarily designed to illustrate the robust performance of interactive Methods when run for problems with two objectives**.

In this paper, however, we generalize this concept, provide a more detailed algorithmic background, discuss it from the algorithmic view, **and design a ready-to-use visualization tool**.

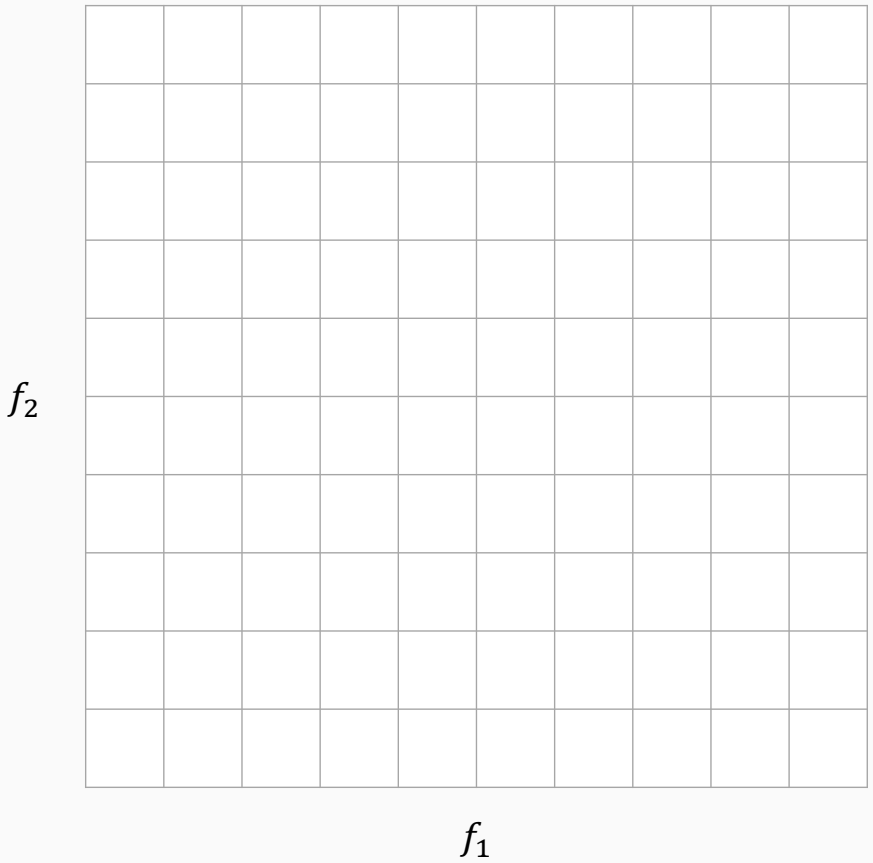


(a) NSGA-II ($w^{DM} = [0.5, 0.5]$) (b) CDEMO$_{ASF}$ ($w^{DM} = [0.5, 0.5]$) (c) DCEMO$_{ASF}$ ($w^{DM} = [0.5, 0.5]$)

M. Kadziński, M. Tomczyk, R. Słowiński, Interactive Cone Contraction for Evolutionary Mutliple Objective Optimization , Advances in Data Analysis with Computational Intelligence Methods, 293-309, 2018

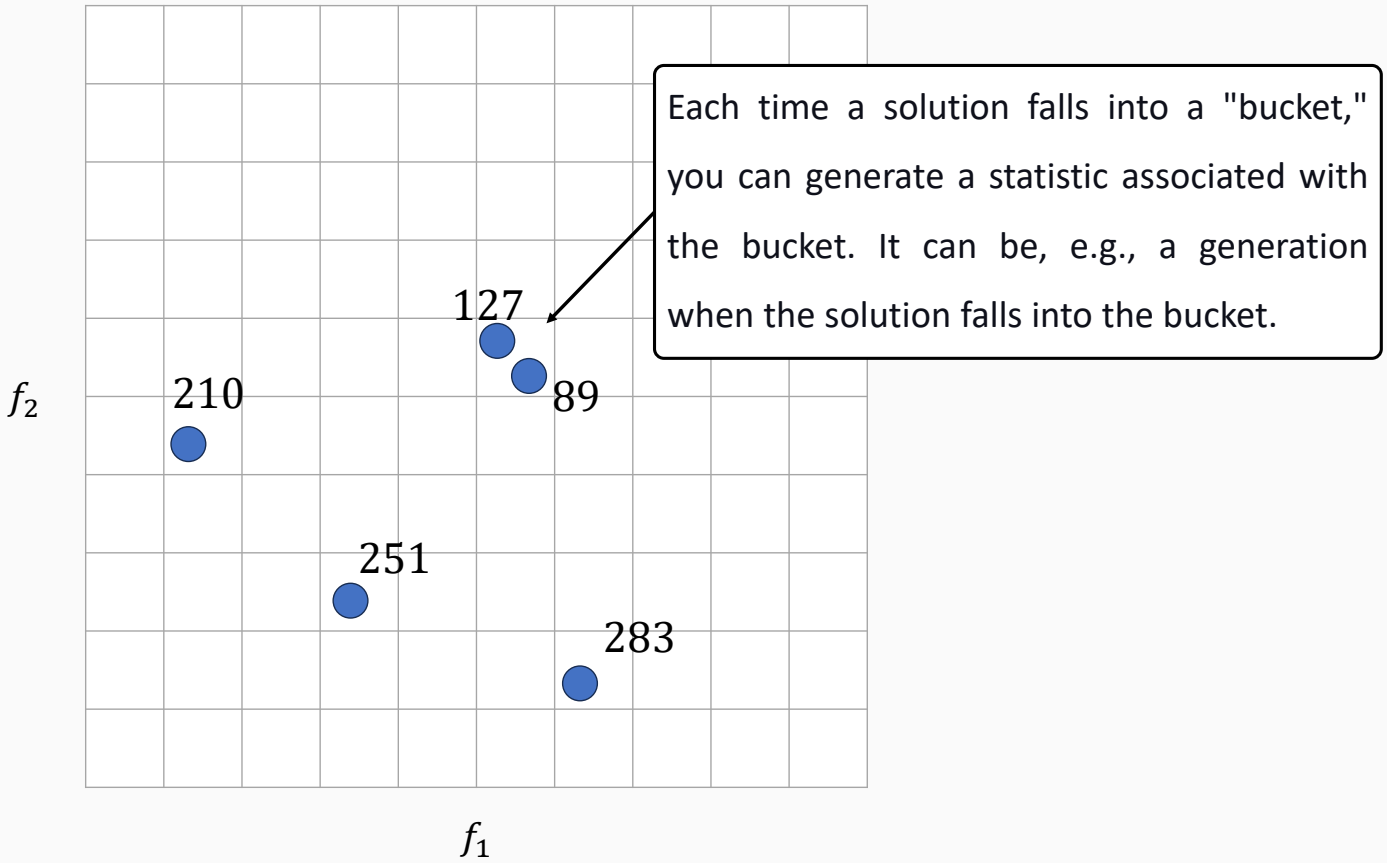
Comprehensive, robust, flexible visualization

