13th GECCO Workshop on Blackbox Optimization Benchmarking (BBOB): Welcome and Introduction to COCO/BBOB

The BBOBies:

A. Auger, D. Brockhoff, T. Glasmachers, N. Hansen, O. Mersmann, T. Tušar

https://coco-platform.org

slides based on previous ones by A. Auger, N. Hansen, and D. Brockhoff

Practical Blackbox Optimization

Given:

Not clear:

which of the many algorithms should I use on my problem?

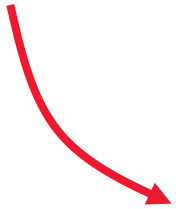
Practical Need: Benchmarking

- understanding of algorithms
- algorithm selection/recommendation
- putting algorithms to a standardized test
  - simplify judgement
  - simplify comparison
  - regression test under algorithm changes

Kind of everybody has to do it (and it is tedious):

- choosing (and implementing) problems, performance measures, visualization, stat. tests, ...
- running a set of algorithms

that's where COCO and BBOB come into play

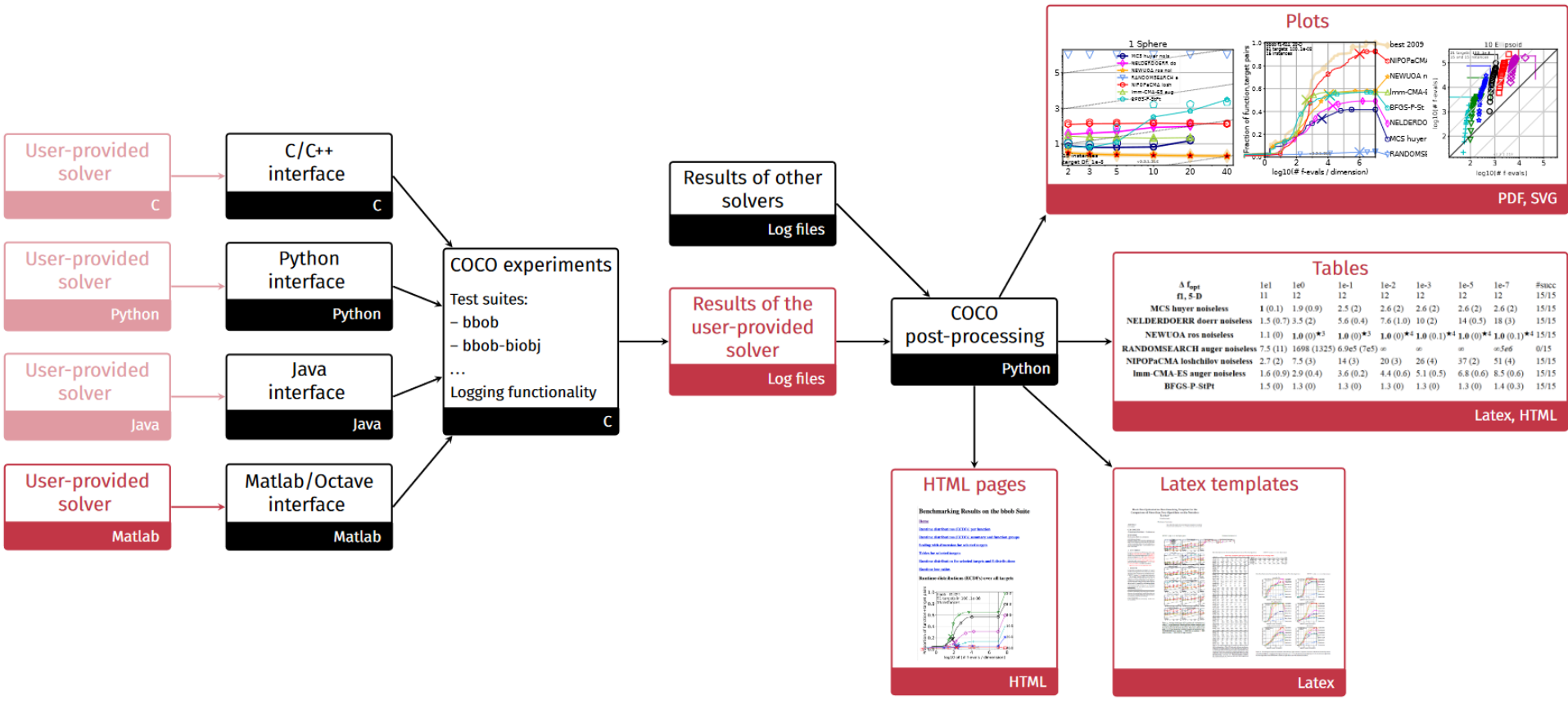


Comparing Continuous Optimizers Platform

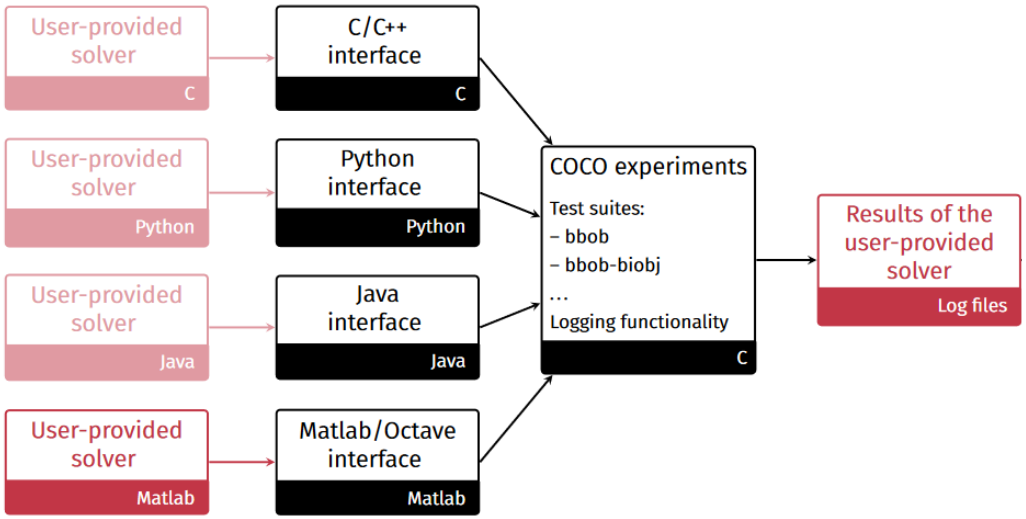
`https://coco-platform.org`

automatized benchmarking

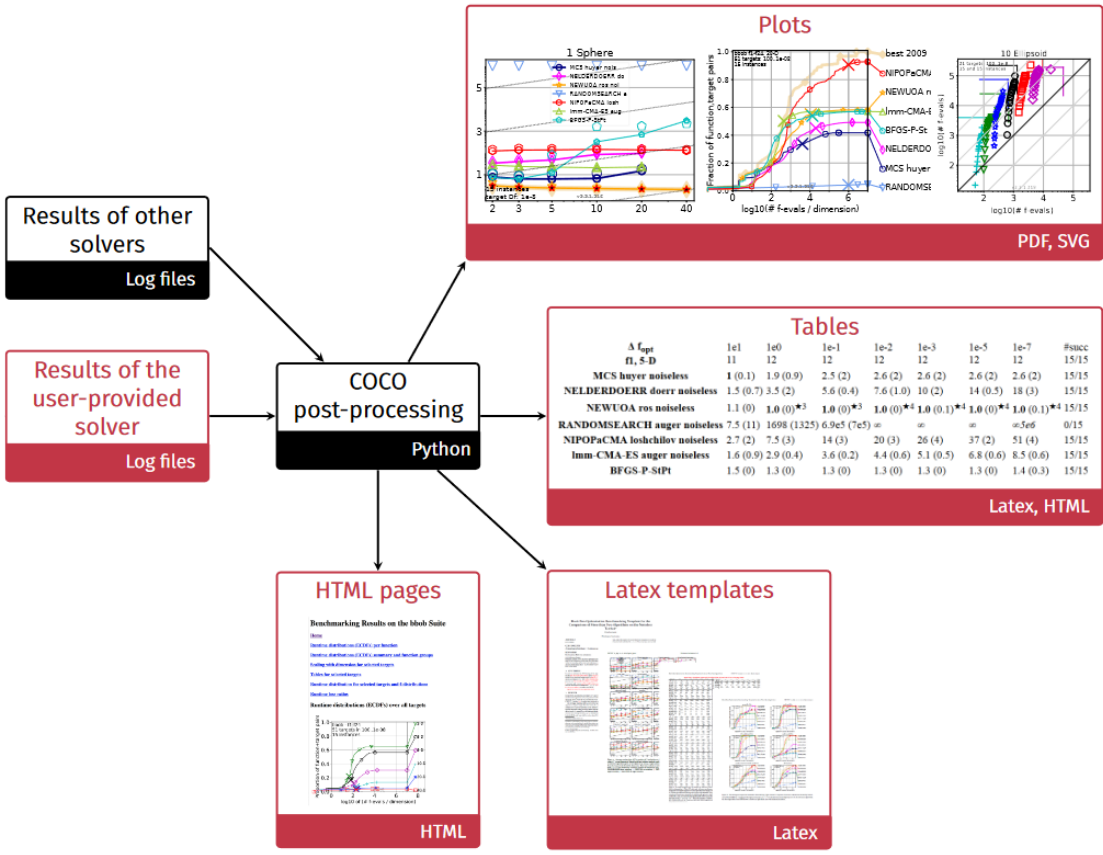
Overview of COCO's Structure



Overview of COCO's Structure



Overview of COCO's Structure



COCO implements a

reasonable, well-founded, and well-documented

pre-chosen methodology

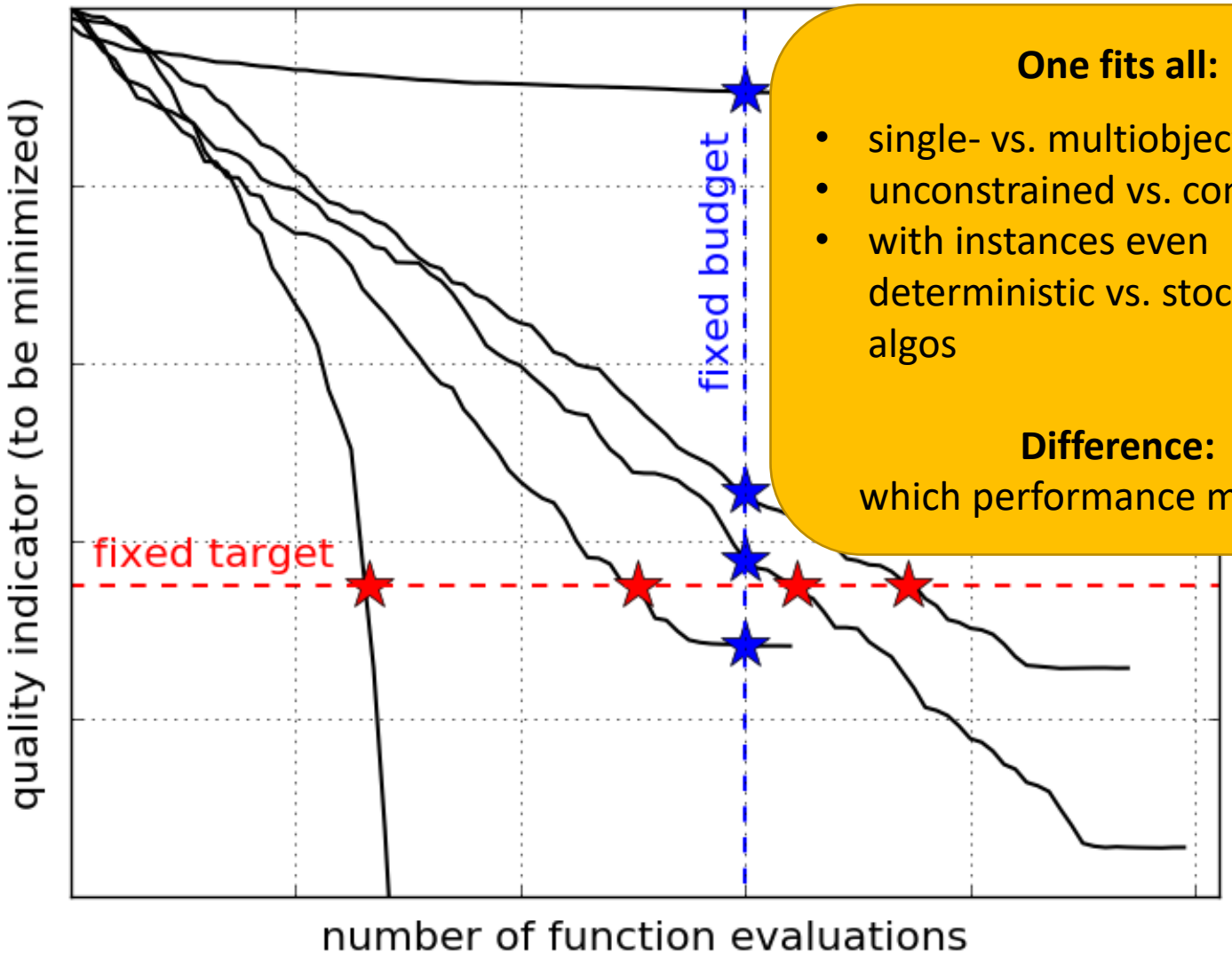
main performance measure:

runtime

until a certain target difficulty is reached

Measuring Performance Empirically

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



One fits all:

- single- vs. multiobjective
- unconstrained vs. constrained
- with instances even deterministic vs. stochastic algos

Difference:
which performance measure?

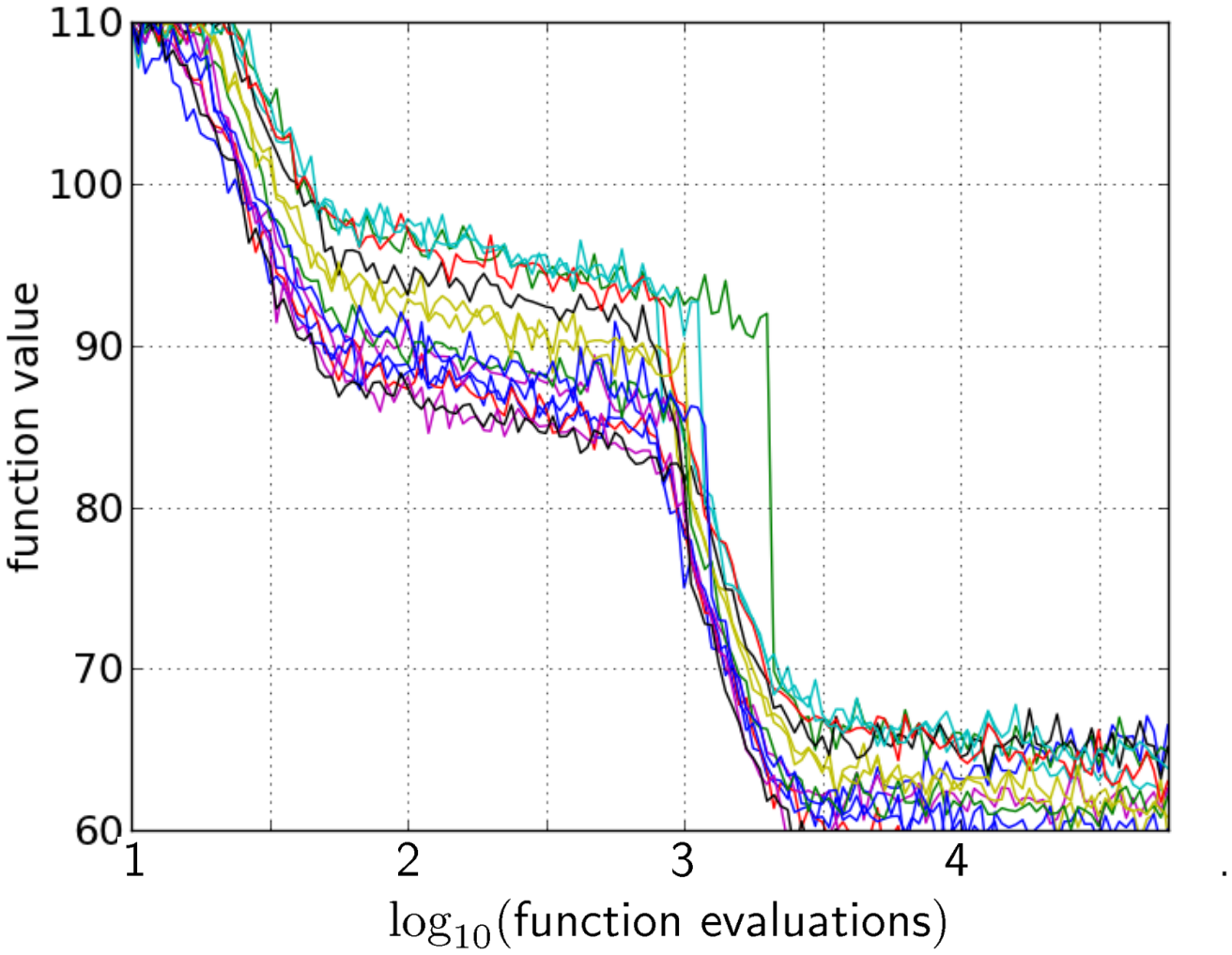
Main Performance Visualization:

