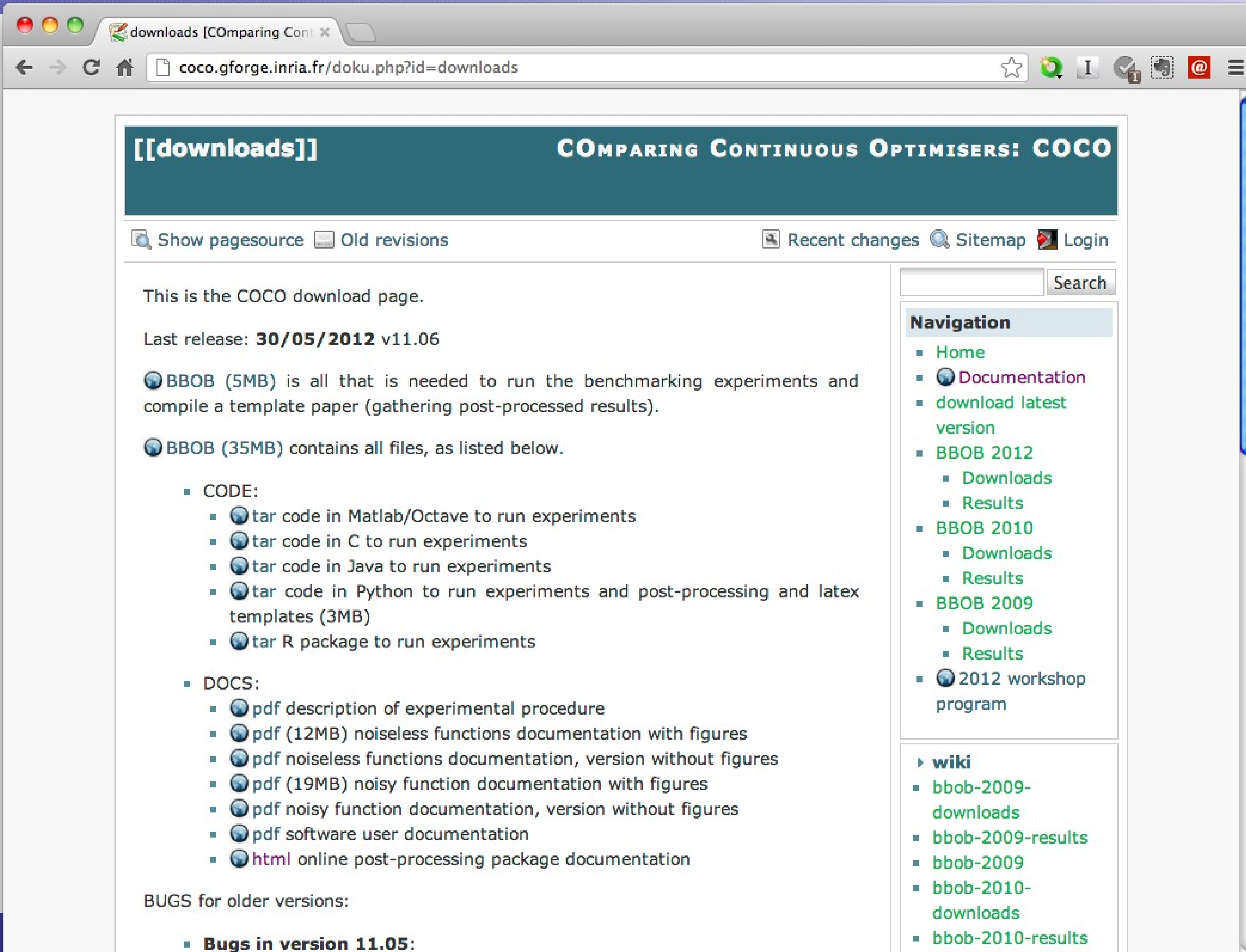
BBOB Black-Box Optimization Benchmarking with COCO (COmparing Continuous Optimizers)

BBOB in practice (for dummies)

BBOB in practice



downloads [COmparing Cont

coco.gforge.inria.fr/doku.php?id=downloads

[[downloads]] COmparing Continuous Optimisers: COCO

Show pagesource Old revisions Recent changes Sitemap Login

Search

This is the COCO download page.

Last release: **30/05/2012** v11.06

BBOB (5MB) is all that is needed to run the benchmarking experiments and compile a template paper (gathering post-processed results).

BBOB (35MB) contains all files, as listed below.