Consider a discretized bounded region of the objective space.



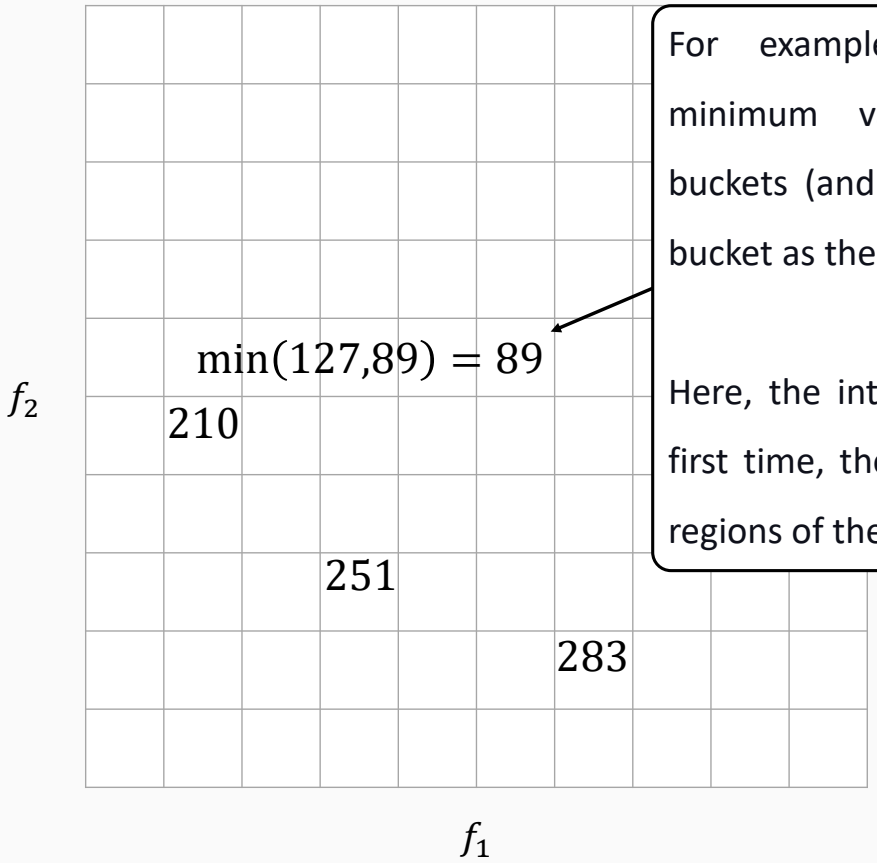
Comprehensive, robust, flexible visualization

You can run the method (one time) and trace the generated solutions.



Comprehensive, robust, flexible visualization

After the (trial) run is completed, you can construct its summary.