Empirical Runtime Distributions

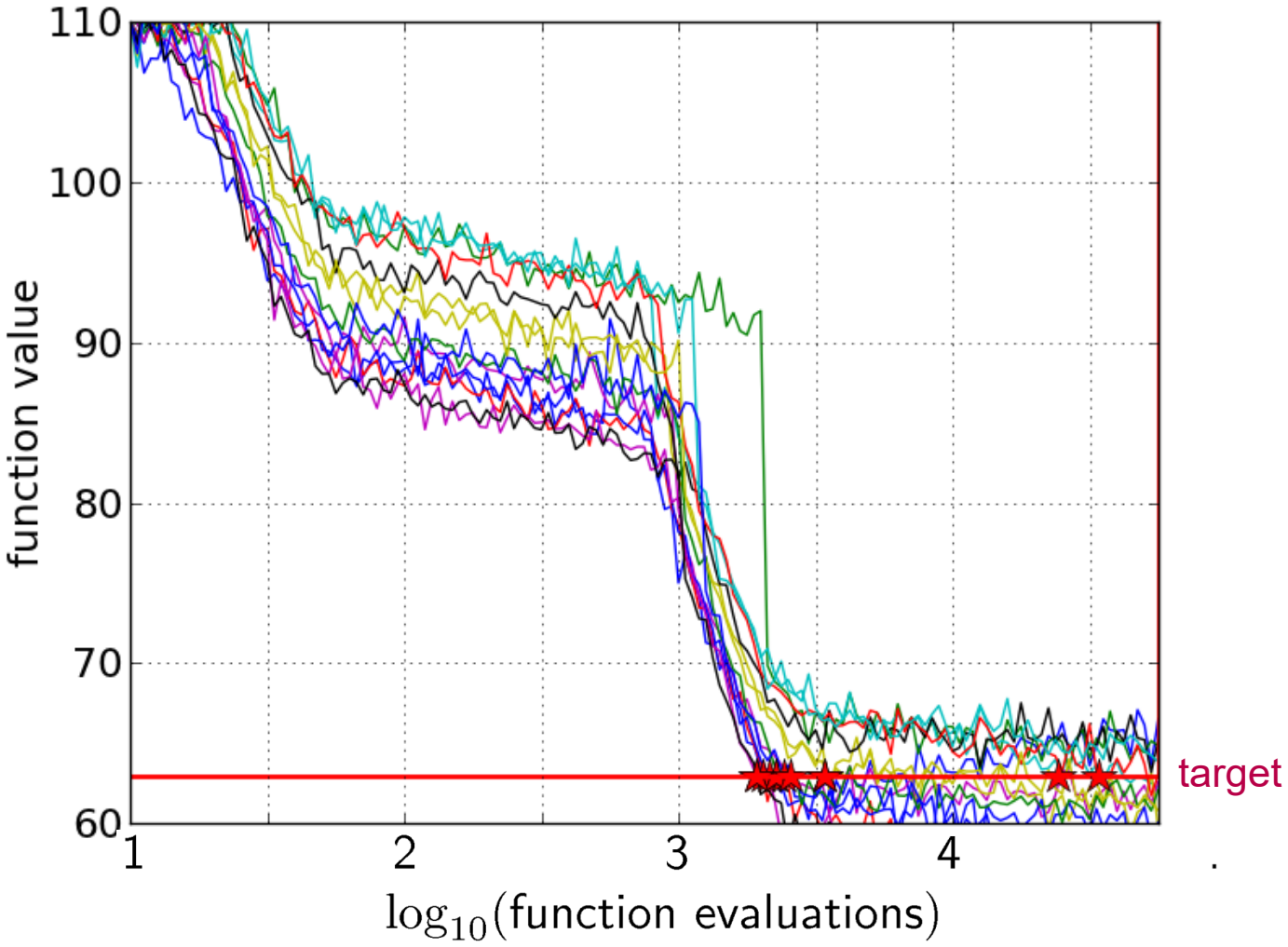
[aka Empirical Cumulative Distribution Function (ECDF) of the Runtime]

[aka data profile with multiple&absolute targets]

