Advanced Optimization

Lecture/Exercise 5: Critically Looking at Data

December 18, 2019

Master AIC

Université Paris-Saclay, Orsay, France

Anne Auger

INRIA Saclay – Ile-de-France

INVENTORS FOR THE DIGITAL WORLD

Dimo Brockhoff

INRIA Saclay – Ile-de-France

Assignment of Papers

2) RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. Gaetano, Francesco

3) A universal catalyst for first-order optimization. Simon, Wafa

5) Efficient optimization of many objectives by approximation-guided evolution. Gérémy

6) A Mean-Variance Optimization Algorithm. Ramine, Gaspard

8) Population Size Adaptation for the CMA-ES Based on the Estimation Accuracy of the Natural Gradient. Florian, Théo

9) CMA-ES with Optimal Covariance Update and Storage Complexity. Eric, Clément

10) Challenges of Convex Quadratic Bi-Objective Benchmark Problems Ghassen, Moez

11) A modified ABC algorithm approach for power system harmonic estimation problems Ansaar

Assignment of Papers

2) RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. Gaetano, Francesco

3) A universal catalyst for first-order optimization. Simon, Wafa

5) Efficient optimization of many objectives by approximation-guided evolution. Gérémy

too many students for one afternoon!

New: 10 minutes talk +
10 minutes questions

...hm. Ramine, Gaspard

...A-ES Based on the ...ient. Florian, Théo

...date and Storage Complexity. Eric, Clement

10) Challenges of Convex Quadratic Bi-Objective Benchmark Problems Ghassen, Moez

11) A modified ABC algorithm approach for power system harmonic estimation problems Ansaar

Organization Oral Exams

old	Wednesday, Feb 5, 2020		new
1pm – 1:30pm	Ansaar	Ansaar	1pm – 1:25pm
1:30pm – 2pm	Simon	Simon	1:25pm – 1:50pm
2pm – 2:30pm	Wafa	Wafa	1:50pm – 2:15pm
2:30pm – 3pm	Théo	Théo	2:15pm – 2:40pm
3pm – 3:30pm	Florian	Florian	2:40pm – 3:05pm
3:30pm – 4pm	Gérémy	Gérémy	3:05pm – 3:30pm
break		Francesco	3:30pm – 3:55pm
4:30pm – 5pm	Francesco	Gaetano	3:55pm – 4:20pm
5pm – 5:30pm	Gaetano		break
5:30pm – 6pm	Eric	Eric	4:35pm – 5pm
6pm – 6:30pm	Clément	Clément	5pm – 5:25pm
6:30pm – 7pm		Ramine	5:25pm – 5:50pm
7pm – 7:30pm		Gaspard	5:50pm – 6:15pm
7:30pm – 8pm		Ghassen	6:15pm – 6:40pm
		Moez	6:40pm – 7:05pm

Course Overview

	Date		Topic
1	Wed, 27.11.2019	Dimo	Randomized Algorithms for Discrete Problems
2	Wed, 4.12.2019	Dimo	Exercise: The Travelling Salesperson Problem
3	Wed, 11.12.2019	Dimo	Evolutionary Multiobjective Optimization I
4	Mon, 16.12.2019	Dimo	Evolutionary Multiobjective Optimization II
5	Wed, 18.12.2019	Dimo	Looking at Data
	Vacation		
6	Wed, 8.1.2020 (morning!)	Anne	Continuous Optimization I
7	Wed, 22.1.2020 (morning!)	Anne	Continuous Optimization II
	Wed, 5.2.2020		oral presentations (individual time slots)

Critically Looking at Data

why?

(Some) Main Research Goals

- **novelty**
- **repeatability**
- **applicability**

A Possible Way to Learn Science...

- ...is to look at how others do it ☺
- ...is to critically ask whether what others are doing is the right thing
- ...is to get your hands dirty and tackle a difficult open question yourself (most time consuming part probably)
- ...is to actively review papers

Paper Review:
"Dynamic Search in Fireworks Algorithm“

Dynamic Search in Fireworks Algorithm

- Read Sec. V
  - Sec. V.B less important
  - read rather only until V.A and look at the results
- Do not care about what the algorithms are actually doing
- Questions:
  - What is well done in the experimental comparison?
  - What can be improved?
  - What shall be done and is not done?
  - Concretely: Mark in Tables I and II what you find remarkable

wrt. repeatability, interpretability, clarity, ...

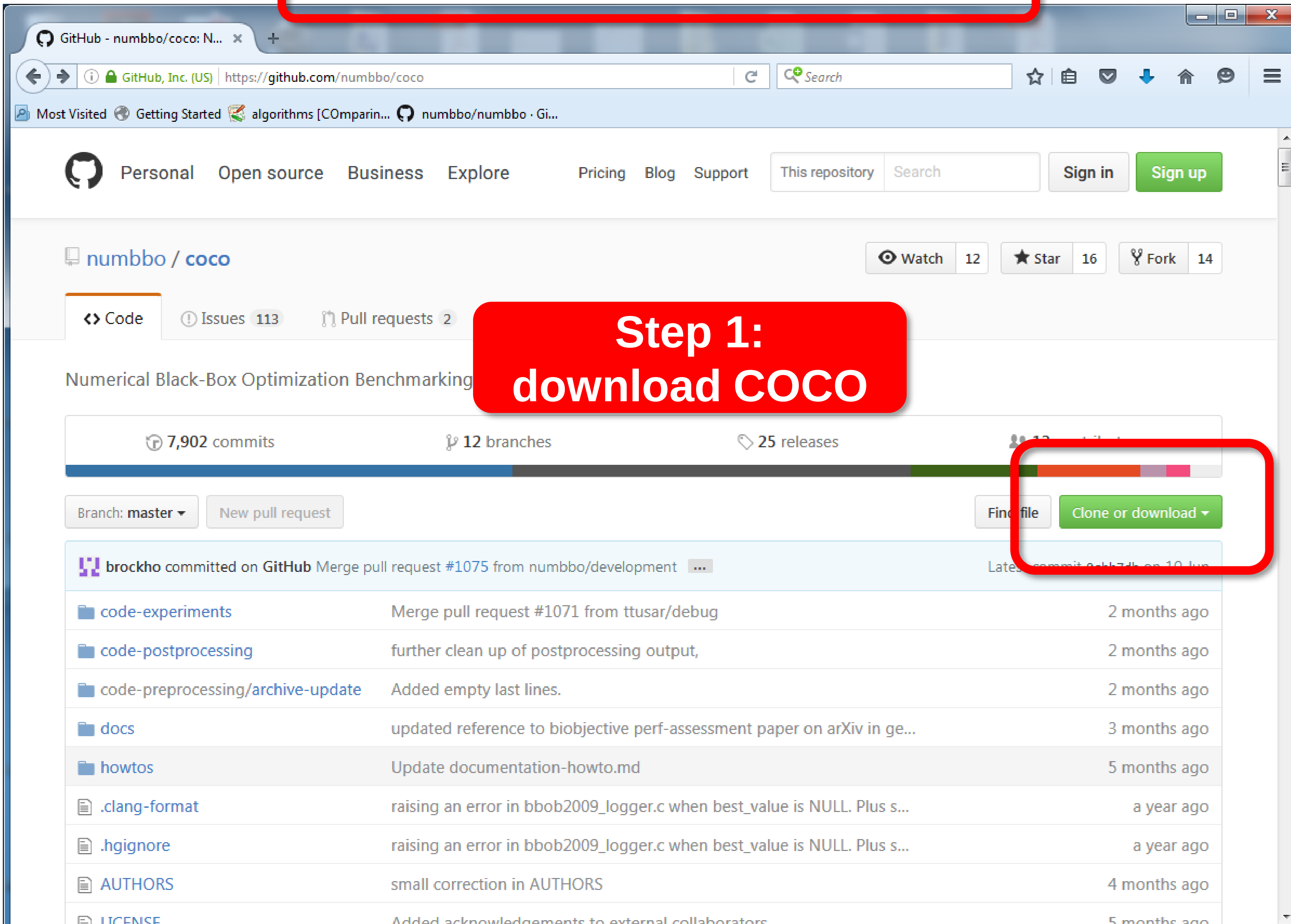
Review Form

- Short summary of what is done (3-4 sentences)
- 1-2 positive points
- 1-2 negative points (to be improved)
- Fill out the following form (from 1 (poor), over 2 (below average), 3 (average), 4 (good) till 5 (excellent)):

	Your evaluation (1—5)
Clarity	
Novelty	
Repeatability	
Consistency (what is promised vs. what is delivered)	
Significance	

Exercise: Looking at COCO Data

https://github.com/numbbo/coco



Step 1: download COCO

https://github.com/numbbo/coco

GitHub - numbbo/coco: N...

GitHub, Inc. (US) https://github.com/numbbo/coco

Most Visited Getting Started algorithms [COmparin... numbbo/numbbo · Gi...

corresponds to the master branch as linked above.

3. In a system shell, `cd` **into** the `coco` or `coco-<version>` folder (framework root), where the file `do.py` can be found. Type, i.e. **execute**, one of the following commands once

```
python do.py run-c
python do.py run-java
python do.py run-matlab
python do.py run-octave
python do.py run-python
```

depending on which language shall be used to run the experiments. `run-*` will build the respective code and run the example experiment once. The build result and the example experiment code can be found under `code-experiments/build/<language>` (`<language>=matlab` for Octave). `python do.py` lists all available commands.

