12th GECCO Workshop on Blackbox Optimization Benchmarking (BBOB): Welcome and Introduction to COCO/BBOB

The BBOBies
https://github.com/numbbo/coco

INVENTORS FOR THE DIGITAL WORLD

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

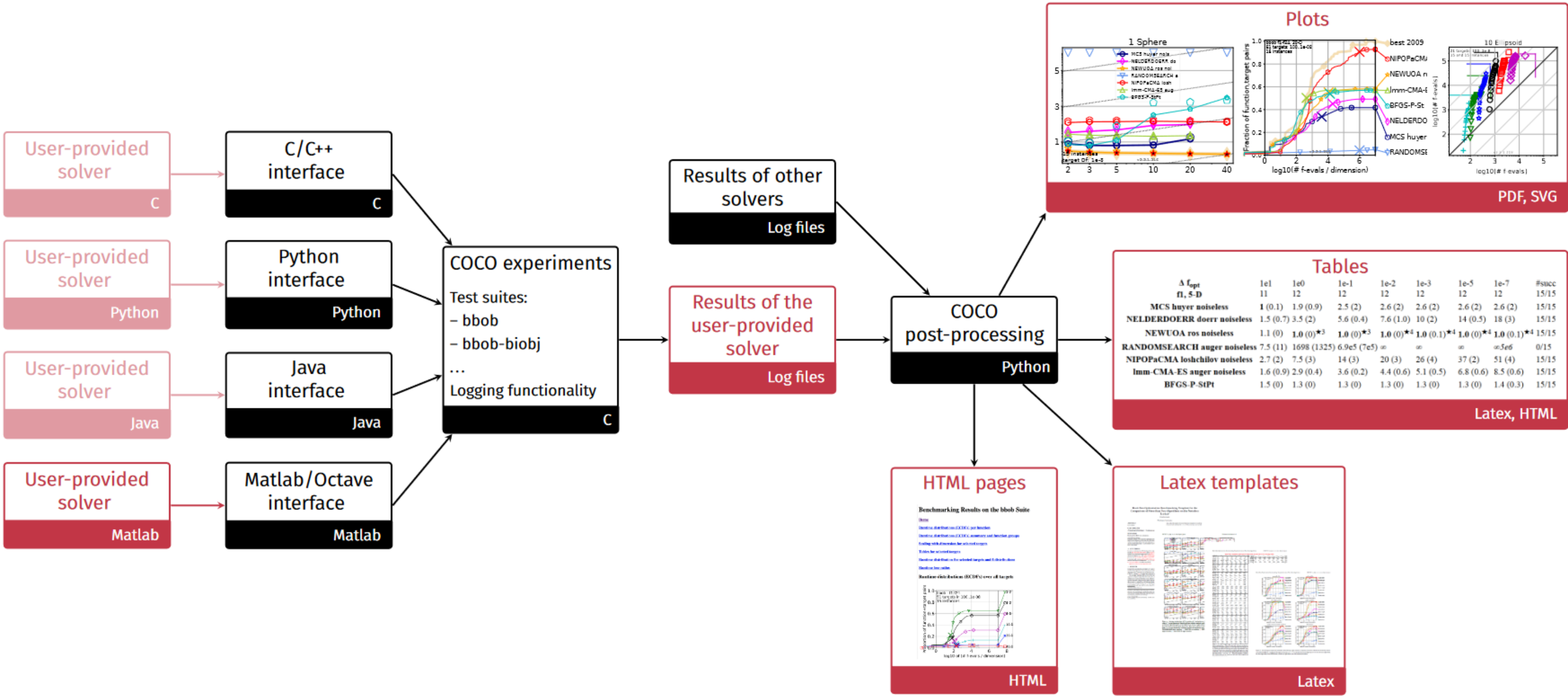
that's where COCO and BBOB come into play