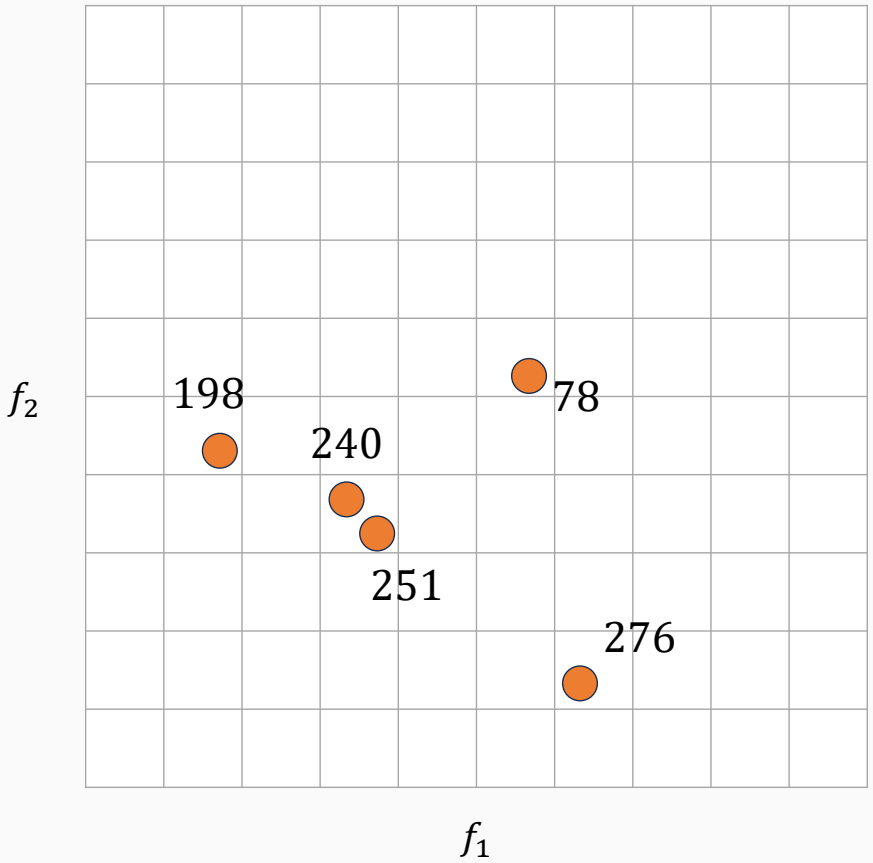


For example, you can calculate the minimum values associated with the buckets (and associate the result with the bucket as the trail summary).

Here, the interpretation is: "when, for the first time, the method discovered different regions of the objective space".

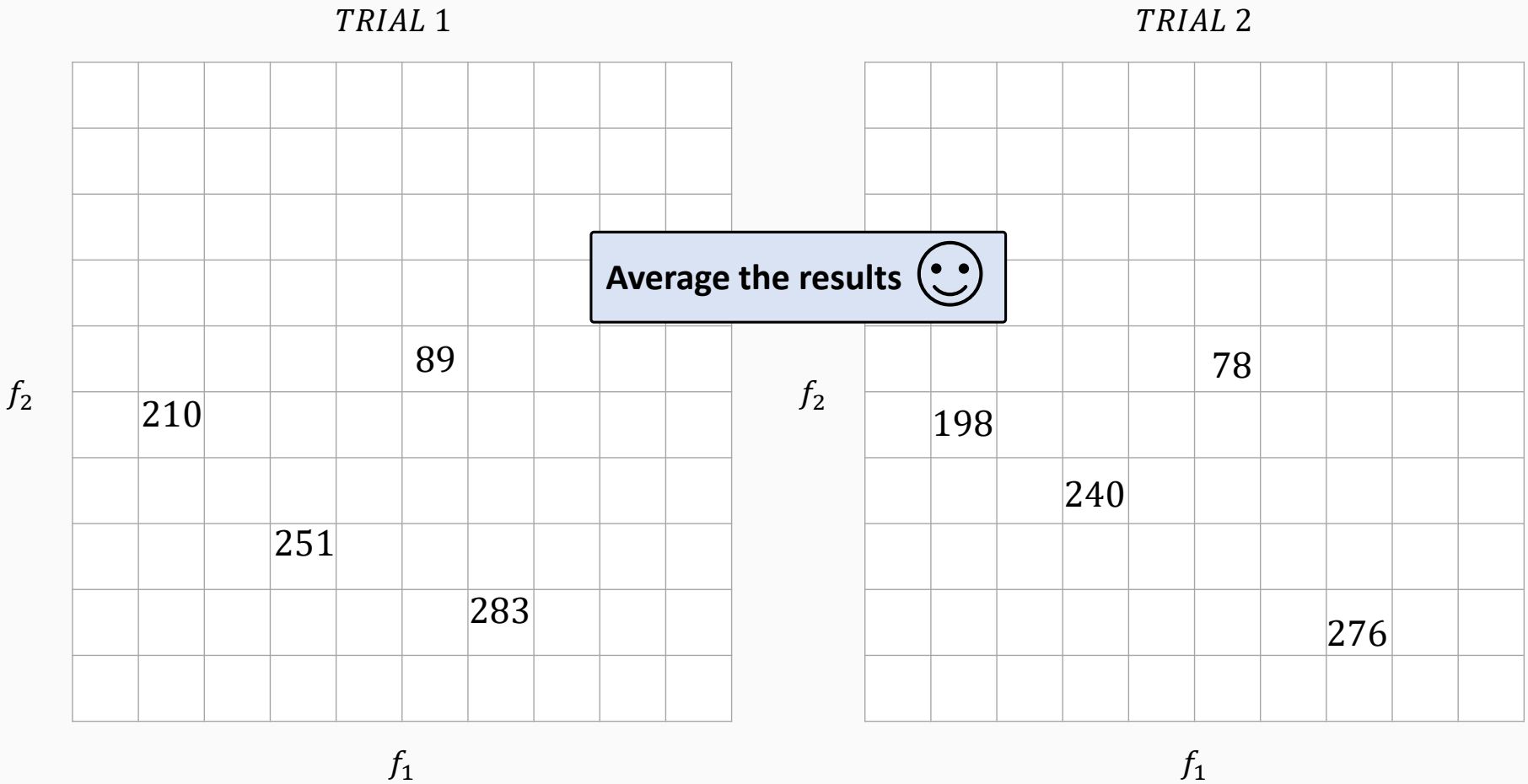
Comprehensive, robust, flexible visualization

You can execute the method, another time, producing another trial summary.