4. On the computer where experiment data shall be post-processed, run

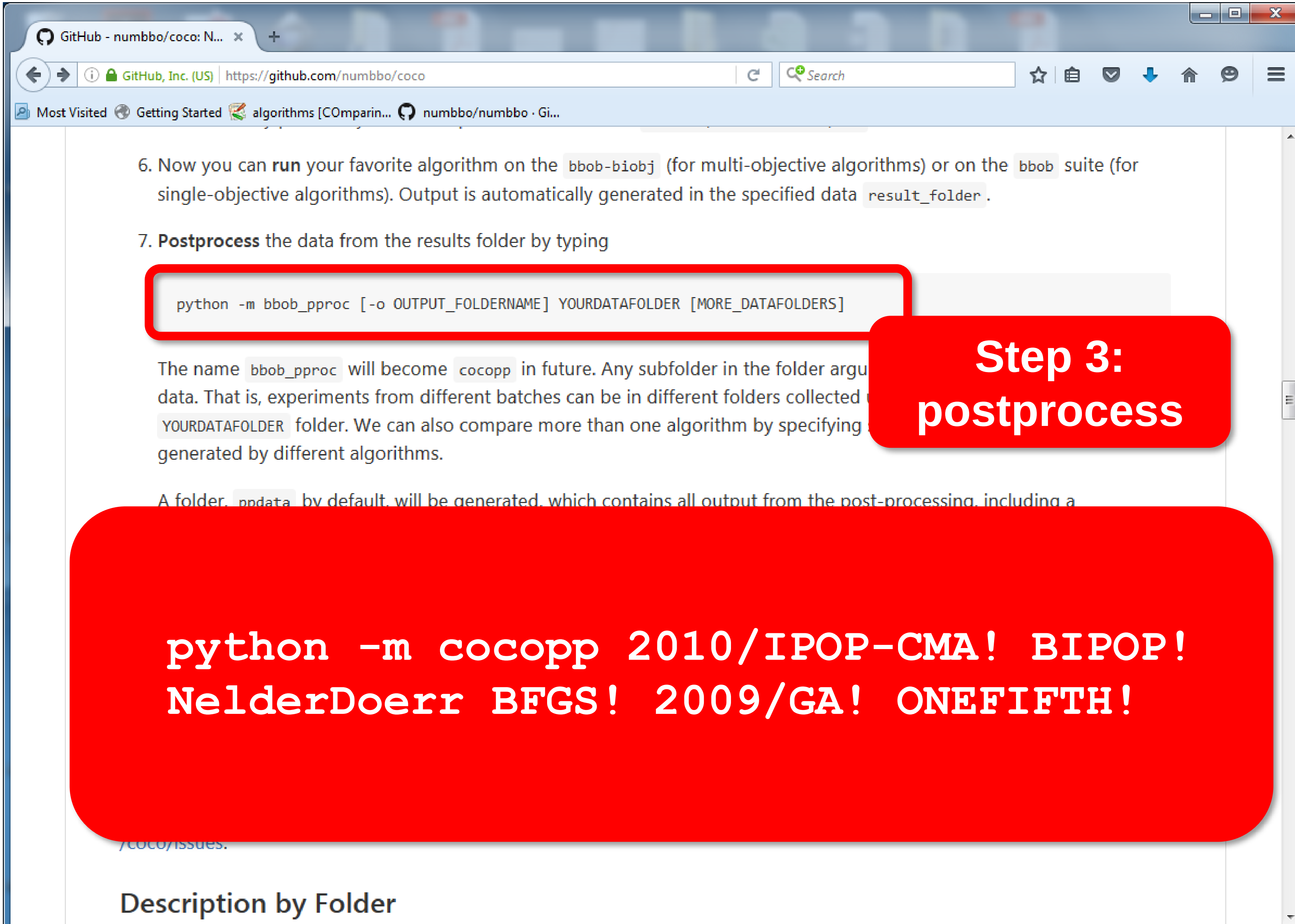
```
python do.py install-postprocessing
```

Step 2:
installation of post-processing

to (user-locally) install the post-processing. From … the builds to a new release.

5. **Copy** the folder `code-experiments/build/YOUR-FAVORITE-LANGUAGE` and its content to another location. In Python it is sufficient to copy the file `example_experiment.py`. Run the example experiment (it already is compiled, in case). As the details vary, see the respective read-me's and/or example experiment files:
   - `C` read me and example experiment
   - `Java` read me and example experiment
   - `Matlab/Octave` read me and example experiment

https://github.com/numbbo/coco



Step 3: postprocess

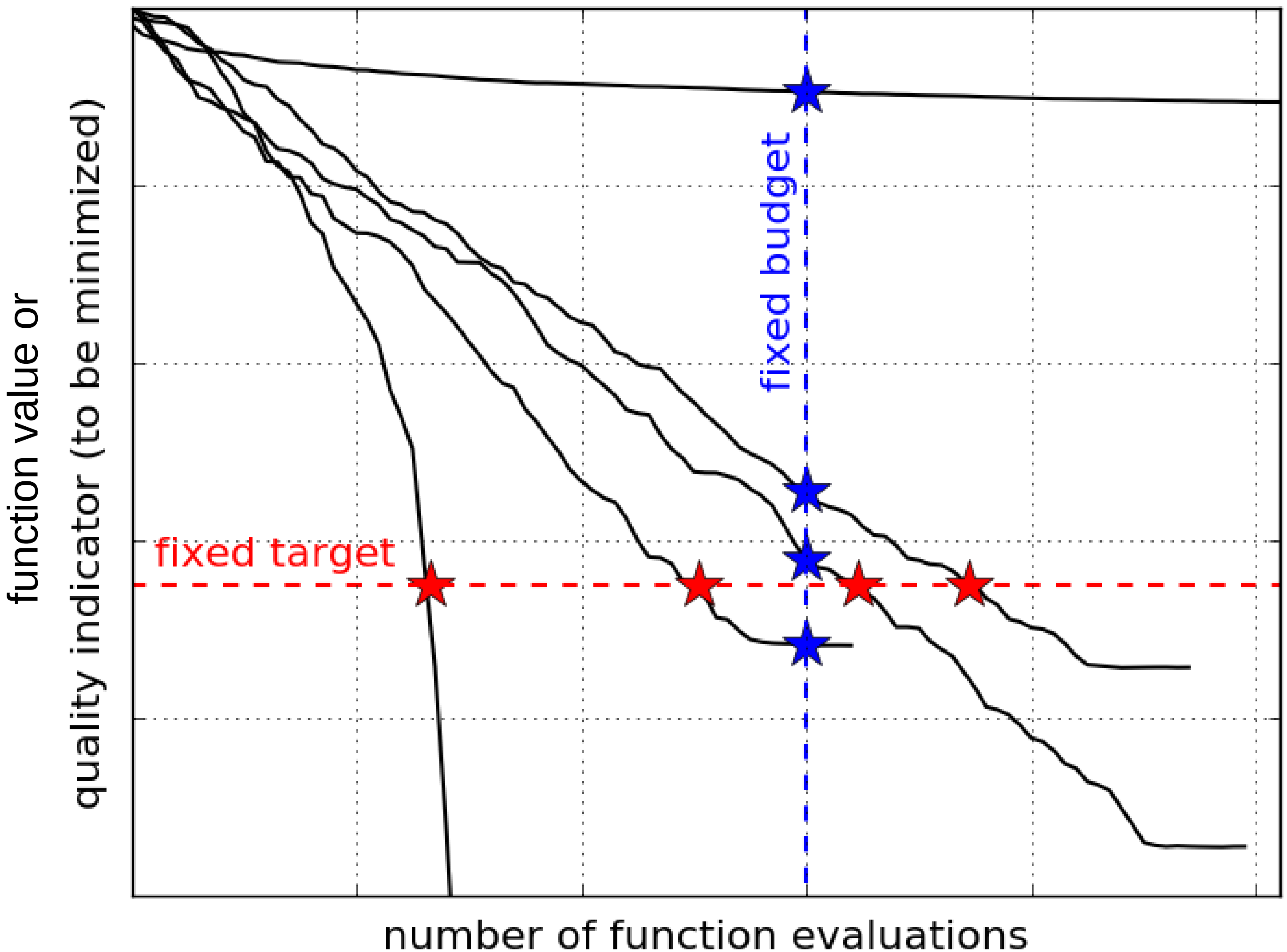
```
python -m cocopp 2010/IPOP-CMA! BIPOP!
NelderDoerr BFGS! 2009/GA! ONEFIFTH!
```

Already done for you:

http://www.cmap.polytechnique.fr/~dimo.brockhoff/advancedOptSaclay/2019/exercises/coco-results/

Measuring Performance Empirically

convergence graphs is all we have to start with...