- CODE:
  - tar code in Matlab/Octave to run experiments
  - tar code in C to run experiments
  - tar code in Java to run experiments
  - tar code in Python to run experiments and post-processing and latex templates (3MB)
  - tar R package to run experiments
- DOCS:
  - pdf description of experimental procedure
  - pdf (12MB) noiseless functions documentation with figures
  - pdf noiseless functions documentation, version without figures
  - pdf (19MB) noisy function documentation with figures
  - pdf noisy function documentation, version without figures
  - pdf software user documentation
  - html online post-processing package documentation

BUGS for older versions:

- **Bugs in version 11.05**:

Navigation

- Home
- Documentation
- download latest version
- BBOB 2012
  - Downloads
  - Results
- BBOB 2010
  - Downloads
  - Results
- BBOB 2009
  - Downloads
  - Results
- 2012 workshop program

wiki

- bbob-2009-downloads
- bbob-2009-results
- bbob-2009
- bbob-2010-downloads
- bbob-2010-results

BBOB in practice

Name	Date Modified	Size	Kind
bbob.v11.06	Today, 1:15	--	Folder
c	May 30, 2012 12:07	--	Folder
docs	October 27, 2012 0:58	--	Folder
java	May 30, 2012 12:07	--	Folder
latextemplates	May 30, 2012 12:06	--	Folder
matlab	May 30, 2012 12:06	--	Folder
python	May 30, 2012 12:06	--	Folder
r	May 30, 2012 12:07	--	Folder

Macintosh HD ▸ Users ▸ hansen ▸ Downloads ▸ bbob.v11.06

BBOB in practice

Name	Date Modified	Size	Kind
▼ bbob.v11.06	Today, 1:15	--	Folder
▶ c	May 30, 2012 12:07	--	Folder
▶ docs	October 27, 2012 0:58	--	Folder
▶ java	May 30, 2012 12:07	--	Folder
▶ latextemplates	May 30, 2012 12:06	--	Folder
▼ matlab	May 30, 2012 12:06	--	Folder
benchmarkinfos.txt	February 9, 2009 16:29	4 KB	Gedit …ument
benchmarks.m	February 10, 2011 16:24	86 KB	Objec…rce File
benchmarksnoisy.m	February 10, 2011 16:24	102 KB	Objec…rce File
exampleexperiment.m	February 1, 2012 19:52	4 KB	Objec…rce File
exampletiming.m	March 7, 2012 14:40	4 KB	Objec…rce File
fgeneric.m	December 7, 2011 17:56	33 KB	Objec…rce File
LICENSE.txt	February 1, 2012 20:23	4 KB	Gedit …ument
MY_OPTIMIZER.m	February 14, 2011 19:30	4 KB	Objec…rce File
README.txt	May 19, 2011 10:47	4 KB	Gedit …ument
▶ python	May 30, 2012 12:06	--	Folder
▶ r	May 30, 2012 12:07	--	Folder

Macintosh HD ▸ Users ▸ hansen ▸ Downloads ▸ bbob.v11.06

BBOB in practice

Matlab script (`exampleexperiment.m`):

```
dimensions = [2, 3, 5, 10, 20, 40];  % small dimensions first, for CPU reasons
functions = benchmarks('FunctionIndices');  % or benchmarksnoisy(...)
instances = [1:5, 31:40];  % 15 function instances

for dim = dimensions
  for ifun = functions
    for iinstance = instances
      fgeneric('initialize', ifun, iinstance, datapath, opt);
      MY_OPTIMIZER('fgeneric', dim, fgeneric('ftarget'), eval(maxfunevals) - f
      disp(sprintf(['  f%d in %d-D, instance %d: FEs=%d with %d restarts, fbes
      fgeneric('finalize');
    end
    disp(['      date and time: ' num2str(clock, ' %.0f')]);
  end
  disp(sprintf('---- dimension %d-D done ----', dim));
end
```