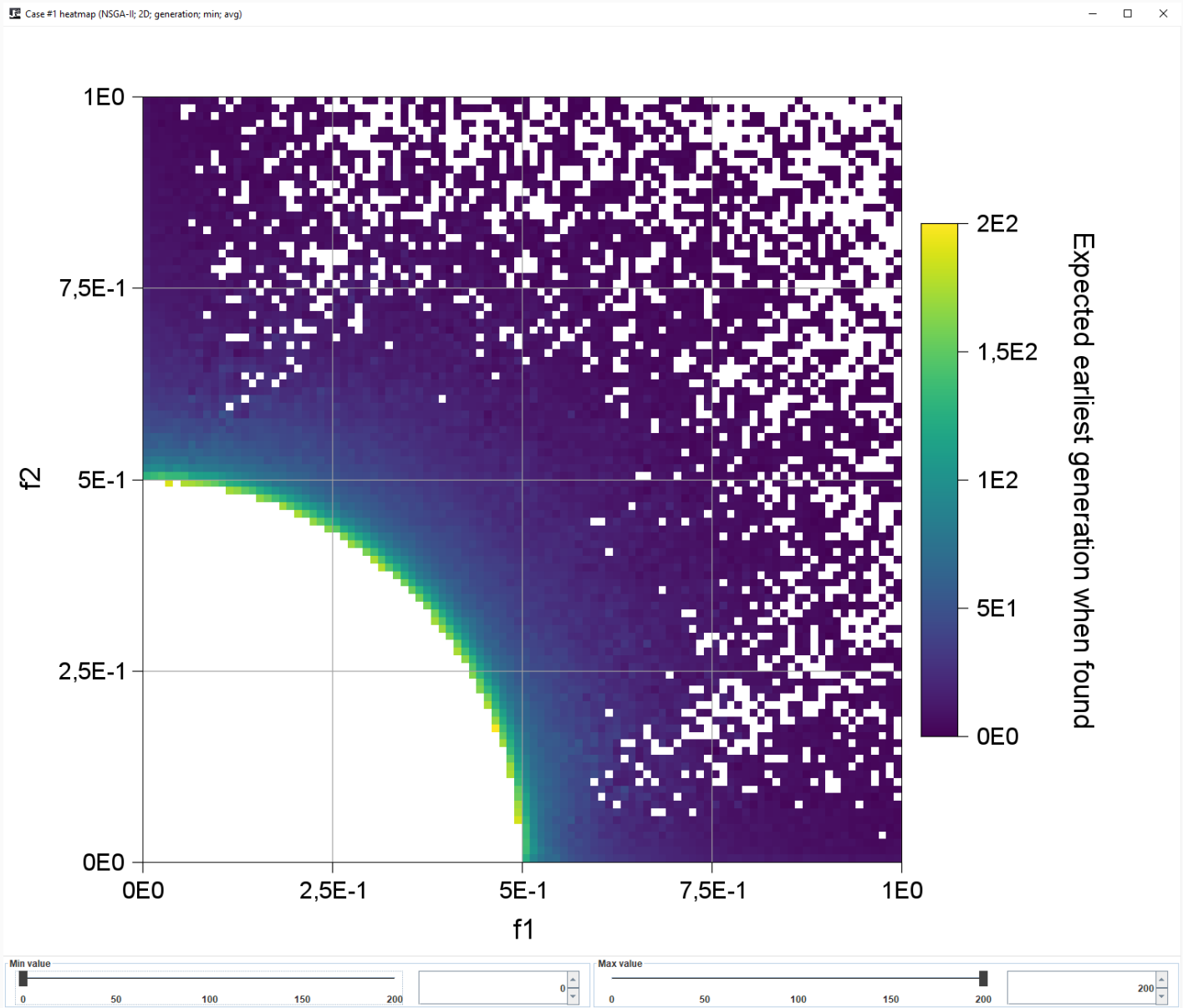


Comprehensive, robust, flexible visualization

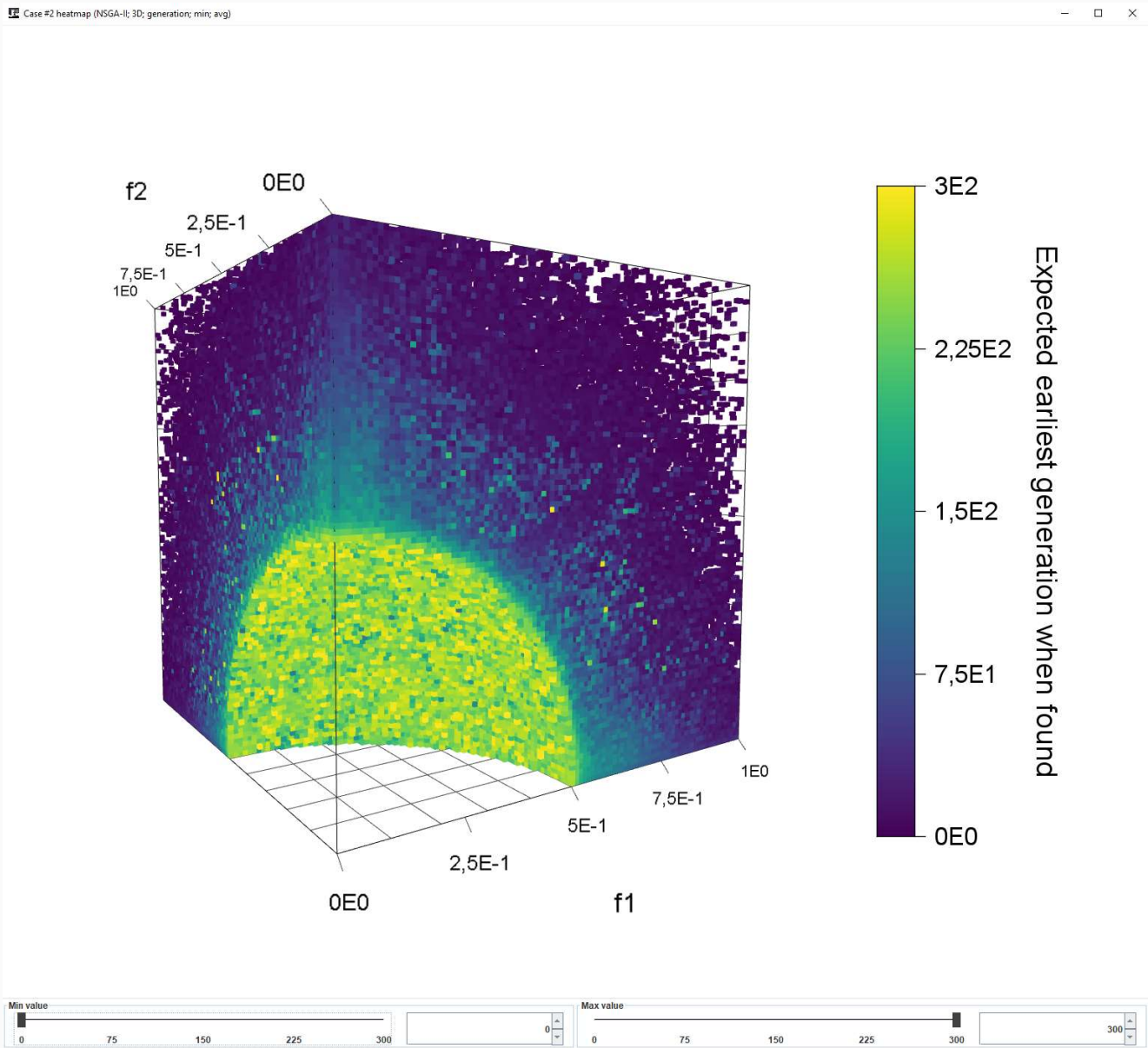
Finally, you can calculate a final statistics (e.g., average) from trial summaries stored in the buckets, providing some comprehensive matrix-like representation of the result, which can be depicted using a heatmap.



Comprehensive, robust, flexible visualization



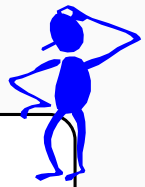
Comprehensive, robust, flexible visualization



Comprehensive, robust, flexible visualization

- ✓ All solutions generated throughout optimization can contribute to the final result (comprehensiveness)
- ✓ The result is a product of many independent executions of the method (robustness)
- ✓ the visualization can be easily adjusted to exhibit the method's performance from different views (versatility):
  1. Bottom: solution-level statistic (per one method, per one run)
  2. Middle: trial-level statistic (per one method, per one run)
  3. Top: all-trials-level statistic (per many runs)