Comparing Continuous Optimizers Platform

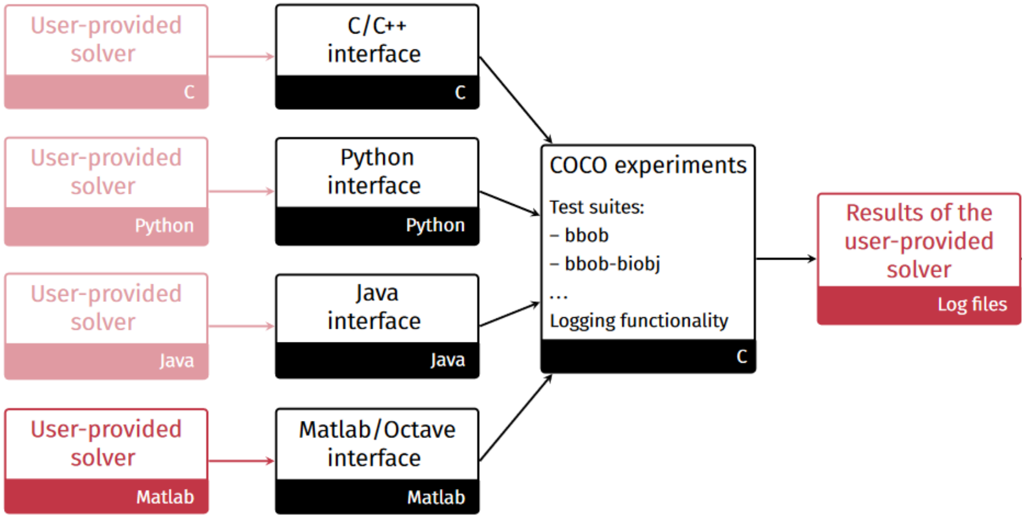
`https://github.com/numbbo/coco`

automatized benchmarking

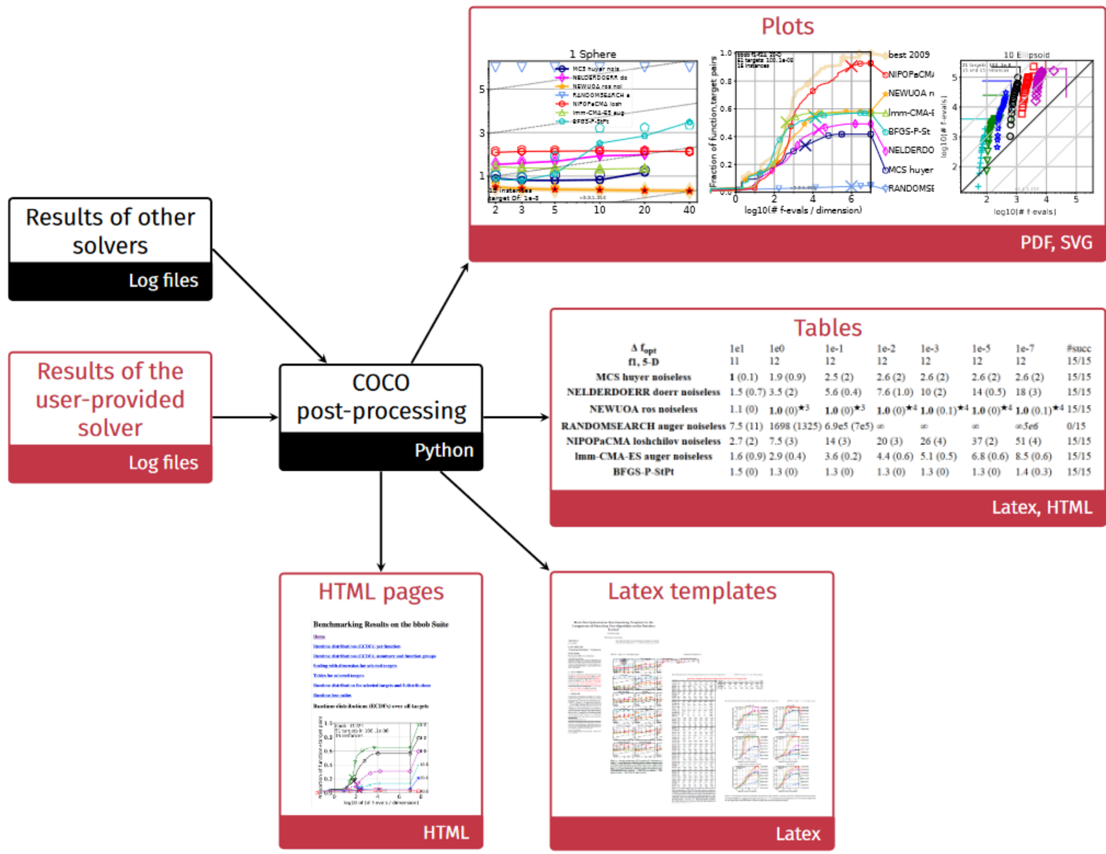
Overview of COCO's Structure



Overview of COCO's Structure



Overview of COCO's Structure



COCO implements a

reasonable, well-founded, and

well-documented

pre-chosen methodology

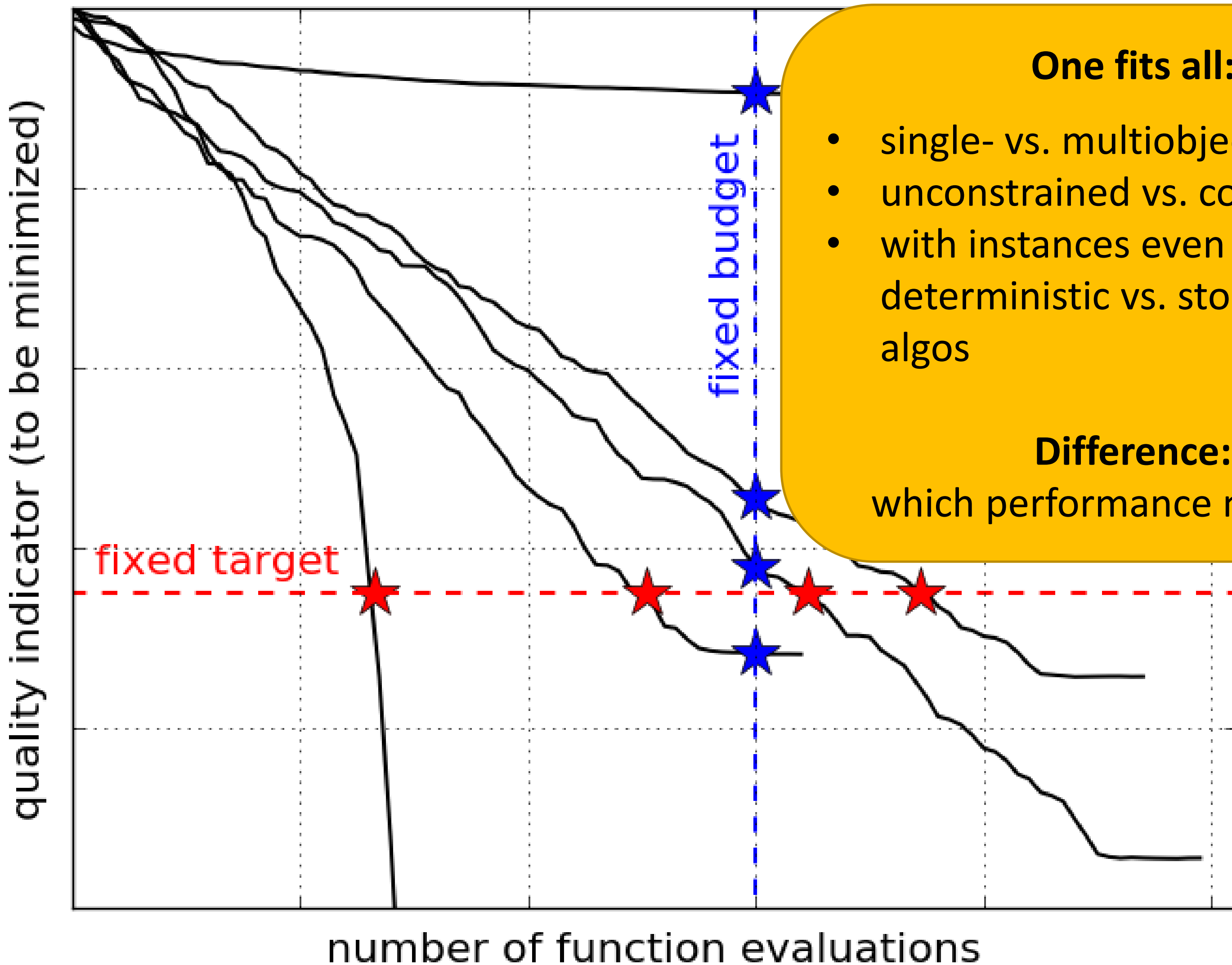
main performance measure:

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?