BBOB in practice

Running the experiment at an OS shell:

```
$ nohup nice octave < exampleexperiment.m > output.txt &
$ less output.txt
```

```
GNU Octave, version 3.6.3
Copyright (C) 2012 John W. Eaton and others.
This is free software; see the source code for copying conditions.
[...]
Read http://www.octave.org/bugs.html to learn how to submit bug reports.

For information about changes from previous versions, type `news'.

  f1 in 2-D, instance 1: FEs=242, fbest-ftarget=-8.1485e-10, elapsed time [h]: 0.00
  f1 in 2-D, instance 2: FEs=278, fbest-ftarget=-6.0931e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 3: FEs=242, fbest-ftarget=-9.2281e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 4: FEs=302, fbest-ftarget=-4.5997e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 5: FEs=230, fbest-ftarget=-9.8350e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 6: FEs=284, fbest-ftarget=-7.0829e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 7: FEs=278, fbest-ftarget=-6.5999e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 8: FEs=272, fbest-ftarget=-8.7044e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 9: FEs=248, fbest-ftarget=-2.6316e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 10: FEs=302, fbest-ftarget=-4.6779e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 11: FEs=272, fbest-ftarget=-5.1499e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 12: FEs=260, fbest-ftarget=-8.8635e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 13: FEs=266, fbest-ftarget=-2.5484e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 14: FEs=218, fbest-ftarget=-9.9961e-09, elapsed time [h]: 0.00
  f1 in 2-D, instance 15: FEs=248, fbest-ftarget=-7.5842e-09, elapsed time [h]: 0.00
      date and time: 2013 3 29 19 59 26
  f2 in 2-D, instance 1: FEs=824, fbest-ftarget=-7.0206e-09, elapsed time [h]: 0.00
  f2 in 2-D, instance 2: FEs=572, fbest-ftarget=-9.2822e-09, elapsed time [h]: 0.00
[...]
```

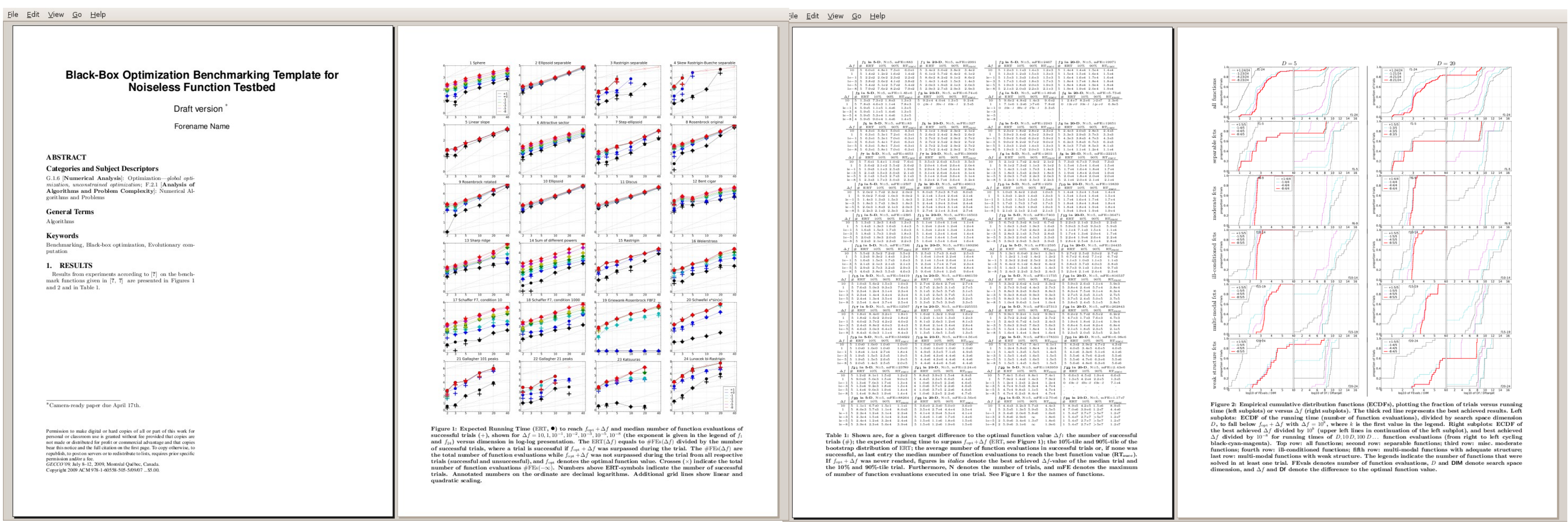
BBOB in practice

Name	Date Modified	Size	Kind
bbob.v13.05	March 8, 2013 13:04	--	Folder
c	March 5, 2013 23:04	--	Folder
docs	March 6, 2013 13:56	--	Folder
java	March 5, 2013 23:04	--	Folder
latextemplates	March 5, 2013 23:04	--	Folder
matlab	March 5, 2013 23:02	--	Folder
python	Today, 20:08	--	Folder
bbob_pproc	March 5, 2013 23:03	--	Folder
bbobbenchmarks.py	November 12, 2012 16:56	74 KB	Python script
benchmarkinfos.txt	February 9, 2009 16:29	4 KB	Gedit ...ument
exampleexperiment.py	February 22, 2013 14:26	4 KB	Python script
exampletiming.py	November 12, 2012 16:56	4 KB	Python script
fgeneric.py	March 3, 2013 19:33	25 KB	Python script
LICENSE.txt	February 1, 2012 20:23	4 KB	Gedit ...ument
README.txt	November 12, 2012 16:56	4 KB	Gedit ...ument
r	March 5, 2013 23:05	--	Folder