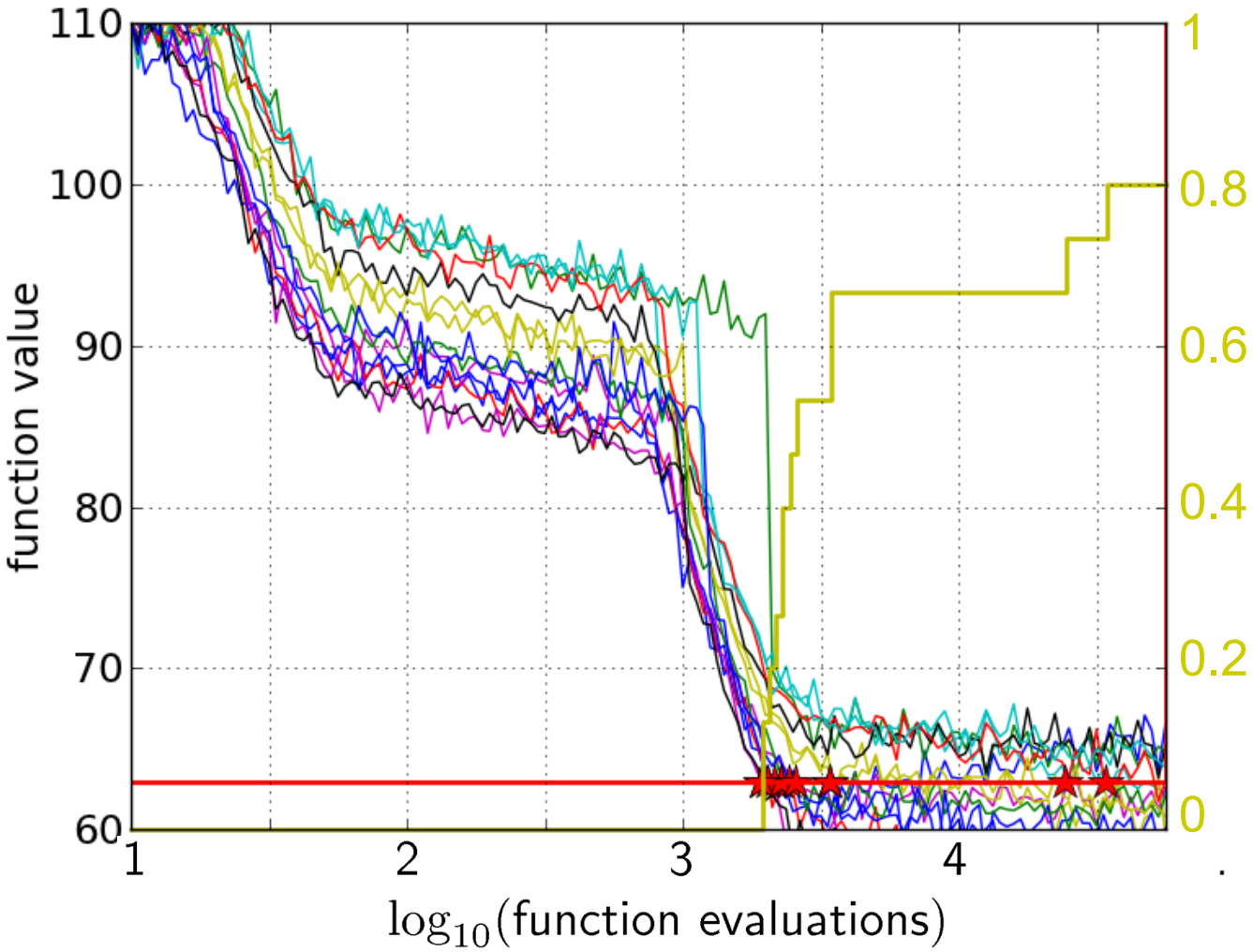
Convergence Graph of 15 Runs



15 Runs ≤ 15 Runtime Data Points



Empirical Cumulative Distribution

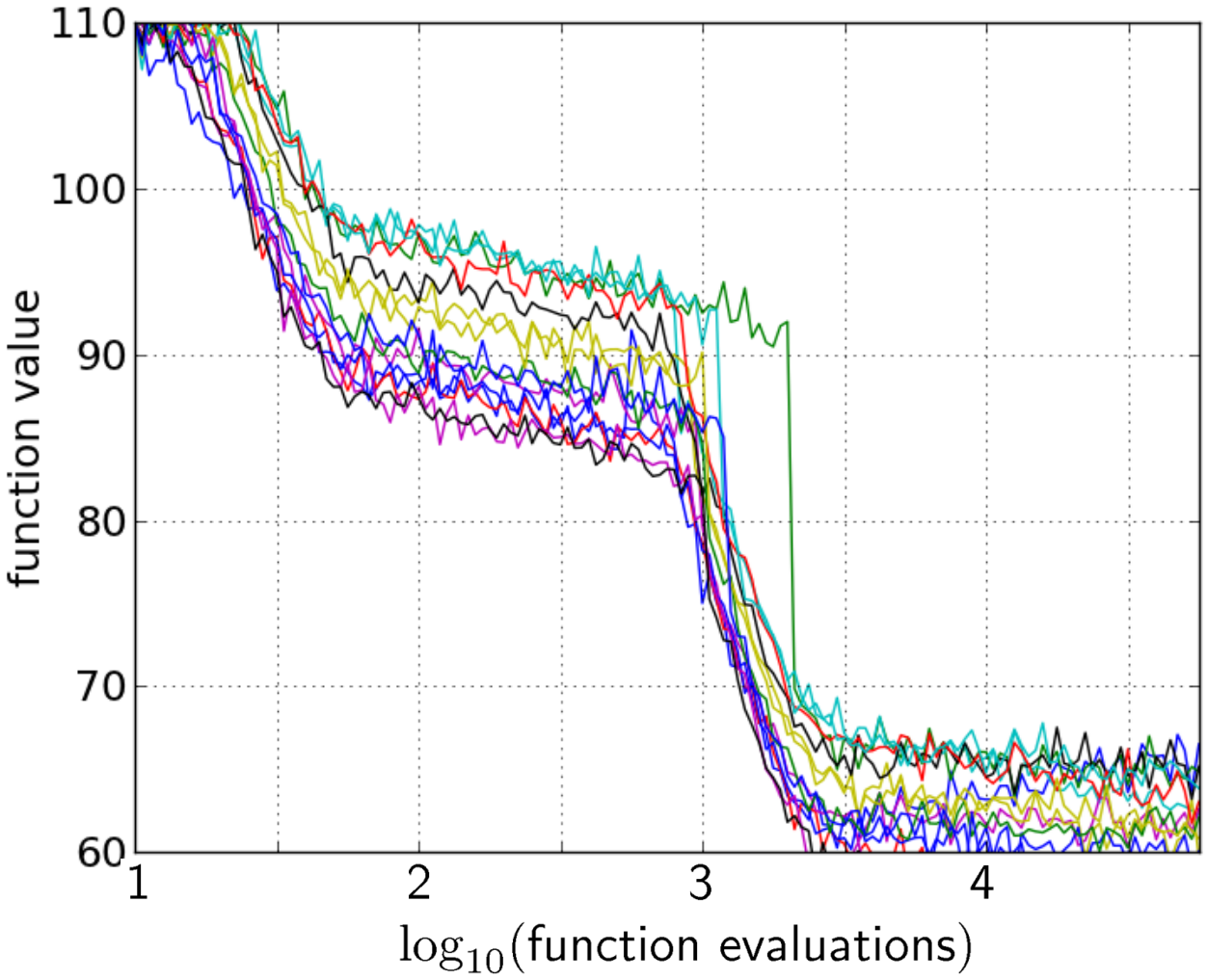


the ECDF of run lengths to reach the target

- 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)

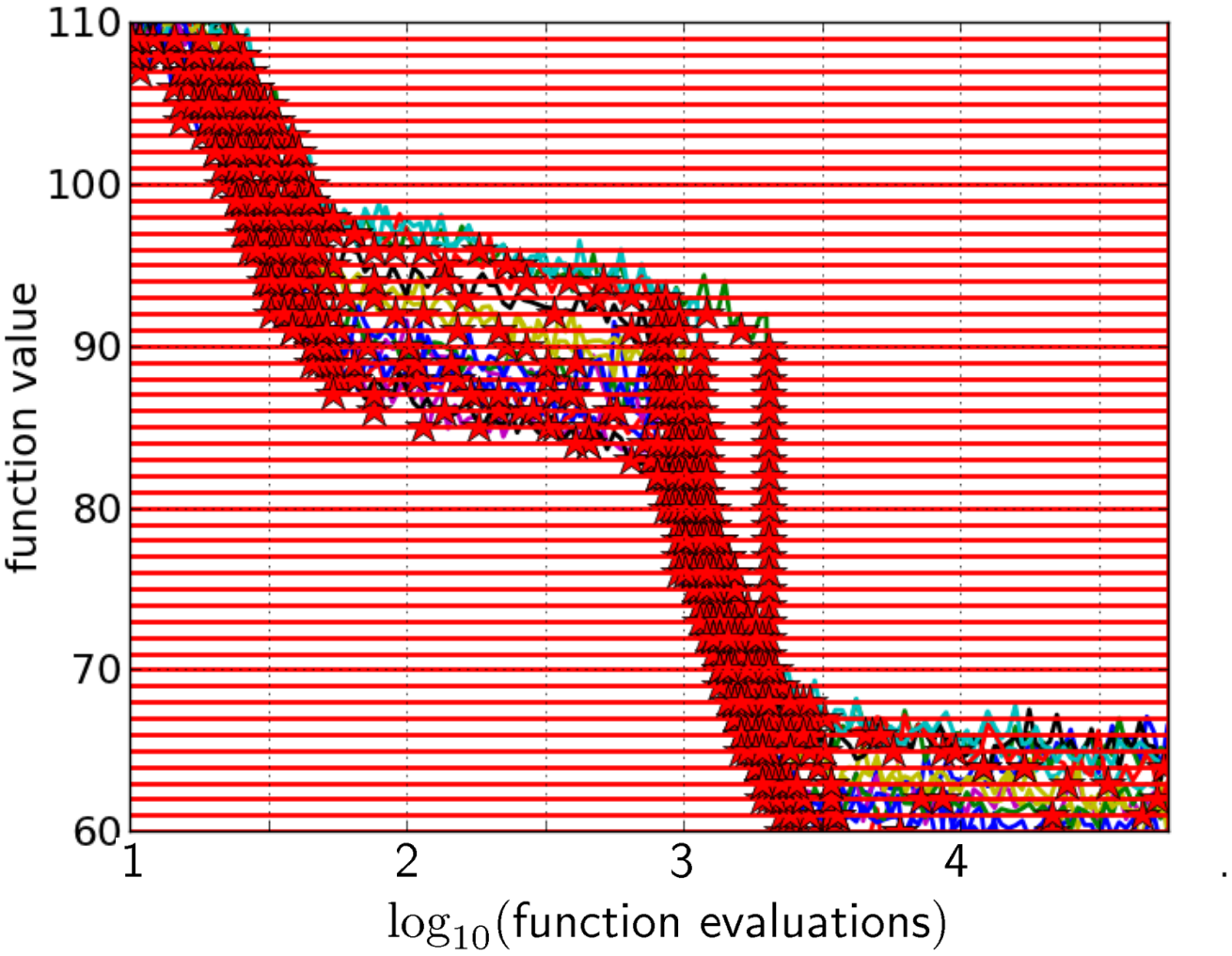
e.g. 60% of the runs need between 2000 and 4000 evaluations
80% of the runs reached the target