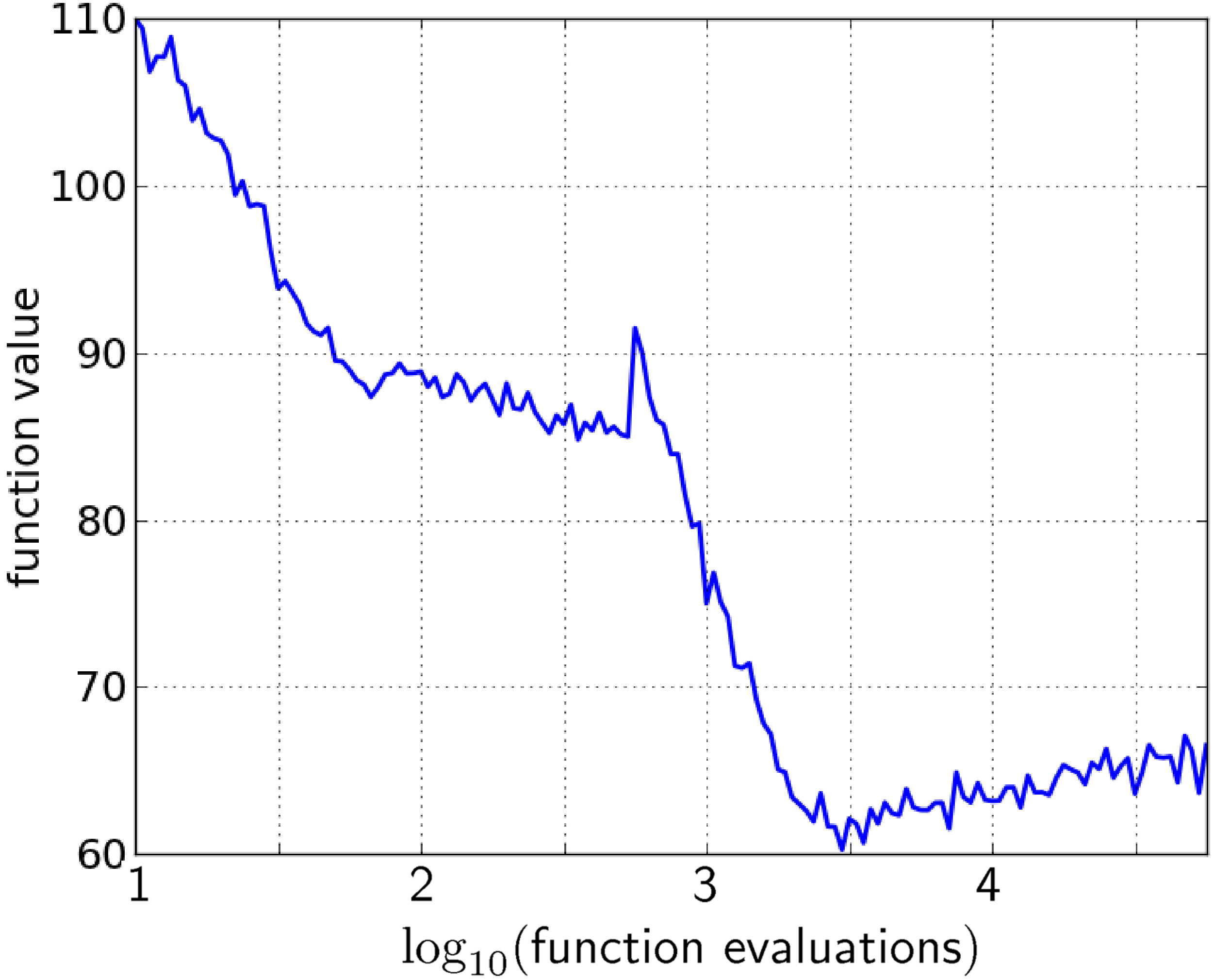


ECDF:

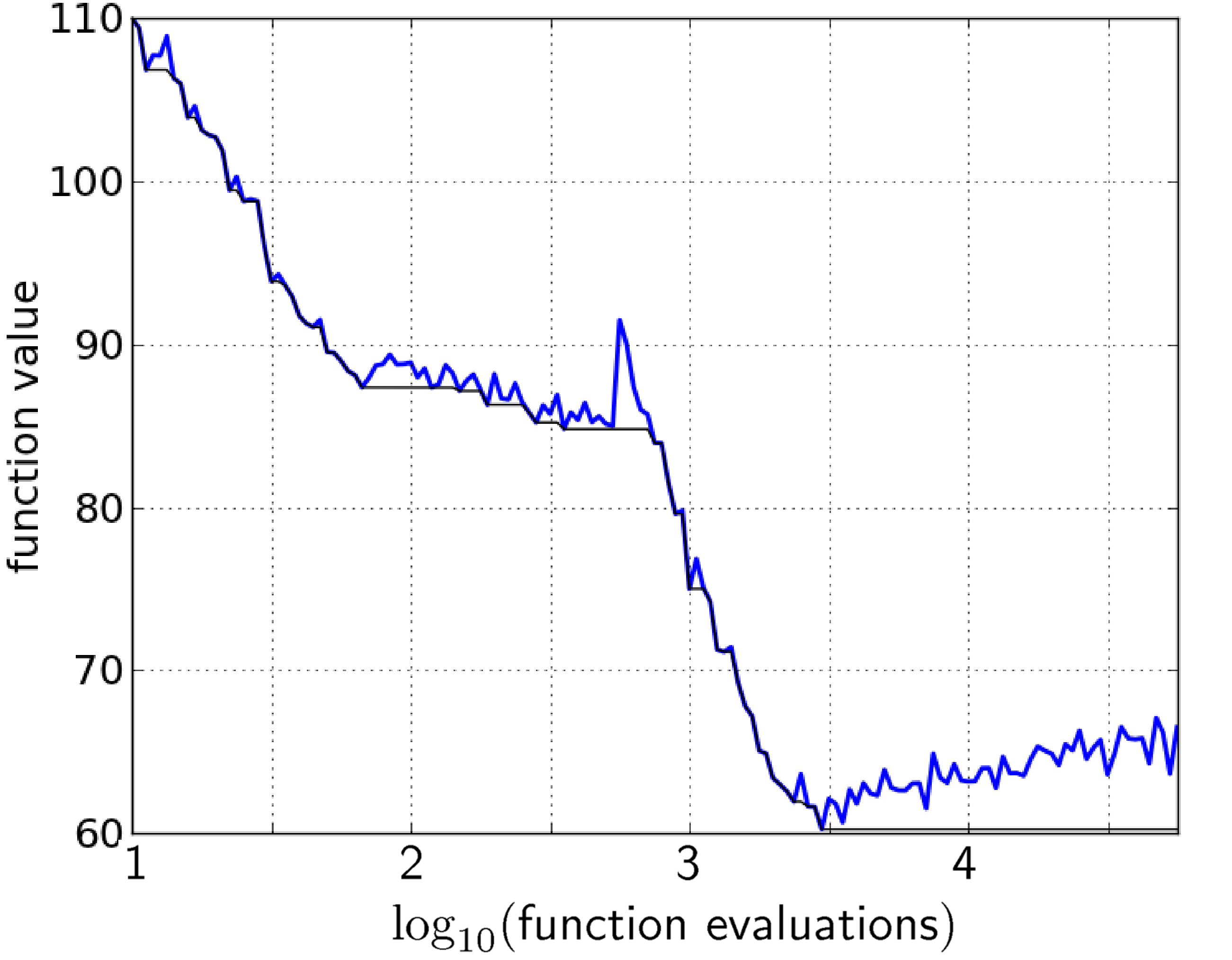
Empirical Cumulative Distribution Function of the Runtime

[aka data profile]