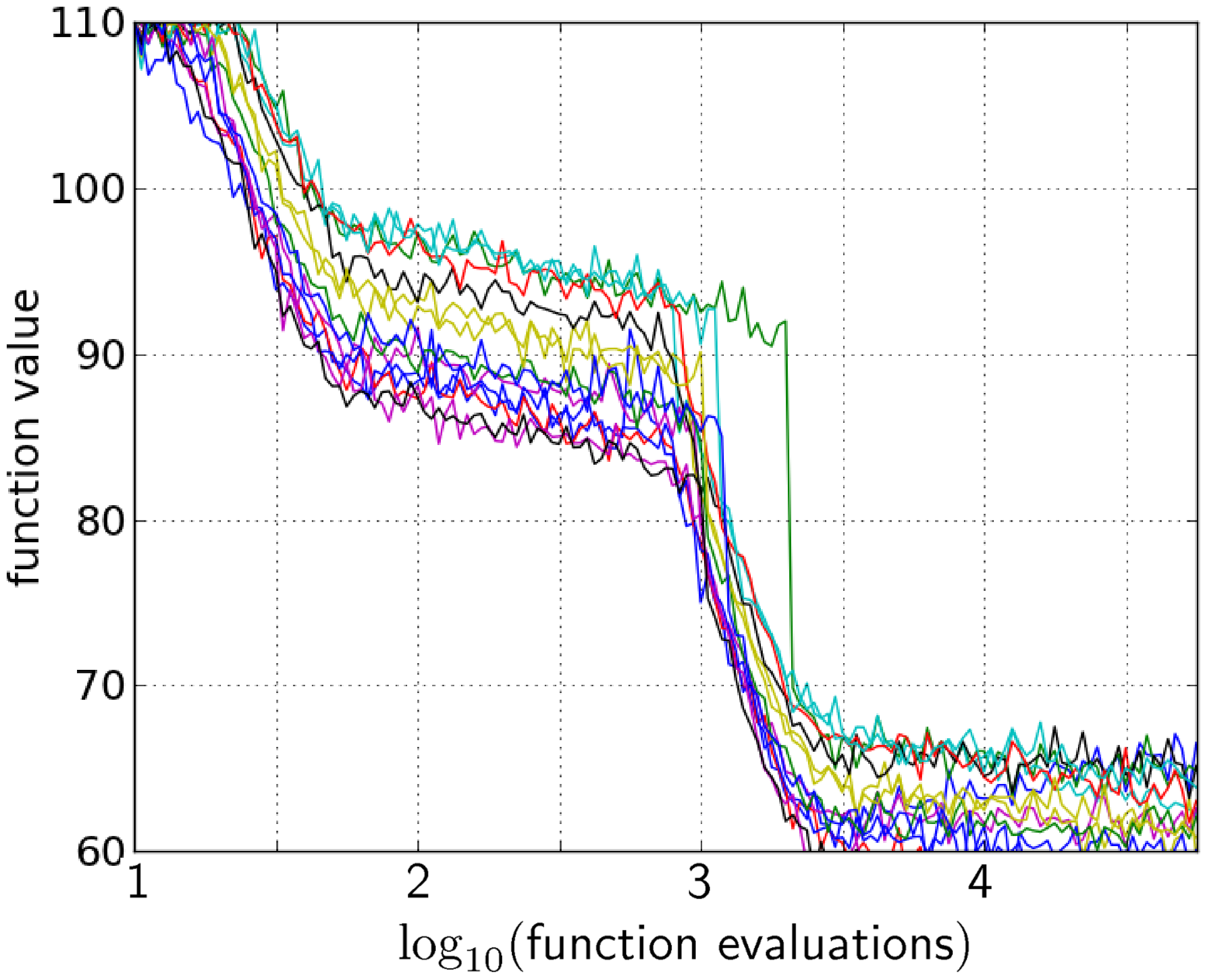
Main Performance Visualization:

Empirical Runtime Distributions

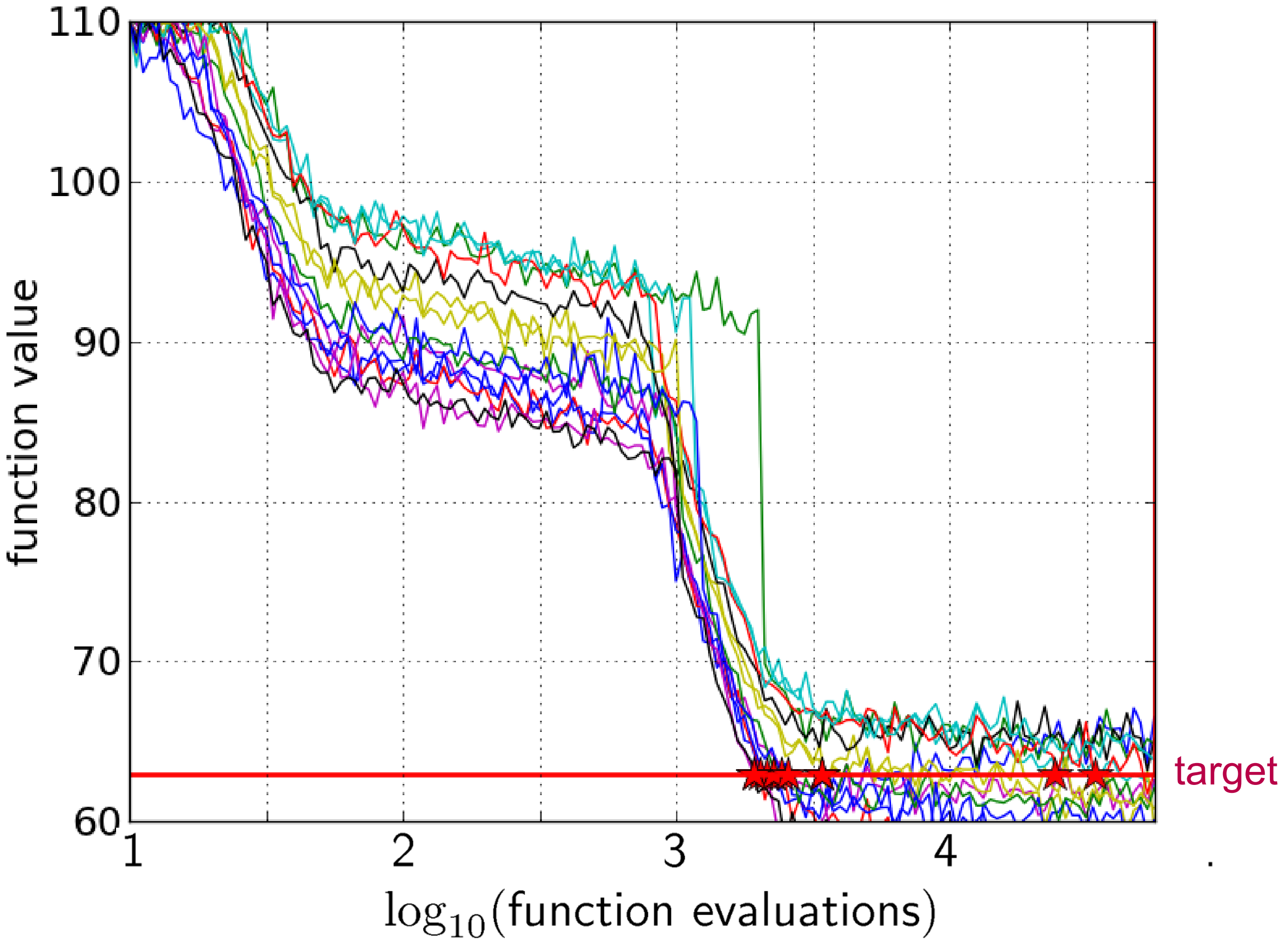
[aka Empirical Cumulative Distribution Function (ECDF) of the Runtime]

[aka data profile]