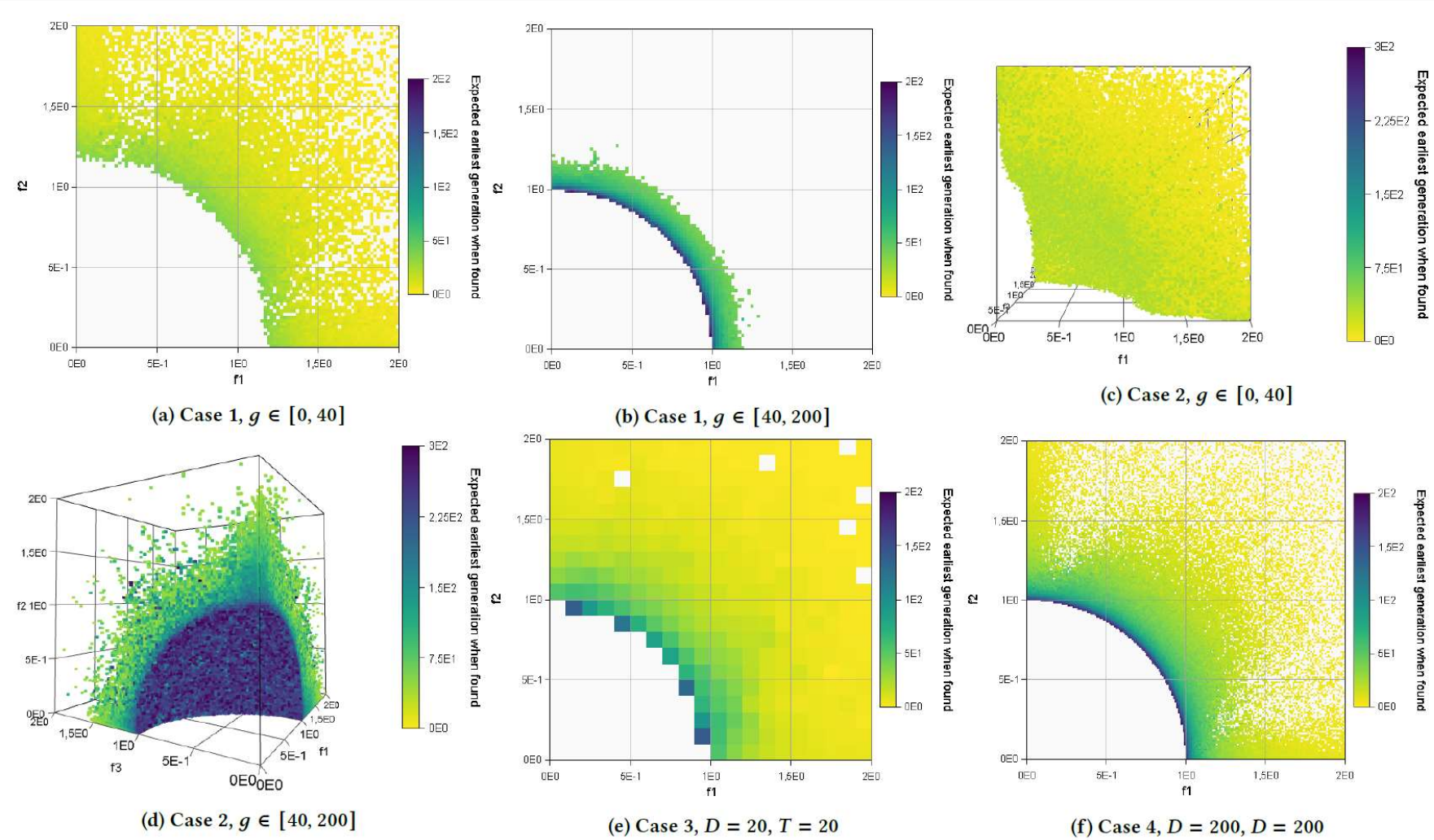
Comprehensive, robust, flexible visualization



In this paper, we focused on the development. We designed a visualization tool that can support researchers in data presentation and analysis. Its main features are:

- It is written in Java, meaning it **should be executable on any standard machine**.
- It is devoted to 2D and 3D visualization.
- The tool is interactive, e.g., the user can **freely manipulate the camera in 3D mode**.
- We focused on high efficiency (e.g., for the 3D case, the rendering is done on the GPU → even for the 100^3 discretization level, the FPS can be counted in hundreds on an average-class GPU).
- The tool allows for **efficient data filtering using sliders**.
- The input data (around several KB) is separated from the tool (around 10MB)
- Time input data consists of the XML file that performs the top-level customization and the binary file that contains the bucket data (the file specification is described in the paper).