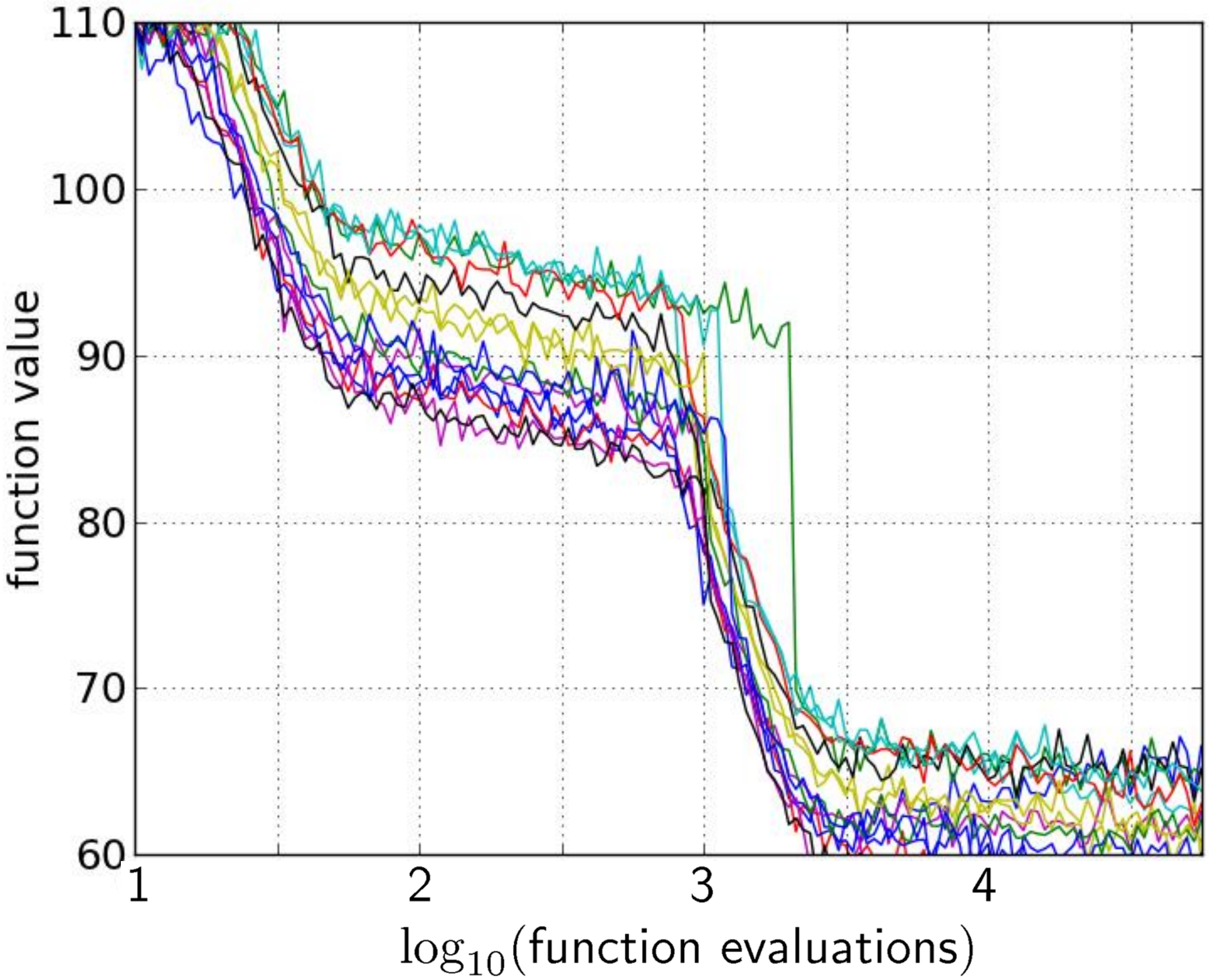
A Convergence Graph



First Hitting Time is Monotonous

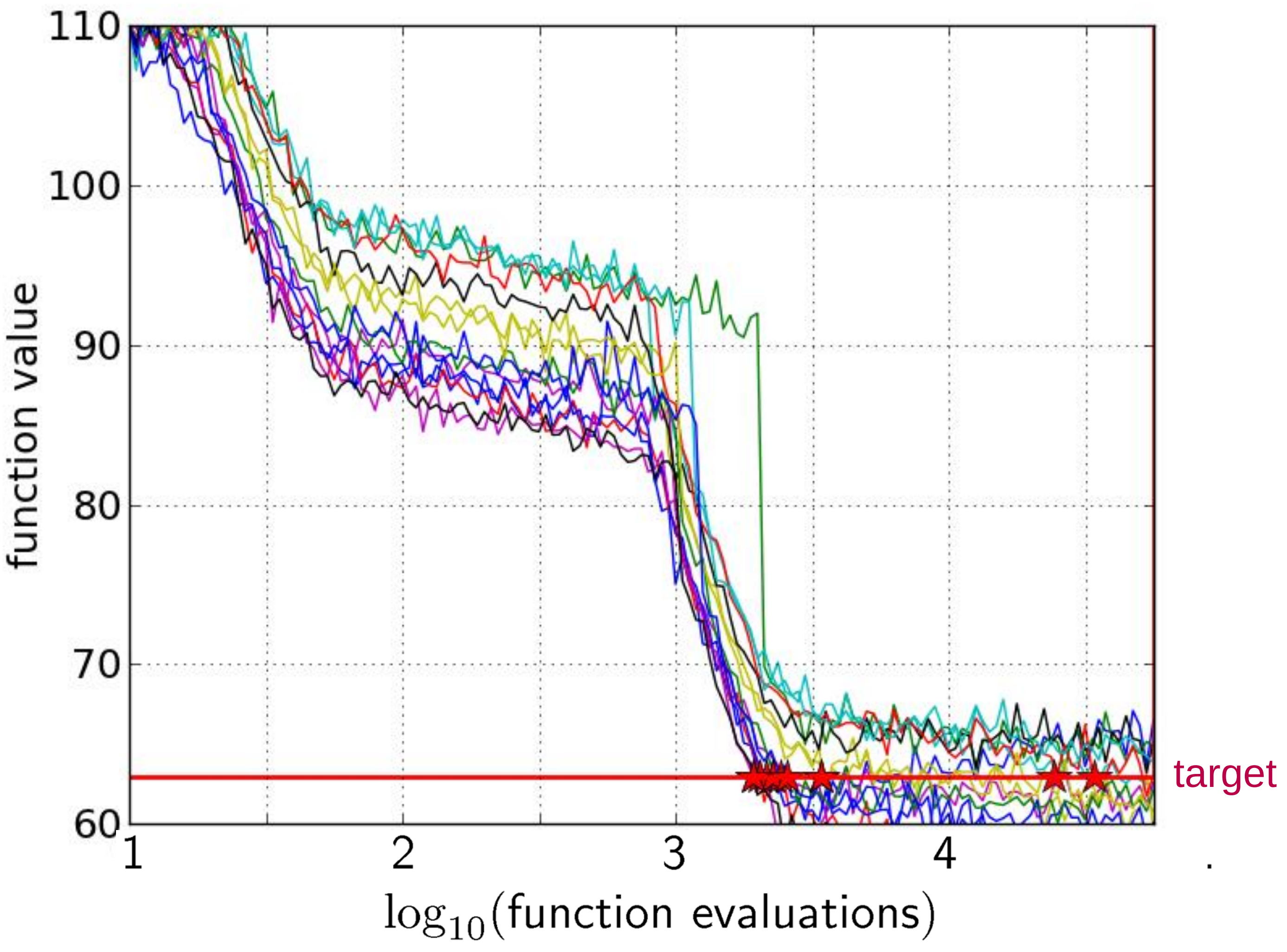


15 Runs

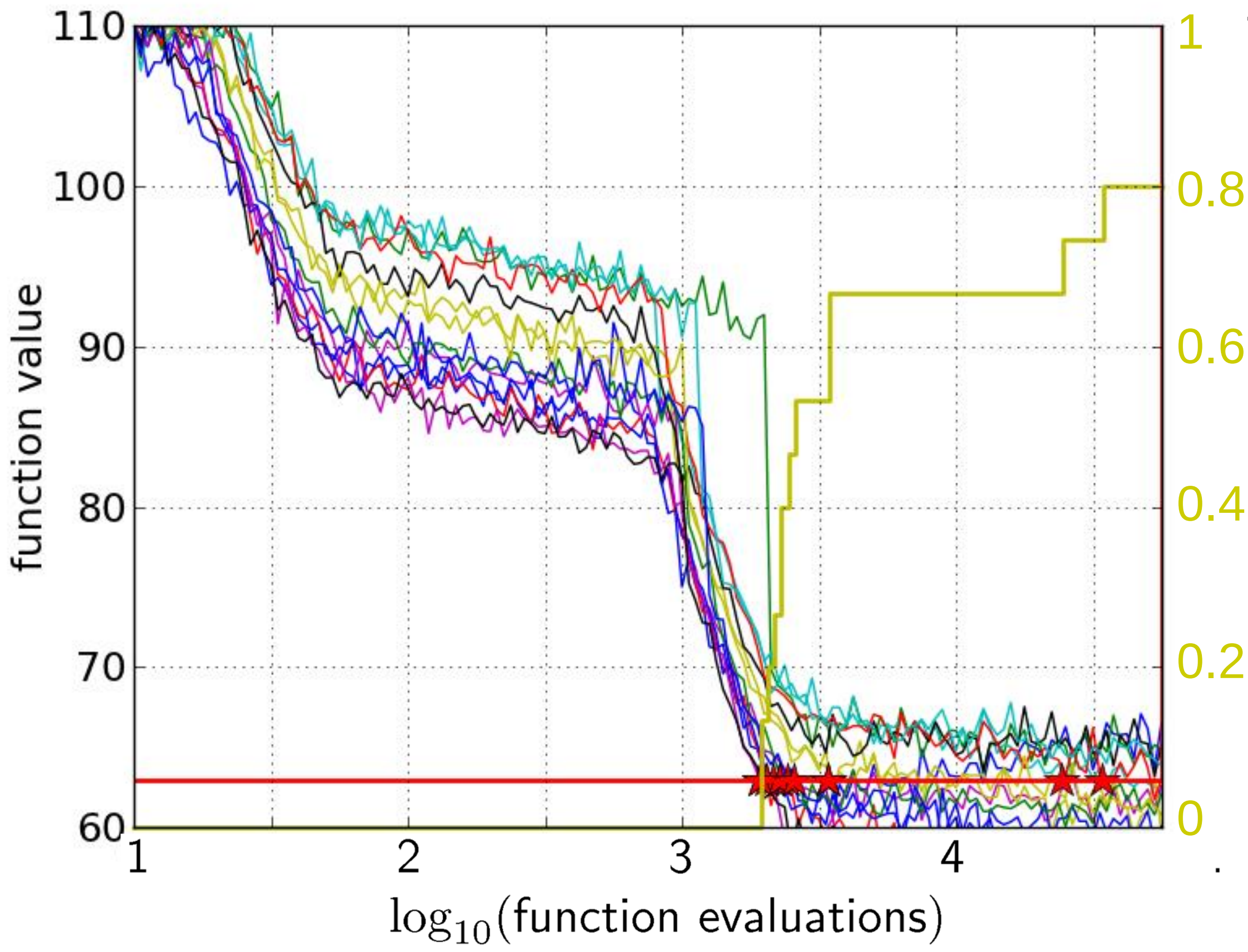

110
100
90
80
70
60
function value
1
2
3
4
$\log_{10}$(function evaluations)

.