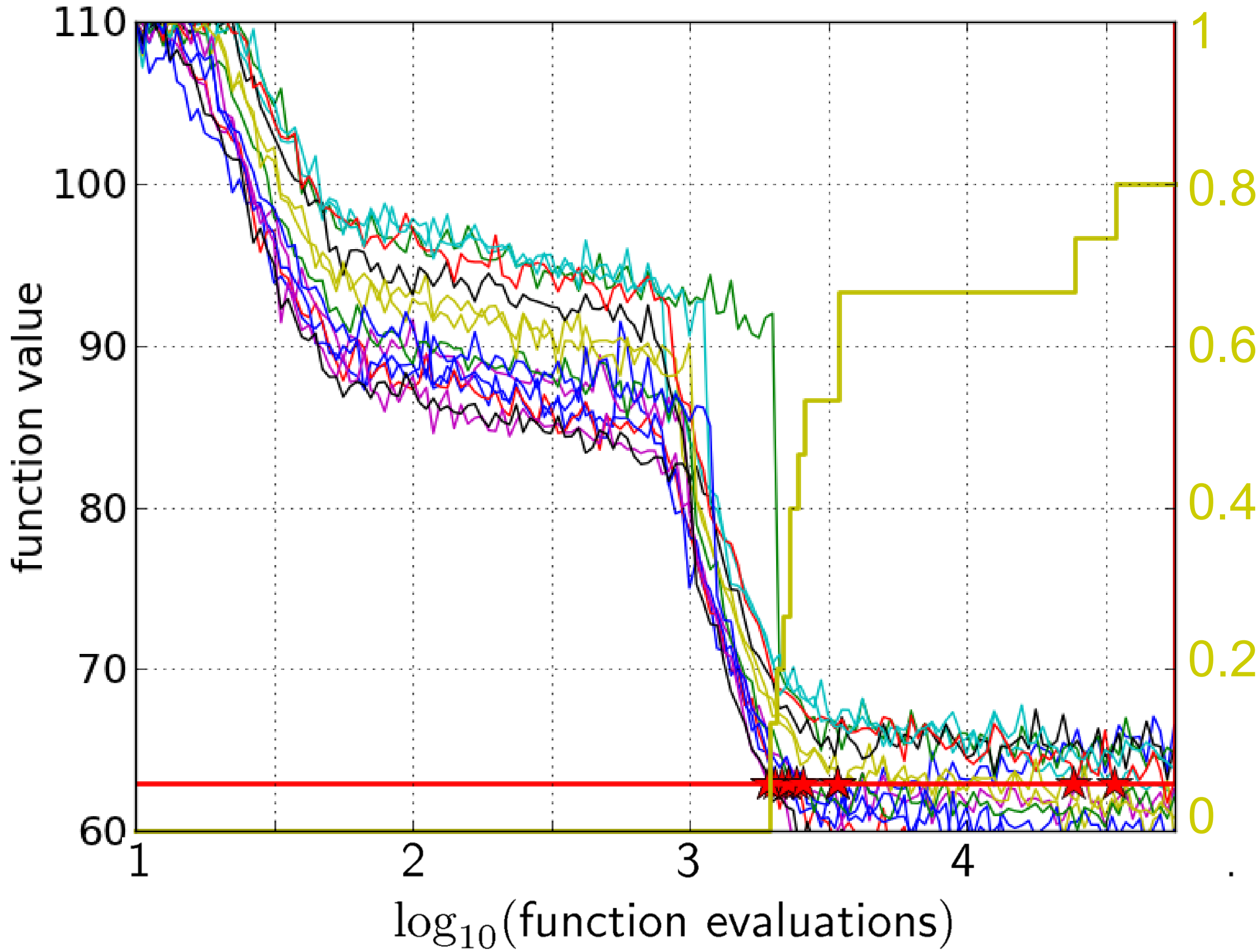
Convergence Graph of 15 Runs



15 Runs ≤ 15 Runtime Data Points



Empirical Cumulative Distribution



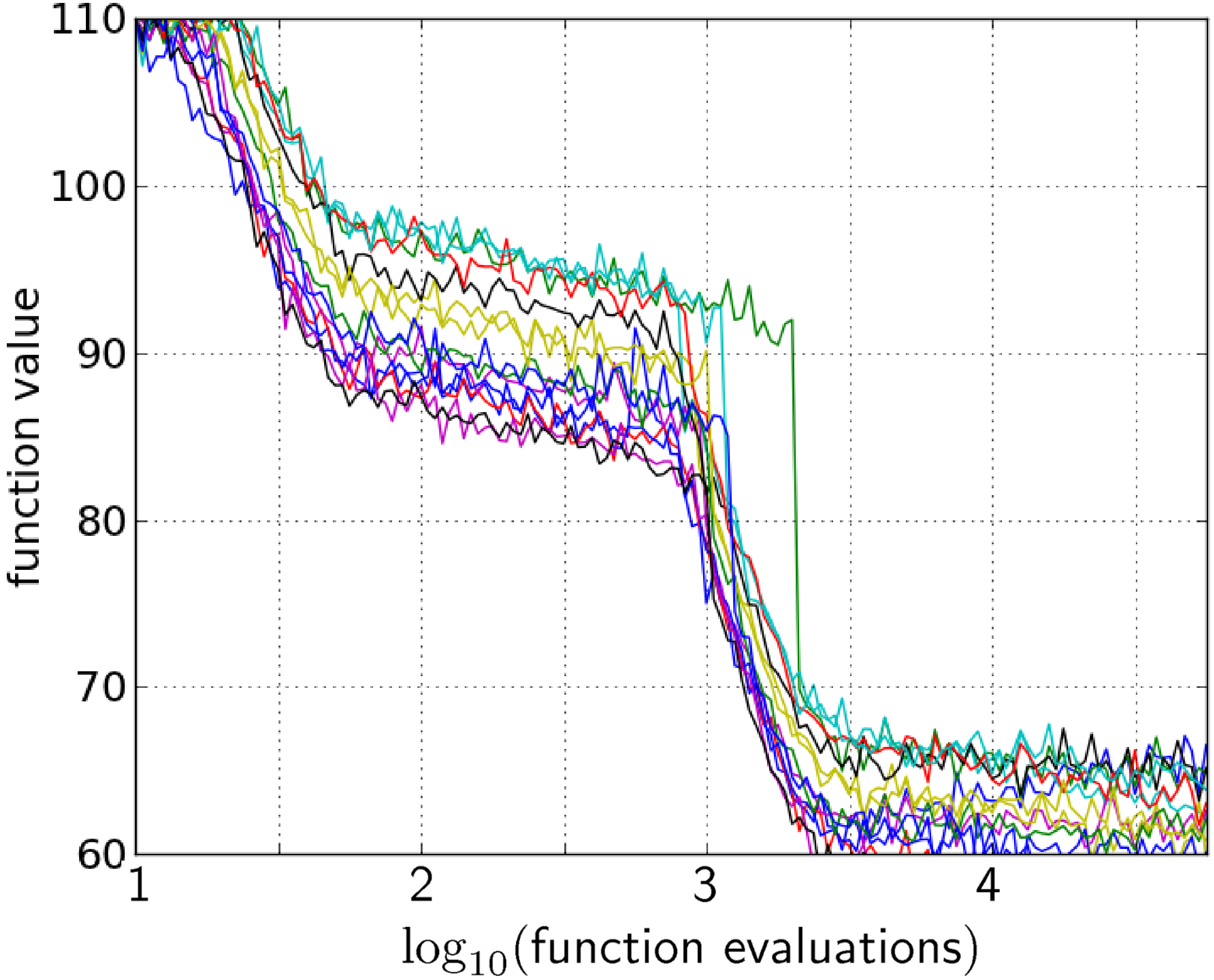
the ECDF of run lengths to reach the target