Aggregation



15 runs

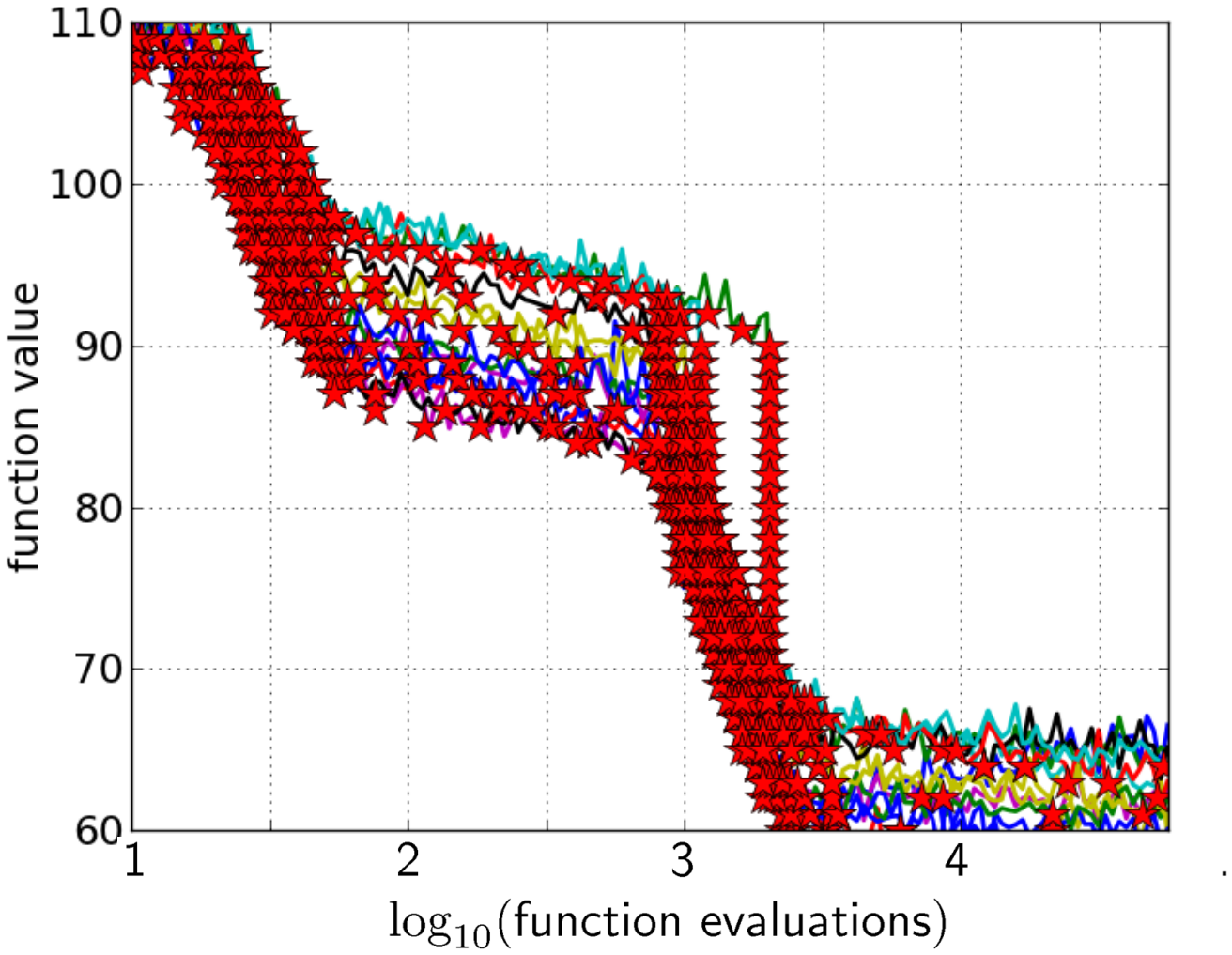
Aggregation



15 runs

50 targets

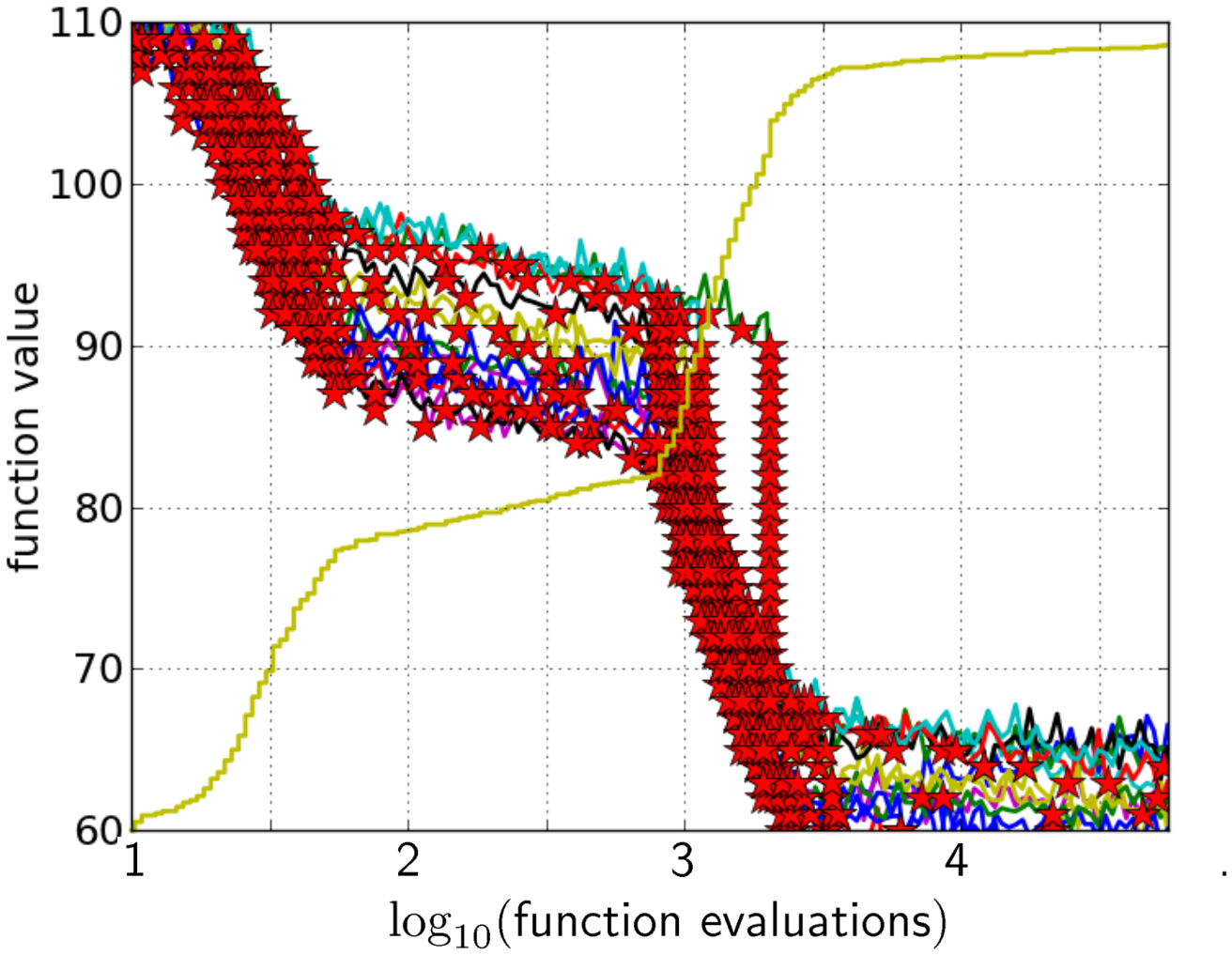
Aggregation



15 runs

50 targets

Aggregation

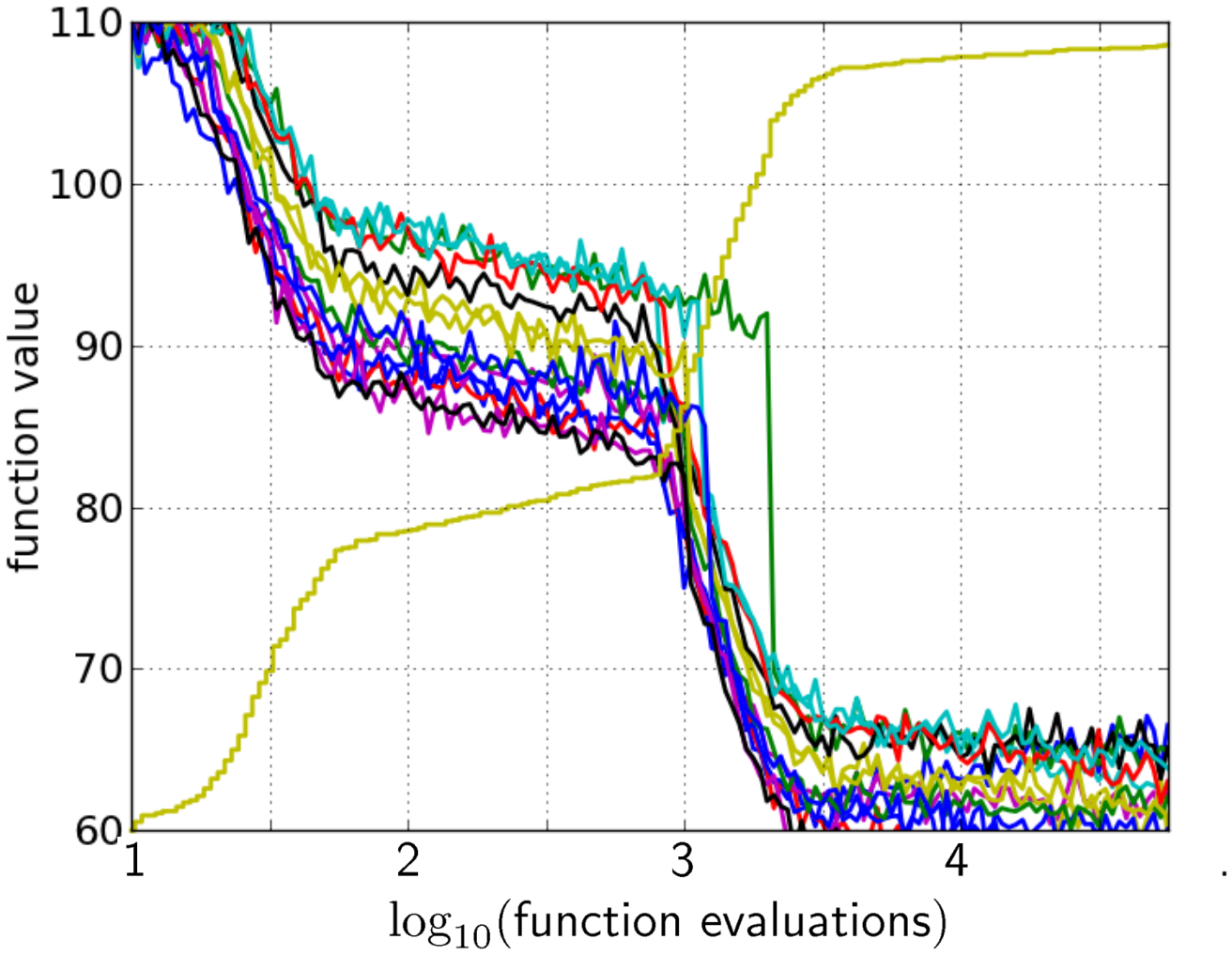


15 runs

50 targets

ECDF with 750 steps

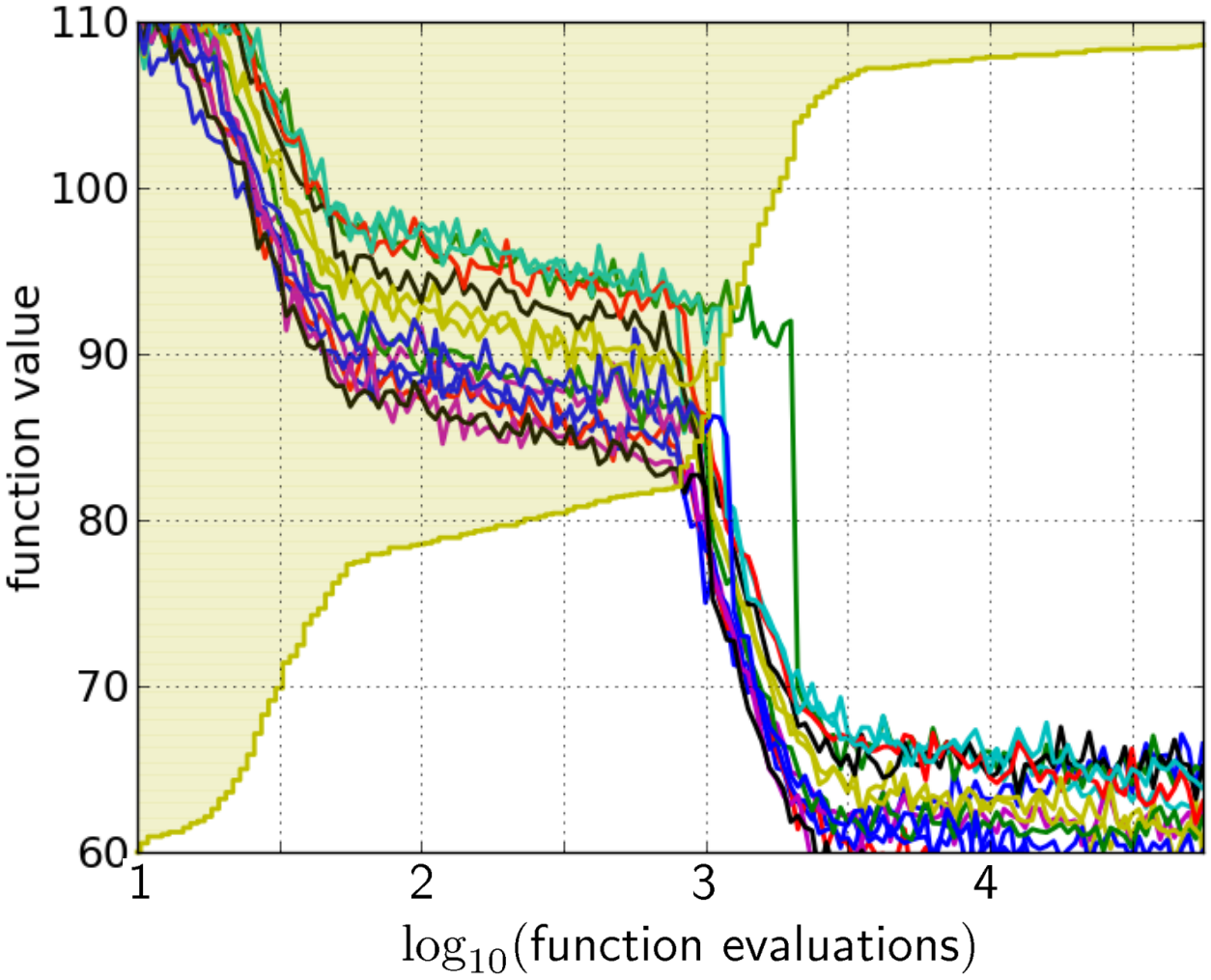
Aggregation



50 targets from
15 runs

...integrated in a
single graph

Interpretation



50 targets from
15 runs
integrated in a
single graph

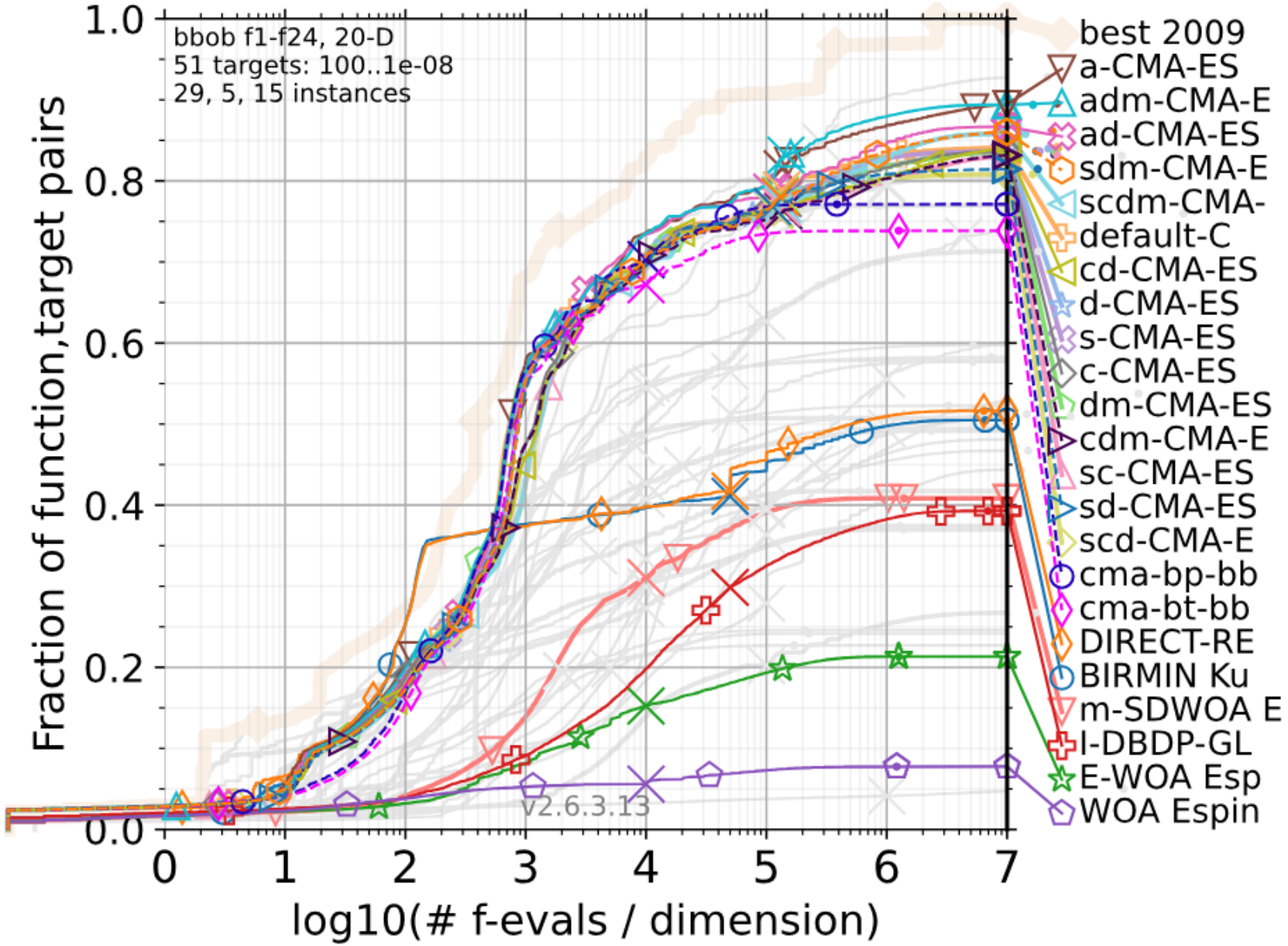
area over the ECDF curve

=

average log runtime

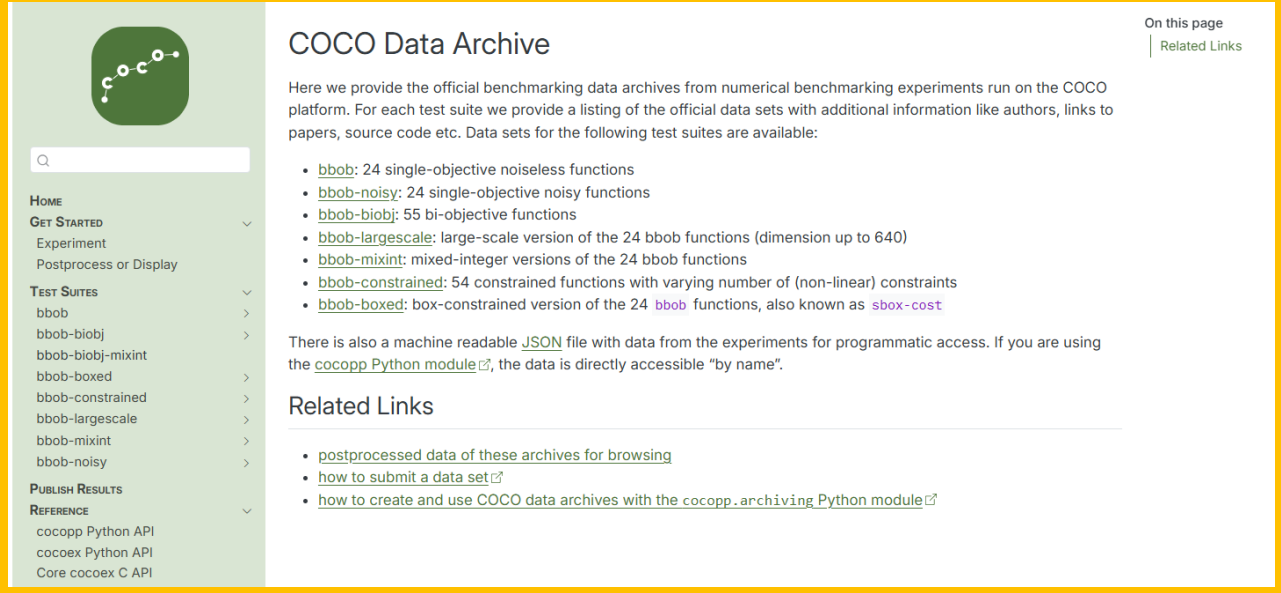
(or geometric avg. runtime) over all targets (difficult and easy) and all runs

Example


bbob f1-f24, 20-D
51 targets: 100..1e-08