15 Runs ≤ 15 Runtime Data Points



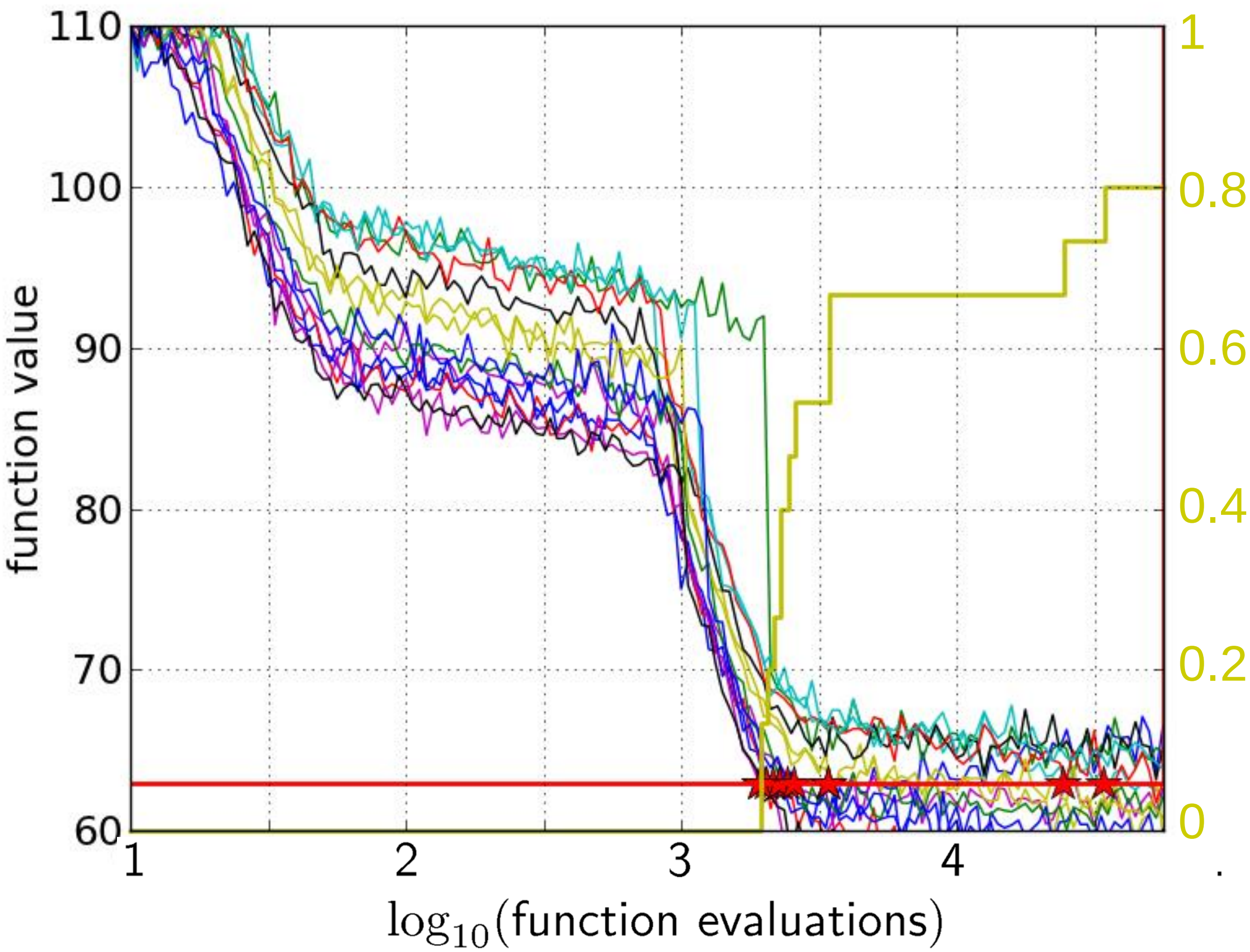
Empirical Cumulative Distribution



the ECDF of run lengths to reach the target

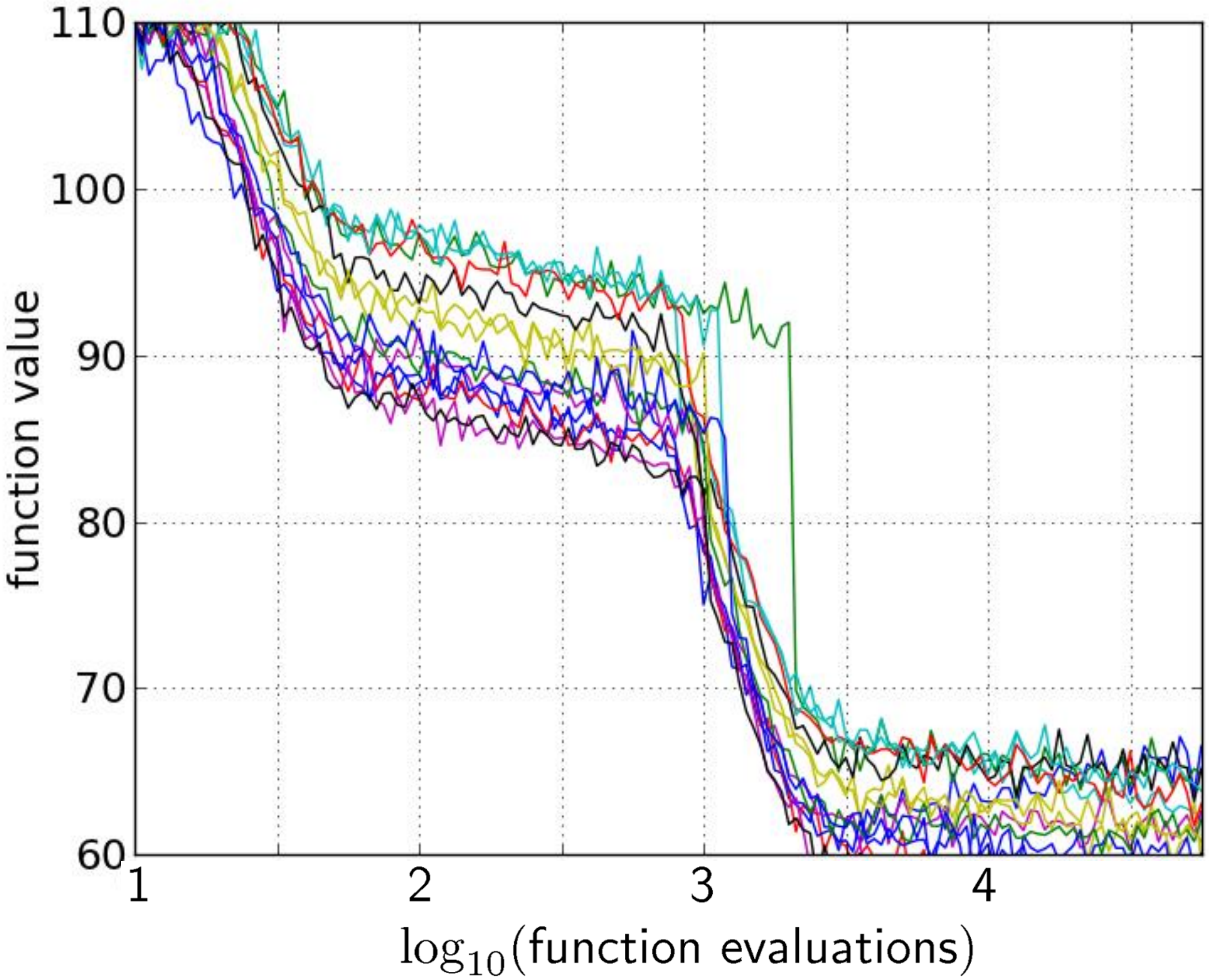
- has for each data point a vertical step of constant size
- displays for each x-value (budget) the count of observations to the left (first hitting times)