Comprehensive, robust, flexible visualization



(a) Case 1, $g \in [0, 40]$

(b) Case 1, $g \in [40, 200]$

(c) Case 2, $g \in [0, 40]$

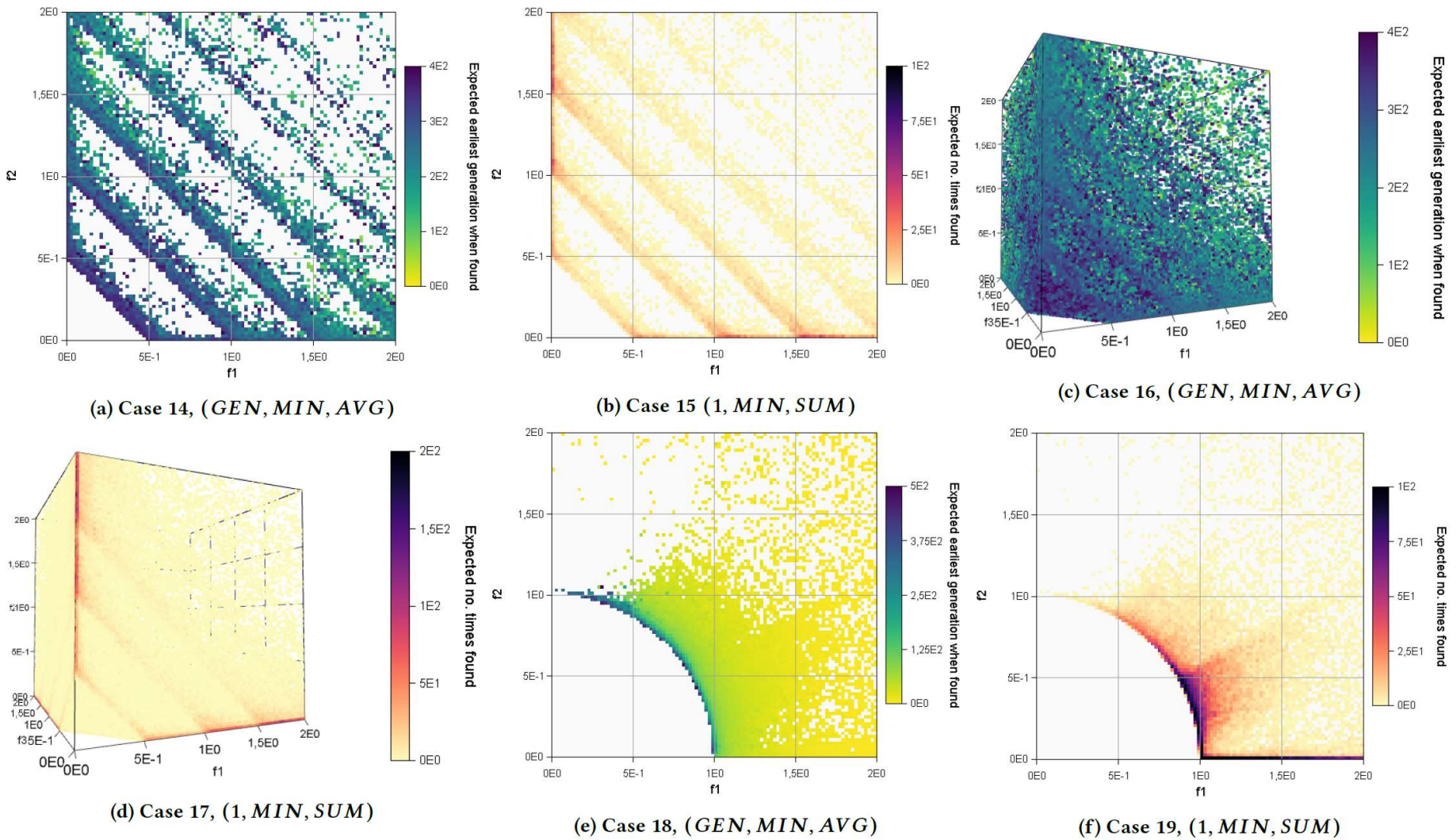
(d) Case 2, $g \in [40, 200]$

(e) Case 3, $D = 20$, $T = 20$

(f) Case 4, $D = 200$, $D = 200$

Figure 2: Screenshots of the visualization tool presenting results for test cases 1–4