- 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

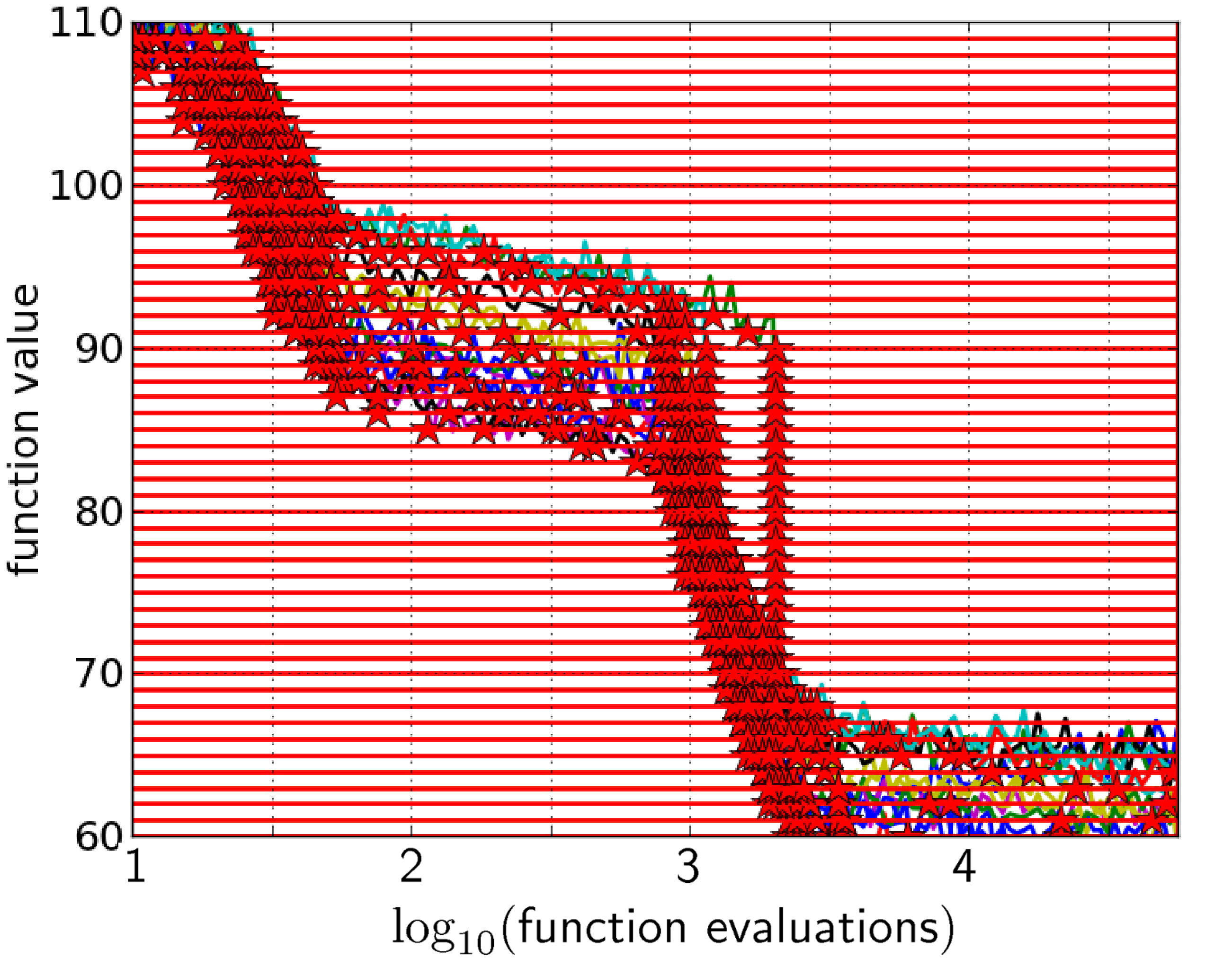
80% of the runs reached the target

Aggregation



15 runs

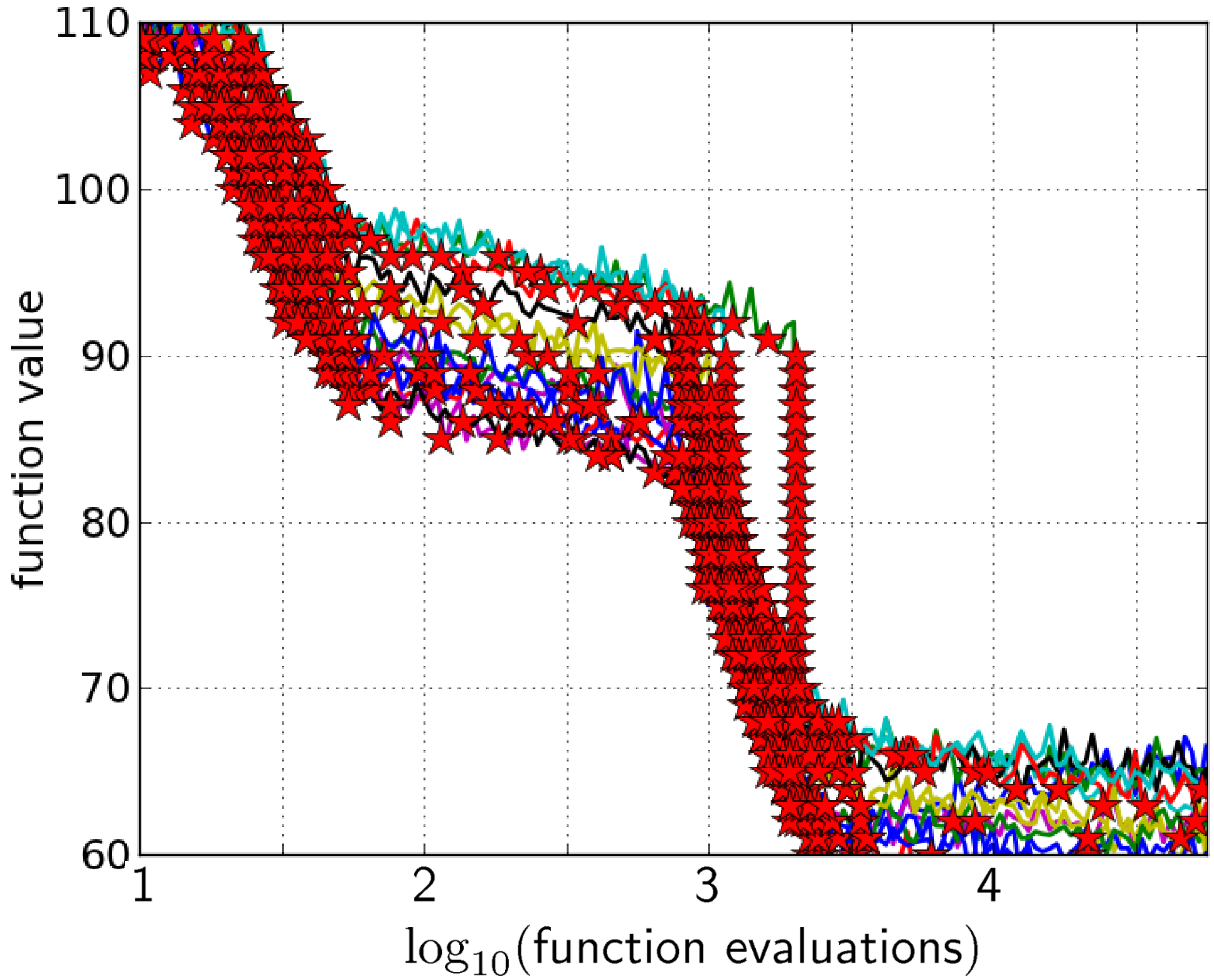
Aggregation



15 runs

50 targets

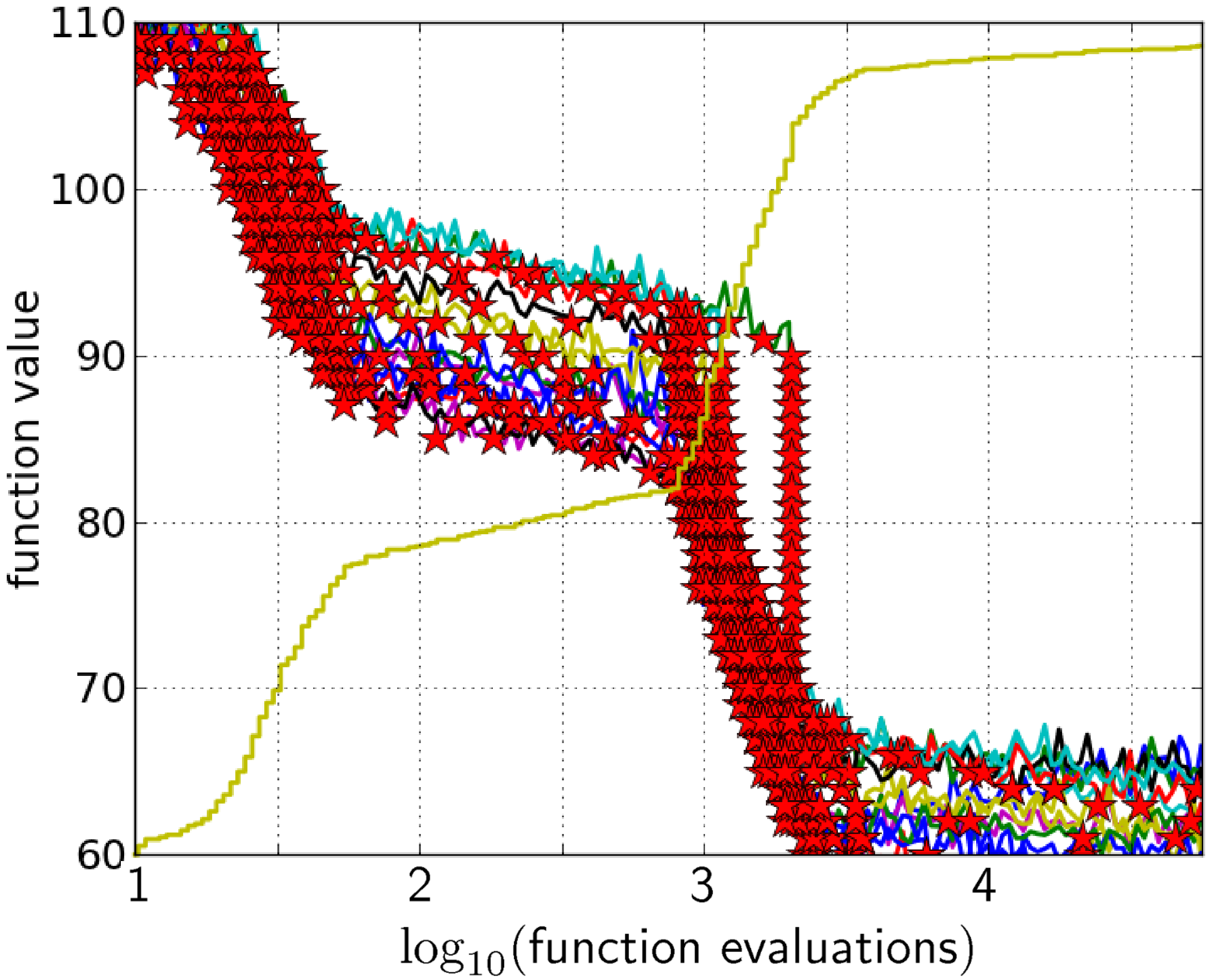
Aggregation



15 runs

50 targets

Aggregation

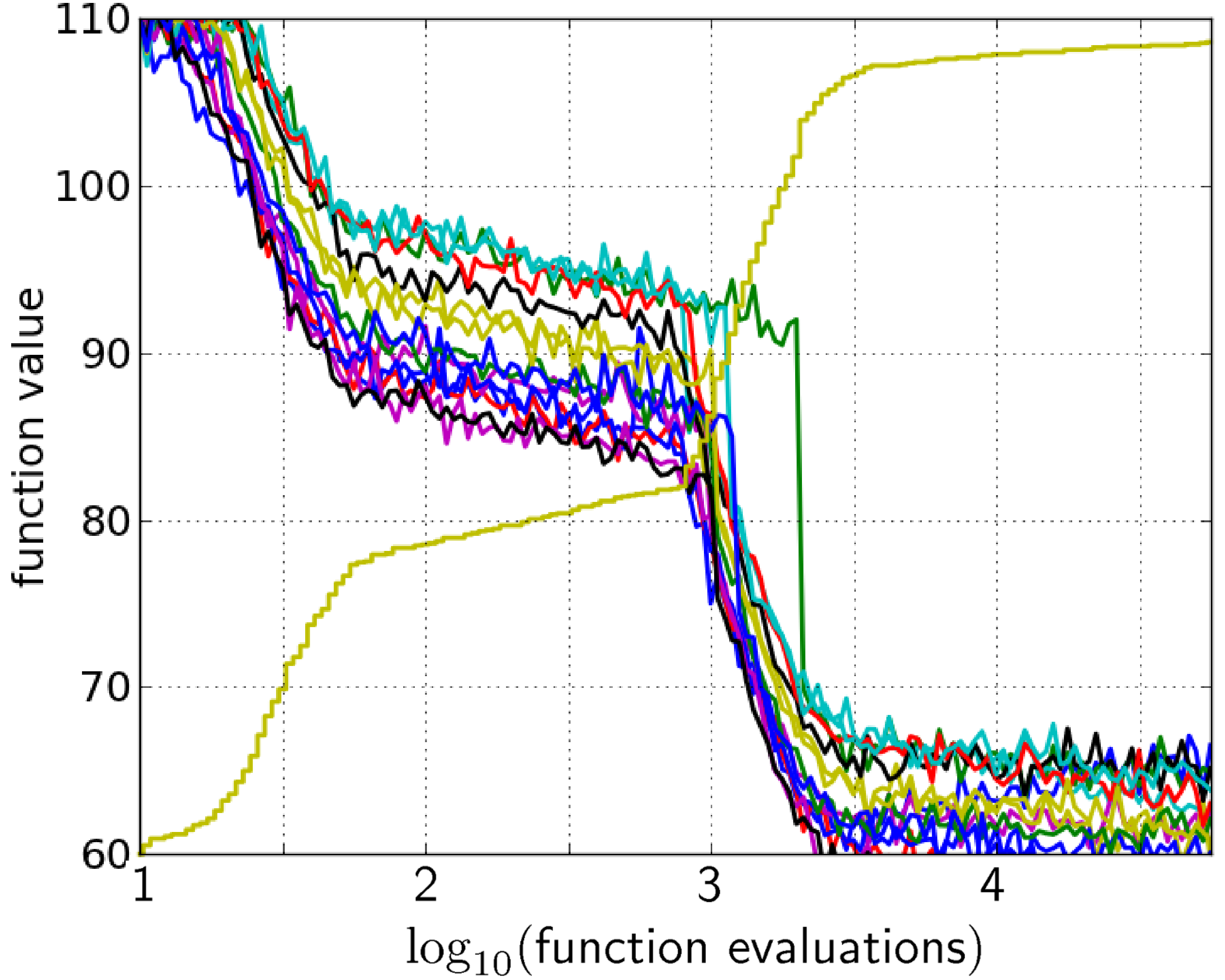


15 runs

50 targets

ECDF with 750 steps

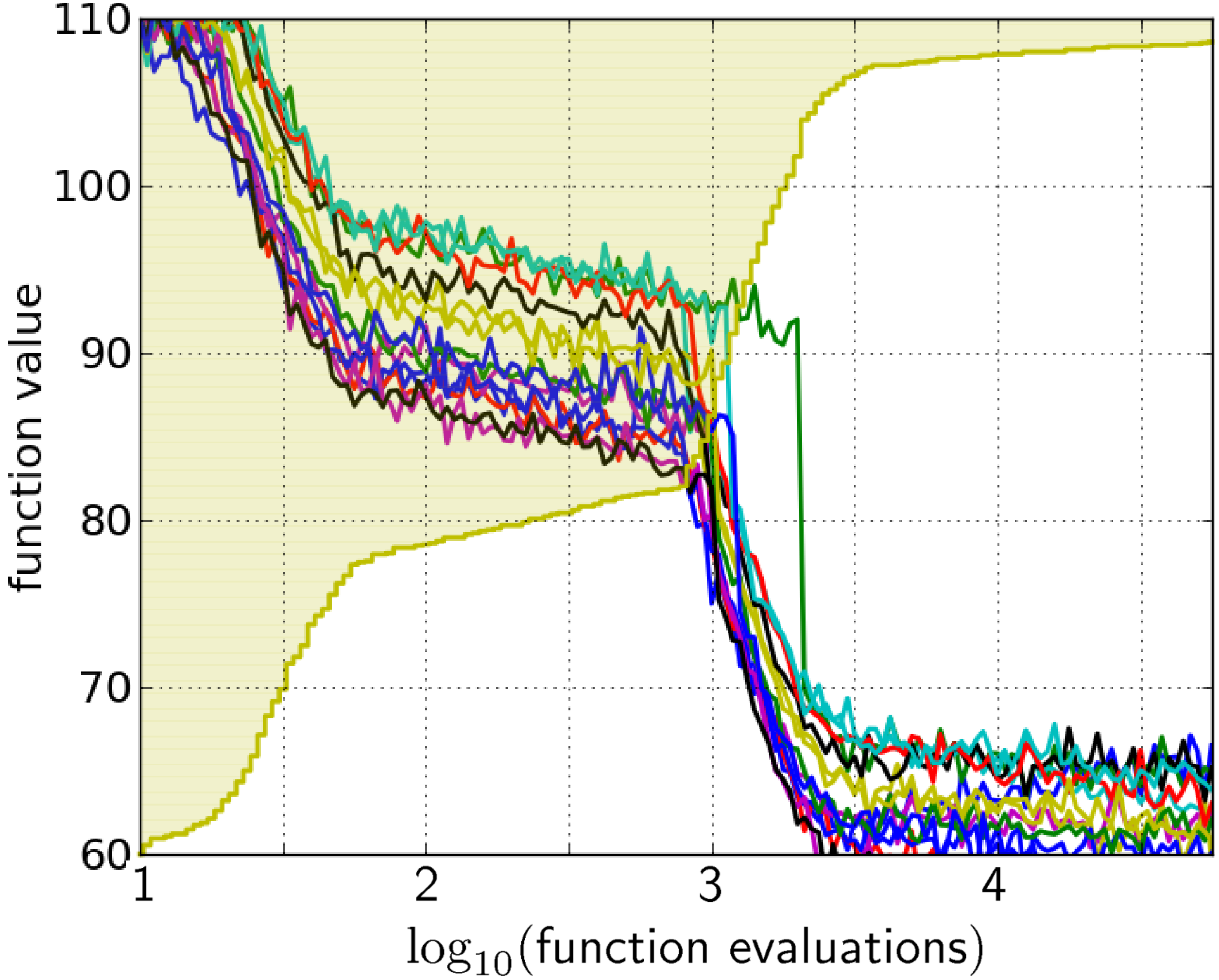
Aggregation



50 targets from 15 runs

...integrated in a single graph

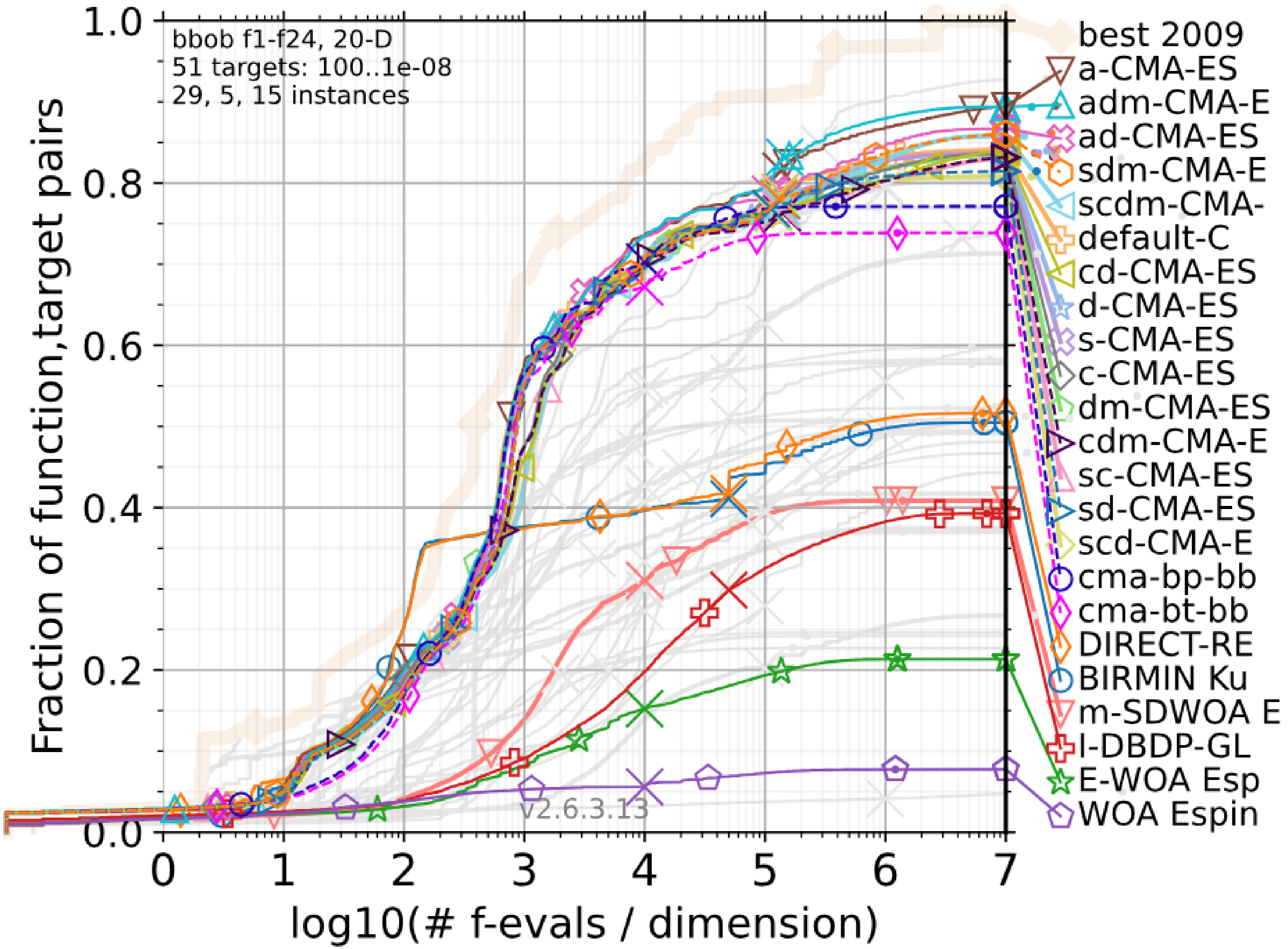
Interpretation



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