Macintosh HD ▸ Users ▸ hansen ▸ Downloads ▸ bbob.v13.05 ▸ python ▸ bbob_pproc

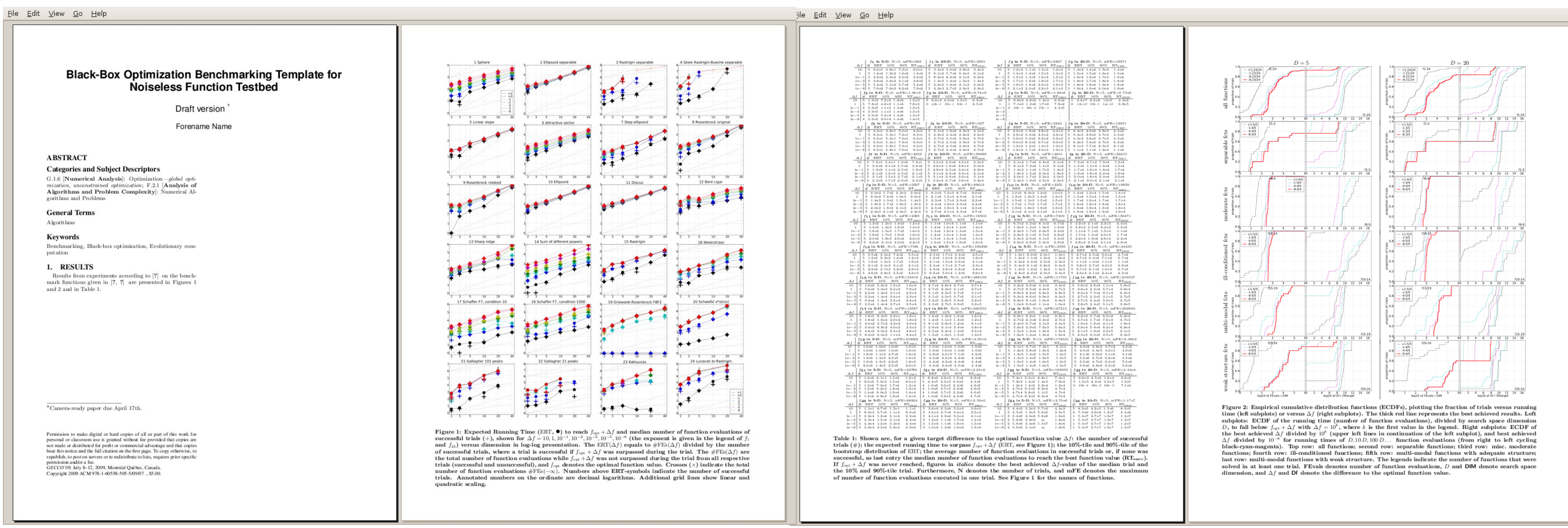
Post-processing at the OS shell:

```
$ python codepath/bbob_pproc/rungeneric.py datapath
[...]
$ pdflatex templateACMarticle.tex
[...]
```



Post-processing at the OS shell:

```
$ python codepath/bbob_pproc/rungeneric.py datapath
[...]
$ pdflatex templateACMarticle.tex
[...]
```



e Edit View Go Help

Black-Box Optimization Benchmarking Template for Noiseless Function Testbed

Draft version *

Forename Name

ABSTRACT

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Benchmarking, Black-box optimization, Evolutionary computation

1. RESULTS

Results from experiments according to [?] on the benchmark functions given in [?, ?] are presented in Figures 1 and 2 and in Table 1.

*Camera-ready paper due April 17th.