Empirical Cumulative Distribution



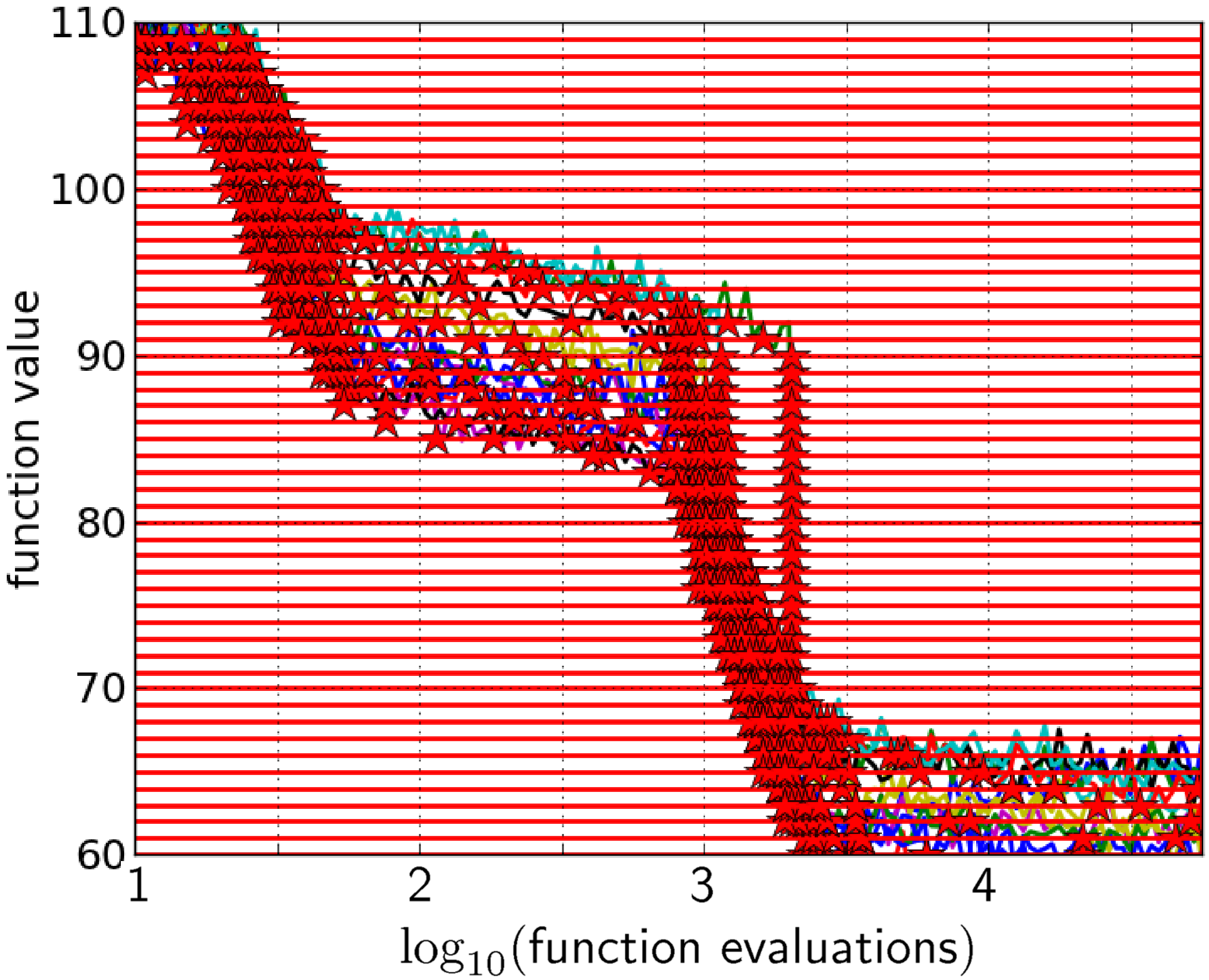
- interpretations possible:
- 80% of the runs reached the target
- e.g. 60% of the runs need between 2000 and 4000 evaluations