l-DBDP-GL
E-WOA Esp
WOA Espin

Example

https://numbbo.github.io/ppdata-archive/

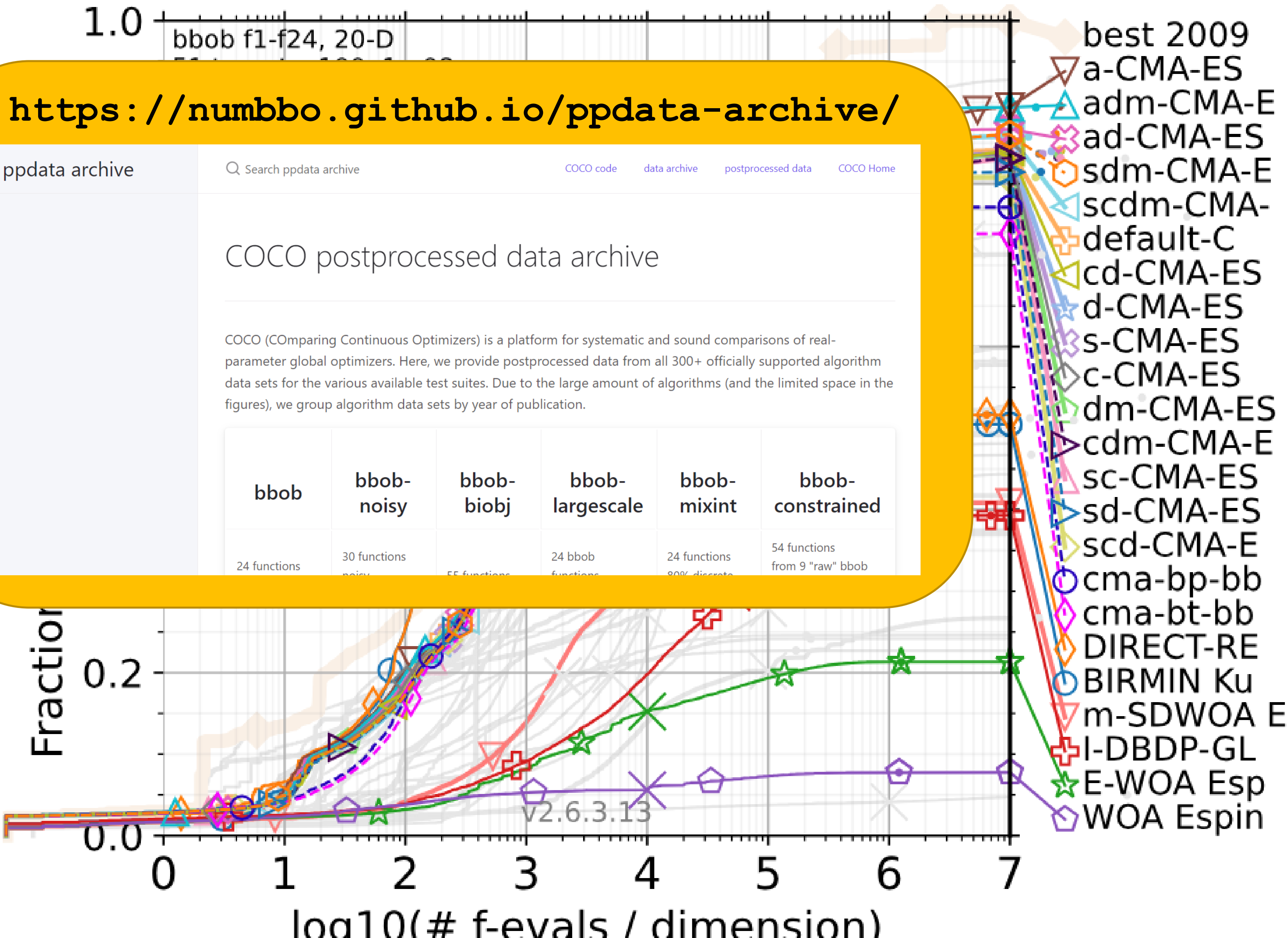
ppdata archive

Search ppdata archive | COCO code | data archive | postprocessed data | COCO Home

COCO postprocessed data archive

COCO (COmparing Continuous Optimizers) is a platform for systematic and sound comparisons of real-parameter global optimizers. Here, we provide postprocessed data from all 300+ officially supported algorithm data sets for the various available test suites. Due to the large amount of algorithms (and the limited space in the figures), we group algorithm data sets by year of publication.