29, 5, 15 instances
Fraction of function,target pairs
log10(# f-evals / dimension)
v2.6.3.13
best 2009
a-CMA-ES
adm-CMA-E
ad-CMA-ES
sdm-CMA-E
scdm-CMA-
default-C
cd-CMA-ES
d-CMA-ES
s-CMA-ES
c-CMA-ES
dm-CMA-ES
cdm-CMA-E
sc-CMA-ES
sd-CMA-ES
scd-CMA-E
cma-bp-bb
cma-bt-bb
DIRECT-RE
BIRMIN Ku
m-SDWOA E
I-DBDP-GL
E-WOA Esp
WOA Espin

Example

https://coco-platform.org/ppdata-archive/



COCO Data Archive

Here we provide the official benchmarking data archives from numerical benchmarking experiments run on the COCO platform. For each test suite we provide a listing of the official data sets with additional information like authors, links to papers, source code etc. Data sets for the following test suites are available:

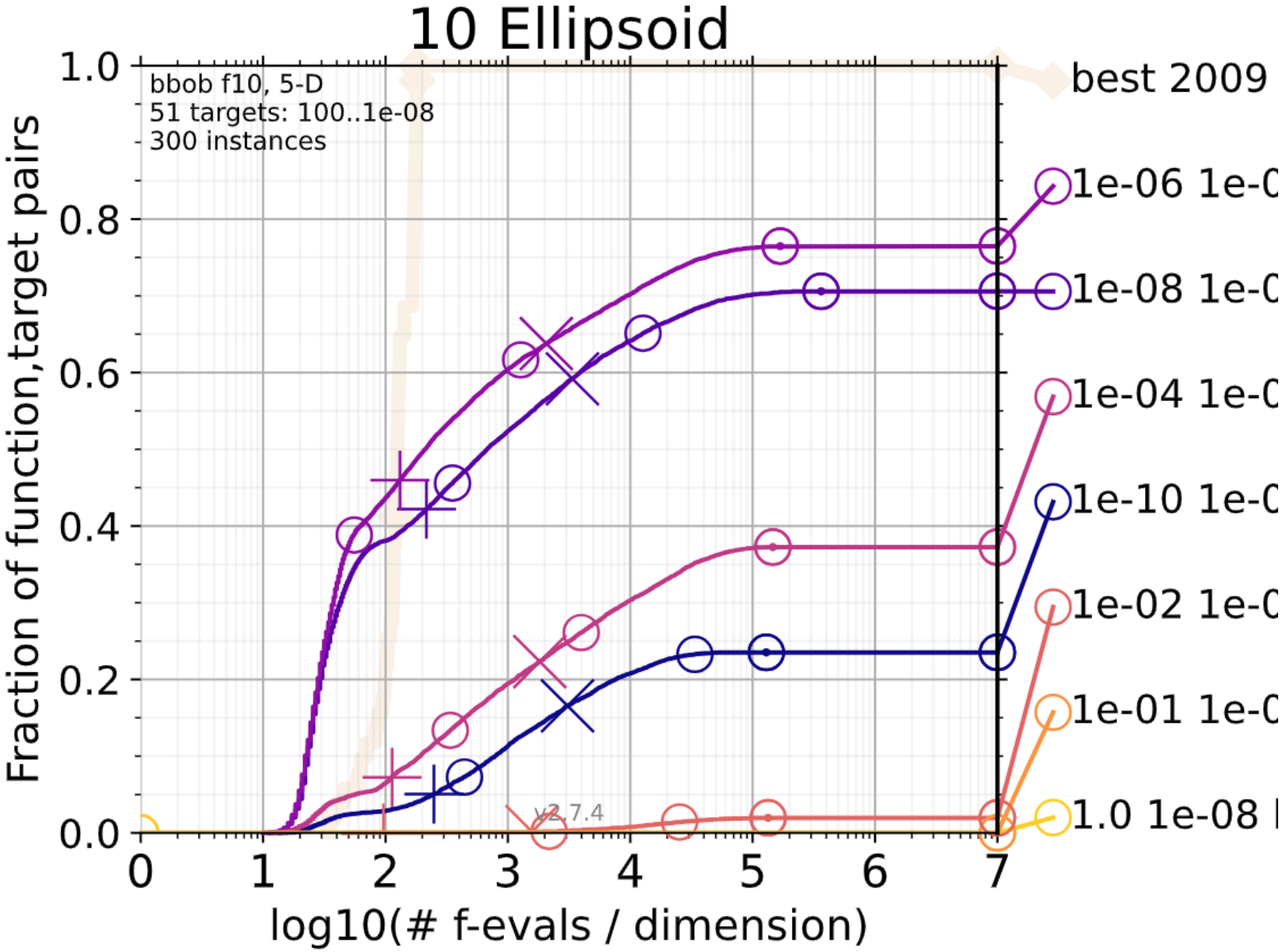
- bbob: 24 single-objective noiseless functions
- bbob-noisy: 24 single-objective noisy functions
- bbob-biobj: 55 bi-objective functions
- bbob-largescale: large-scale version of the 24 bbob functions (dimension up to 640)
- bbob-mixint: mixed-integer versions of the 24 bbob functions
- bbob-constrained: 54 constrained functions with varying number of (non-linear) constraints
- bbob-boxed: box-constrained version of the 24 `bbob` functions, also known as `sbox-cost`

There is also a machine readable JSON file with data from the experiments for programmatic access. If you are using the cocopp Python module, the data is directly accessible "by name".