Aggregation



15 runs

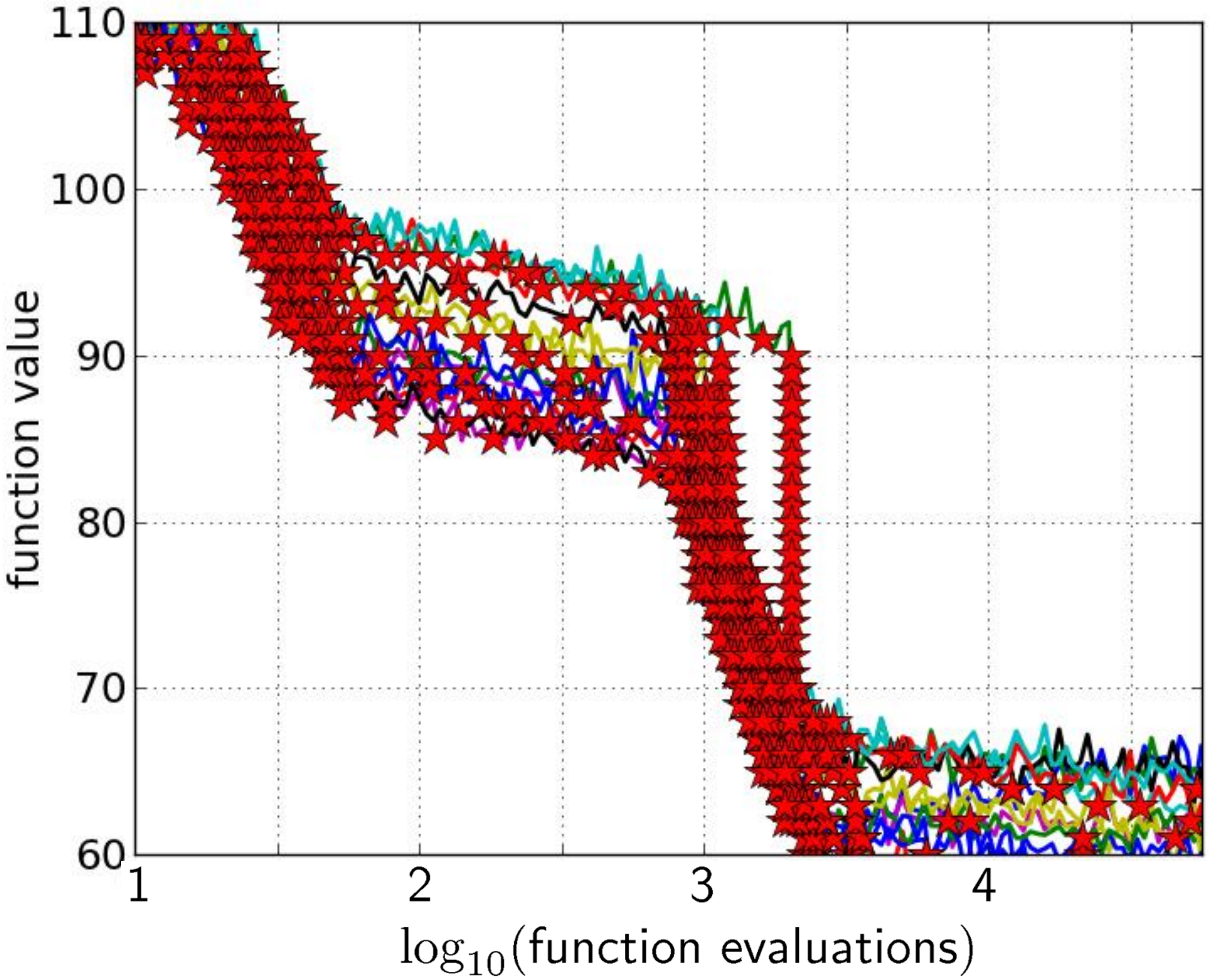
Aggregation



15 runs

50 targets

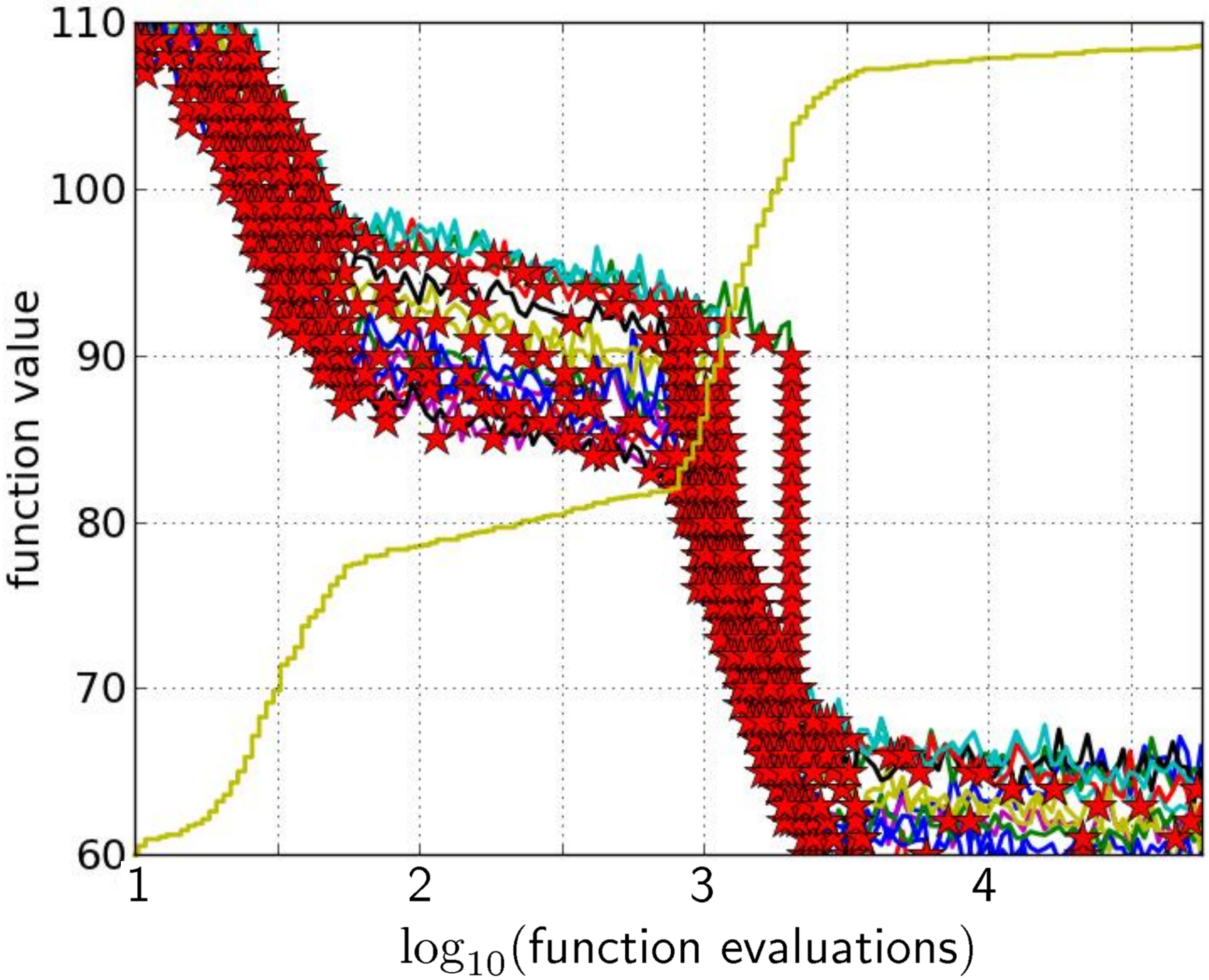
Aggregation



15 runs

50 targets

Aggregation

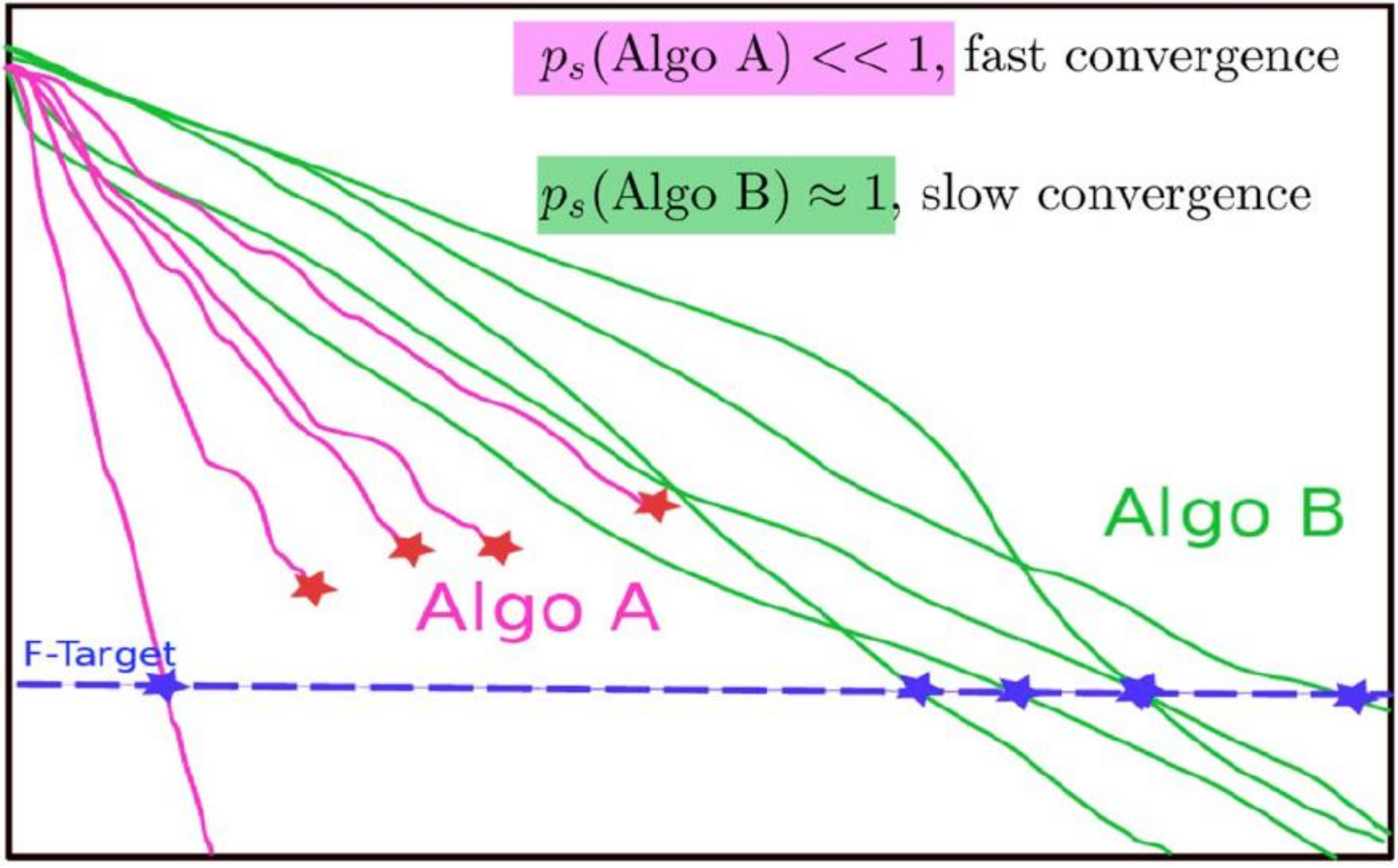


15 runs

50 targets

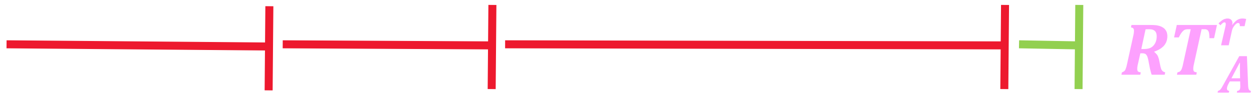
ECDF with 750 steps

Fixed-target: Measuring Runtime



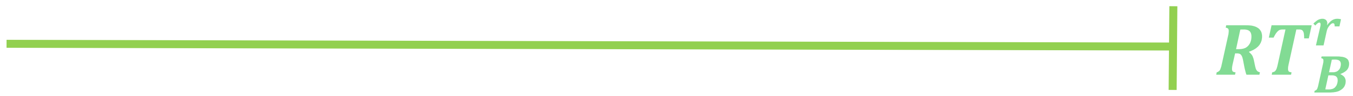
Fixed-target: Measuring Runtime

- Algo Restart A:



p_s(Algo Restart A) = 1

- Algo Restart B:



p_s(Algo Restart B) = 1

Fixed-target: Measuring Runtime

- Expected running time of the restarted algorithm:

$$E[RT^r] = \frac{1 - p_s}{p_s} E\left[RT_{unsuccessful}\right] + E[RT_{successful}]$$

- Estimator average running time (aRT):

$$\widehat{p_s} = \frac{\#\text{successes}}{\#\text{runs}}$$

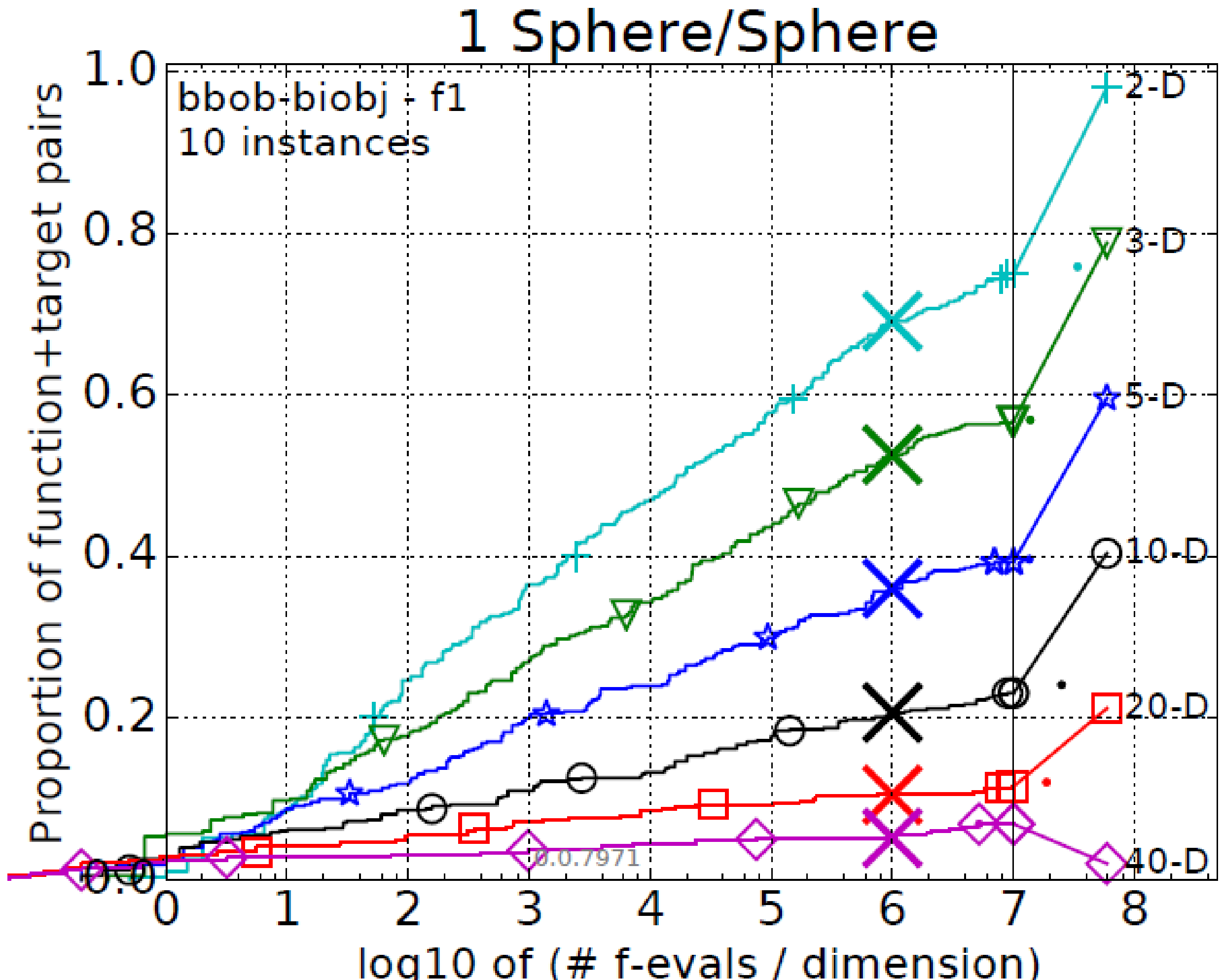
$$\widehat{RT_{unsucc}} = \text{Average evals of unsuccessful runs}$$