bbob	bbob-noisy	bbob-biobj	bbob-largescale	bbob-mixint	bbob-constrained
24 functions	30 functions		24 bbob	24 functions	54 functions from 9 "raw" bbob



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	16 data sets
bbob-mixint (since 2019)	24 noiseless fcts	5 data sets
bbob-biobj-mixint (s. 2019)	92 bi-objective fcts	-
bbob-constrained (s. 2022)	54 constrained fcts	9 data sets
sbox-cost new	24 box-constr. fcts	2 data sets

https://numbbo.github.io/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(…)]

Session 1: Mixint & Multiobjective Opt.

11:40 – 12:15 The BBOBies: **"Introduction to BBOB"**

mix-int

12:15 – 12:40 Tristan Marty, Yann Semet, Anne Auger, Sébastien Héron, Nikolaus Hansen: **Benchmarking CMA-ES with Basic Integer Handling on a Mixed-Integer Test Problem Suite**

bbob-biobj

12:40 – 13:05 Dimo Brockhoff, Pascal Capetillo, Jonathan Hornewall, Raphael Walker: **Benchmarking the Borg algorithm on the Biobjective bbob-biobj Testbed**

13:05 – 13:30 Victoria Johnson, João Duro, Visakan Kadirkamanathan, Robin Purshouse: **A distributed multi-disciplinary design optimization benchmark test suite with constraints and multiple conflicting objectives**