Comprehensive, robust, flexible visualization



(a) Case 14, (GEN, MIN, AVG)

(b) Case 15 $(1, MIN, SUM)$

(c) Case 16, (GEN, MIN, AVG)

(d) Case 17, $(1, MIN, SUM)$

(e) Case 18, (GEN, MIN, AVG)

(f) Case 19, $(1, MIN, SUM)$

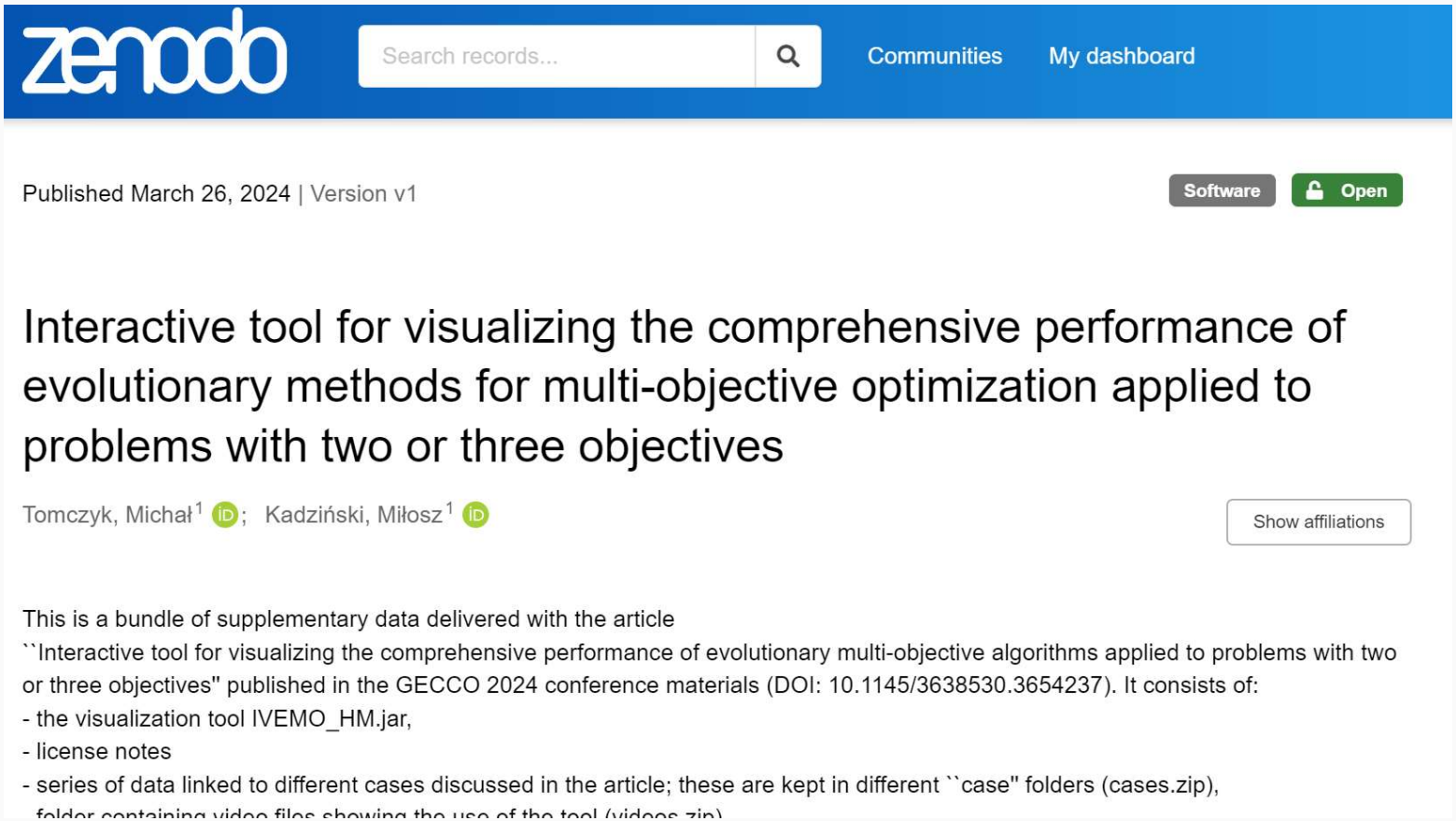
Figure 5: Screenshots of the visualization tool presenting results for test cases 14–19