$$\widehat{RT_{succ}} = \text{Average evals of successful runs}$$

$$aRT = \frac{\text{total \#evals}}{\#\text{successes}}$$

ECDFs with Simulated Restarts

What we typically plot are ECDFs of the simulated restarted algorithms:



The single-objective BBOB functions

https://coco.gforge.inria.fr/downloads/download16.00/bbobdocfunctions.pdf

The bbob Testbed

- 24 functions in 5 groups:

1 Separable Functions	
f1	Sphere Function
f2	Ellipsoidal Function
f3	Rastrigin Function
f4	Büche-Rastrigin Function
f5	Linear Slope
2 Functions with low or moderate conditioning	
f6	Attractive Sector Function
f7	Step Ellipsoidal Function
f8	Rosenbrock Function, original
f9	Rosenbrock Function, rotated
3 Functions with high conditioning and unimodal	
f10	Ellipsoidal Function
f11	Discus Function
f12	Bent Cigar Function
f13	Sharp Ridge Function
f14	Different Powers Function

4 Multi-modal functions with adequate global structure	
f15	Rastrigin Function
f16	Weierstrass Function
f17	Schaffers F7 Function
f18	Schaffers F7 Functions, moderately ill-conditioned
f19	Composite Griewank-Rosenbrock Function F8F2
5 Multi-modal functions with weak global structure	
f20	Schwefel Function
f21	Gallagher's Gaussian 101-me Peaks Function
f22	Gallagher's Gaussian 21-hi Peaks Function
f23	Katsuura Function
f24	Lunacek bi-Rastrigin Function

- 6 dimensions: 2, 3, 5, 10, 20, (40 optional)

Notion of Instances

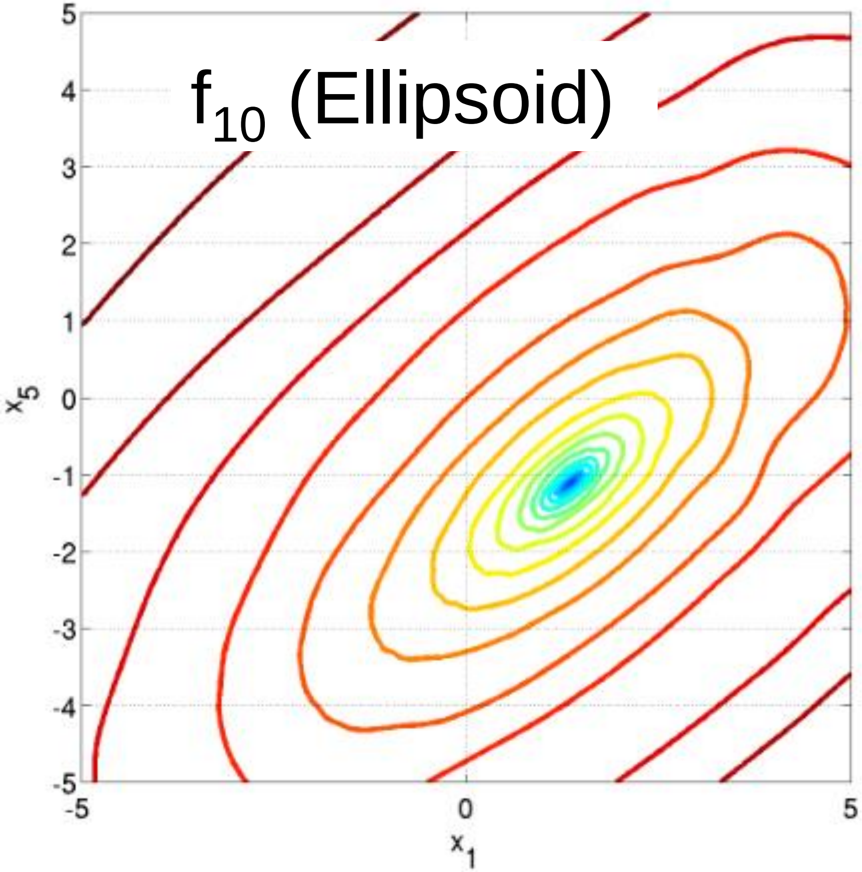
- All COCO problems come in form of instances
  - e.g. as translated/rotated versions of the same function
- Prescribed instances typically change from year to year
  - avoid overfitting
  - 5 instances are always kept the same

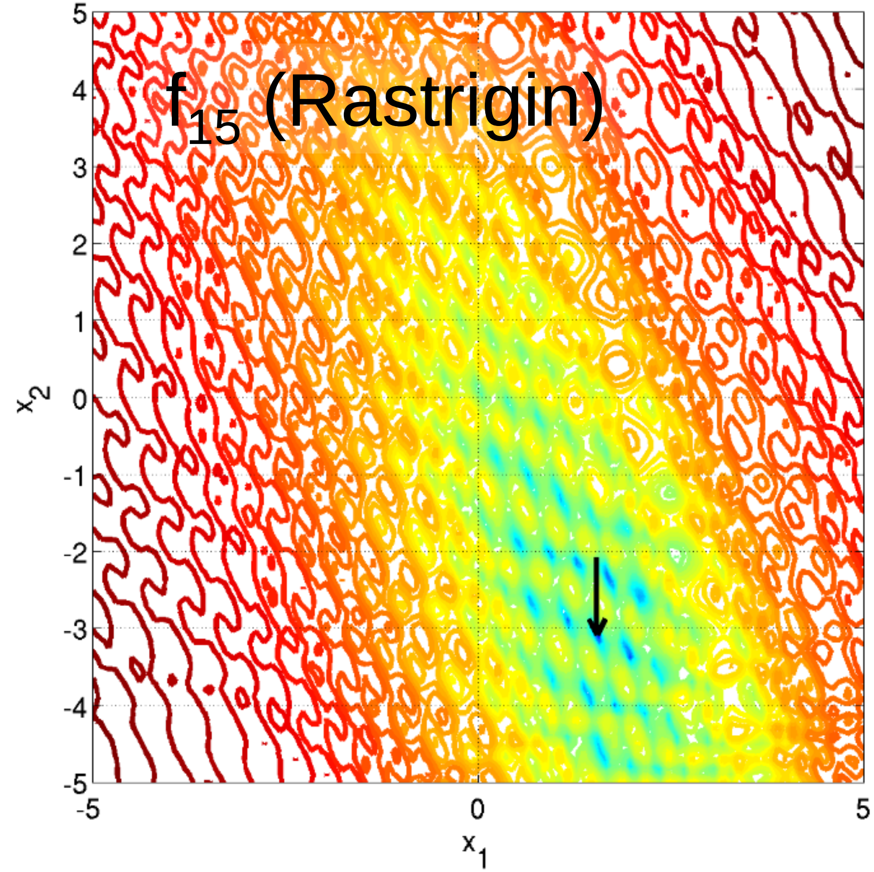
Plus:

  - the bbob functions are locally perturbed by non-linear transformations

Notion of Instances

- All COCO problems come in form of instances





linear transformations

Exercise (Part 1)

Objectives:

- investigate the performance of these 6 algorithms:
  - CMA-ES ("IPOP-CMA-ES" version)
  - CMA-ES ("BIPOP-CMA-ES" version)
  - Nelder-Mead simplex (use "NelderDoerr" version here)
  - BFGS quasi-Newton
  - Genetic Algorithm: discretization of cont. variables ("GA")
  - ONEFIFTH: (1+1)-ES with 1/5 rule
- postprocessed available here: **`http://www.cmap.polytechnique.fr/~dimo.brockhoff/advancedOptSaclay/2019/exercises/coco-results/`**
- so now: investigate the data!

Exercise (Part 2)

Objective:

investigate the data:

a) which algorithms are the best ones?
b) does this depend on the dimension? Or on other things?
c) look at single graphs: can we say something about the algorithms' invariances, e.g. wrt. rotations of the search space?
d) what's the impact of covariance-matrix-adaptation?
e) what do you think: are the displayed algorithms well-suited for problems with larger dimension?