Related Links

- postprocessed data of these archives for browsing
- how to submit a data set
- how to create and use COCO data archives with the `cocopp.archiving` Python module

Example w/ New `--parameter-sweep`



Available Test Suites in COCO

bbob (since 2009)	24 noiseless fcts	250+ data sets
bbob-noisy (since 2009)	30 noisy fcts	40+ data sets
bbob-biobj (since 2016)	55 bi-obj. fcts	39 data sets
bbob-largescale (since 2019)	24 noiseless fcts	17 data sets
bbob-mixint (since 2019)	24 noiseless fcts	8 data sets
bbob-biobj-mixint (s. 2019)	92 bi-objective fcts	-
bbob-constrained (s. 2022)	54 constrained fcts	9 data sets
bbob-boxed (s.2023)	24 box-constr. fcts	3 data sets

= former sbox-cost

https://coco-platform.org/data-archive/

Easy Data Access

```
pip install cocopp
python -m cocopp exdata/myfolder BIPOP BFGS
```

Easy Data Access

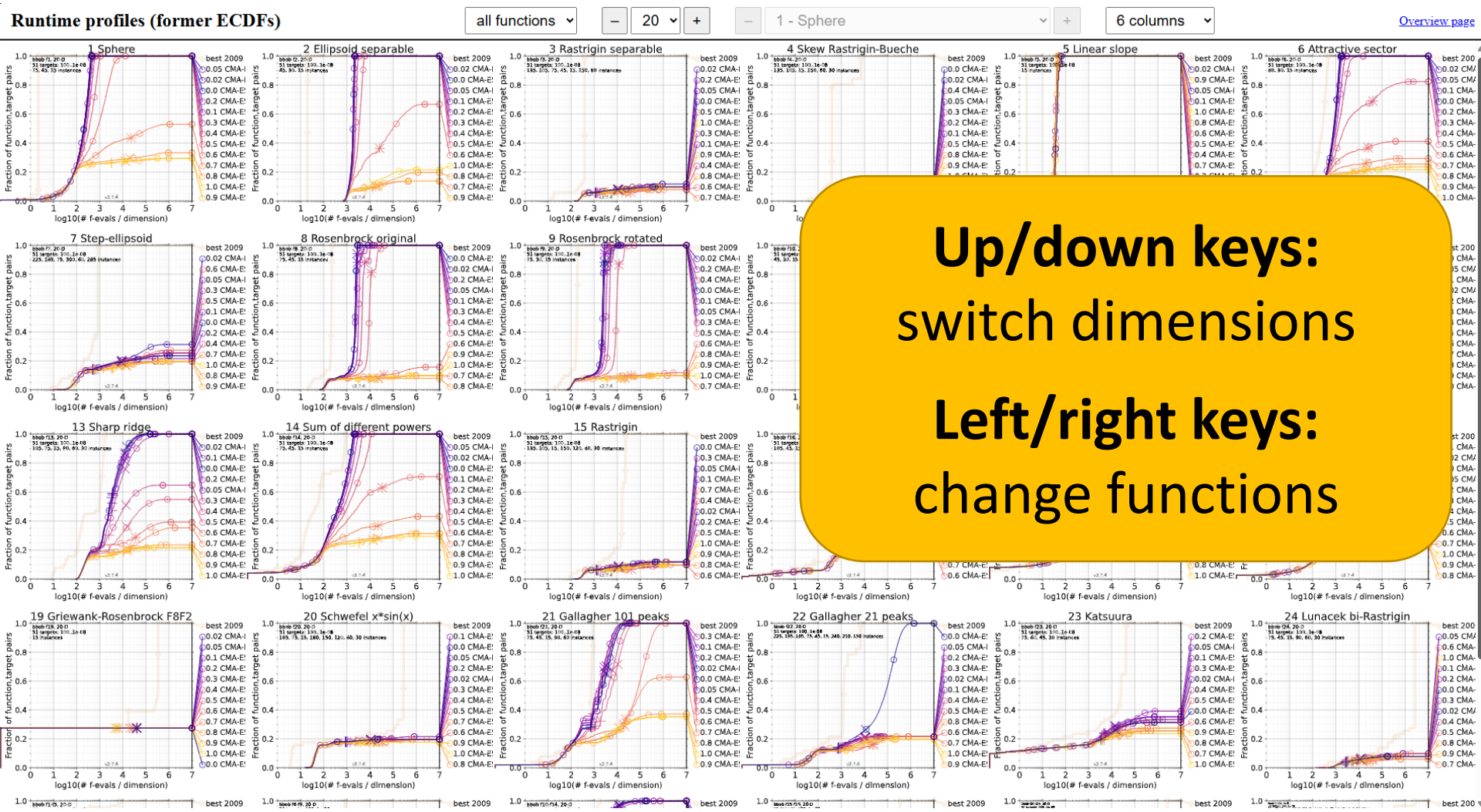
```
pip install cocopp
python -m cocopp exdata/myfolder BIPOP BFGS

[…]
ValueError: 'BIPOP' has multiple matches in the data archive:
    2009/BIPOP-CMA-ES_hansen_noiseless.tgz
    2012/BIPOPaCMA_loshchilov_noiseless.tgz
    […]
    2017/KL-BIPOP-CMA-ES-Yamaguchi.tgz
Either pick a single match, or use the `get_all` or `get_first` method,
or use the ! (first) or * (all) marker and try again.

python -m cocopp exdata/myfolder BIPOP! BFGS!
```

[data access of course also available within cocopp.main(…)]

News Since the Last Workshop in 2023

- New webpage: **`coco-platform.org`**
- **`pip install coco-experiment`** possible
- More interactive output (thanks to Tobias!)

News Since the Last Workshop in 2023



News Since the Last Workshop in 2023

- New webpage: **`coco-platform.org`**
- **`pip install coco-experiment`** possible
- More interactive output (thanks to Tobias!)
- Postprocessing allows **`--parameter sweep`**
- A new "noisifier" to wrap around COCO functions in Python
  - additive half-Cauchy distributed outliers
  - substractive half-Cauchy distributed outliers
  - additive Gaussian noise with varying variance
  - on p percent of the search space (frozen noise, $0 \leq p \leq 1$)
  - more in the coming talks

Overview of the Workshop

Session I: On the Impact of Outliers, Mixed-Integer Optimization

11:40 – 12:00 The BBOBies: Blackbox Optimization Benchmarking with COCO

12:00 – 12:20 **Alexandre Chotard:** On the Robustness of BFGS and Nelder-Mead to Positive and Negative Noise Outliers on the BBOB Test Suite

12:20 – 12:40 **Dimo Brockhoff:** Benchmarking Powell's Legacy: Performance of Five Derivative-Free Solvers in pdfo on the bbob Test Suite w/ and w/o Outliers

12:40 – 13:00 **Oskar Girardin:** Benchmarking CMA-ES under Additive and Subtractive Noise on the BBOB Testbed

13:00 – 13:20 **Duc-Manh Nguyen:** Benchmarking Improved Variants of CMA-ES-PDM on the bbob-mixint Testbed

13:20 – 13:30 The BBOBies: Session Wrap Up

Session II: An Algorithmic Jam Session

15:00 – 15:20 **Elena Raponi:** Cascading CMA-ES Instances for Generating Input-diverse Solution Batches

15:20 – 15:40 **Samuel Tebbet, George De Ath, Tinkle Chugh:** BEACON: Continuous Bi-objective Benchmark problems with Explicit Adjustable COrrelatioN control

15:40 – 16:00 **Jakub Kudela:** Benchmarking Seven Multi-objective Optimization Methods from the PlatEMO Platform on the bbob-biobj Test Suite

16:00 – 16:20 **Tobias Glasmachers:** Benchmarking the (1+1) Limited Memory Matrix Adaptation Evolution Strategy on the bbob-largescale Testbed

16:20 – 16:50 General Discussion