Presentation of the tool

Time for running the tool 

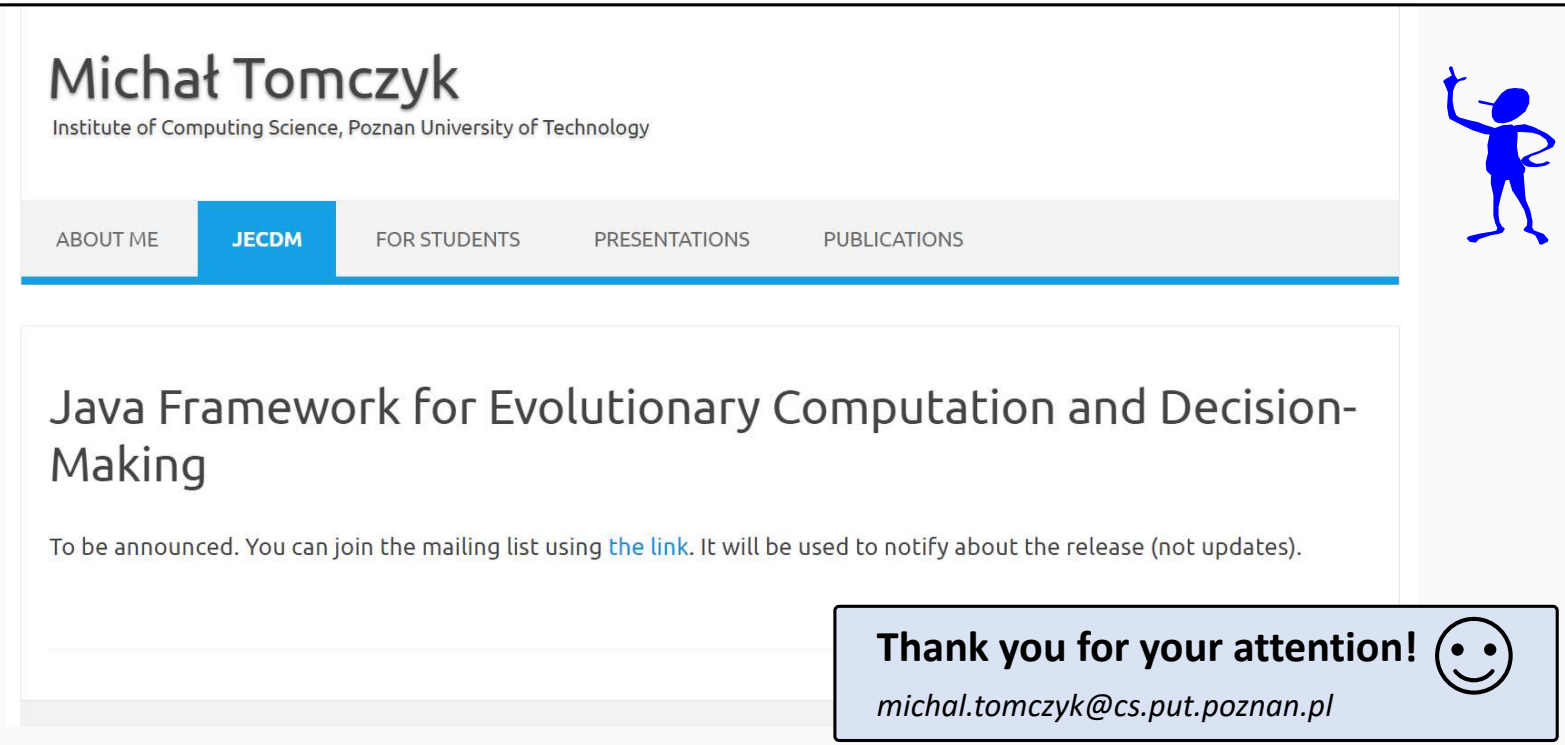
Zenodo repository

- The tool (JAR file) can be downloaded from my Zenodo repository: https://zenodo.org/records/10877574.
- The repository contains input data for the several use cases presented in the paper.



JECDM framework

Although the application is the result of this study (you have to generate the input data on your own), the codes for generating the input data will soon be available as part of the Java framework for Evolutionary Computation and Decision Making (JECDM) that is currently under development (released in next 3-5 months). See my website: https://www.cs.put.poznan.pl/mtomczyk/index.php/jecdm



Thank you for your attention!

michal.tomczyk@cs.put.poznan.pl