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

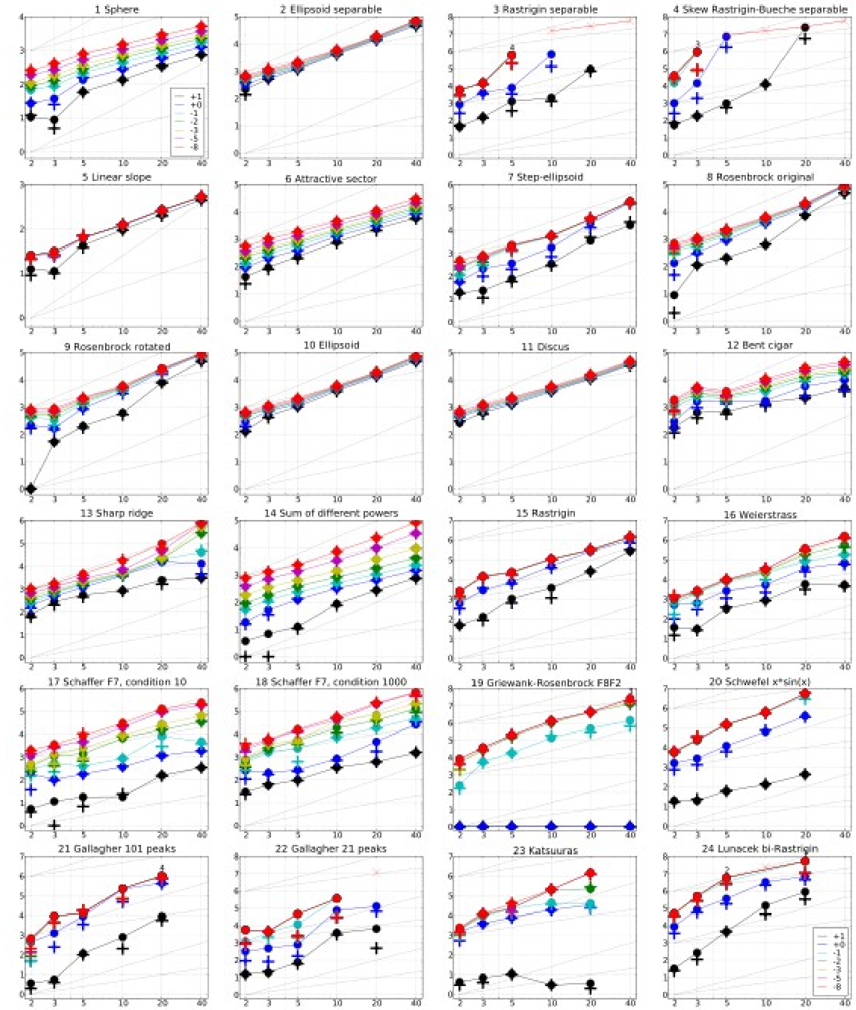


Figure 1: Expected Running Time (ERT, ●) to reach $f_{\rm opt}+\Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation. The $\mathrm{ERT}(\Delta f)$ equals to $\#\mathrm{FEs}(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\rm opt}+\Delta f$ was surpassed during the trial. The $\#\mathrm{FEs}(\Delta f)$ are the total number of function evaluations while $f_{\rm opt}+\Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and $f_{\rm opt}$ denotes the optimal function value. Crosses (×) indicate the total number of function evaluations $\#\mathrm{FEs}(-\infty)$. Numbers above ERT-symbols indicate the number of successful trials. Annotated numbers on the ordinate are decimal logarithms. Additional grid lines show linear and quadratic scaling.

Page 4 of 6

File Edit View Go Help

Δf	f_1 in 5-D, N=5, mFE=883 #	ERT	10%	90%	RT_{succ}	f_1 in 20-D, N=5, mFE=2991 #	ERT	10%	90%	RT_{succ}
10	5	6.0e1	4.8e1	7.3e1	6.0e1	5	3.4e2	3.0e2	3.8e2	3.4e2
1	5	1.4e2	1.2e2	1.6e2	1.4e2	5	6.1e2	5.7e2	6.4e2	6.1e2
1e−1	5	2.2e2	2.0e2	2.3e2	2.2e2	5	8.6e2	8.2e2	9.1e2	8.6e2
1e−3	5	3.8e2	3.6e2	4.1e2	3.8e2	5	1.4e3	1.4e3	1.5e3	1.4e3
1e−5	5	5.4e2	5.1e2	5.7e2	5.4e2	5	2.0e3	1.9e3	2.1e3	2.0e3
1e−8	5	7.9e2	7.6e2	8.2e2	7.9e2	5	2.9e3	2.7e3	2.9e3	2.9e3

Δf	f_2 in 5-D, N=5, mFE=2467 #	ERT	10%	90%	RT_{succ}	f_2 in 20-D, N=5, mFE=19971 #	ERT	10%	90%	RT_{succ}
10	5	1.2e3	1.1e3	1.4e3	1.2e3	5	1.4e4	1.4e4	1.5e4	1.4e4
1	5	1.3e3	1.2e3	1.5e3	1.3e3	5	1.5e4	1.5e4	1.6e4	1.5e4
1e−1	5	1.5e3	1.3e3	1.6e3	1.5e3	5	1.6e4	1.6e4	1.7e4	1.6e4
1e−3	5	1.7e3	1.6e3	1.8e3	1.7e3	5	1.8e4	1.7e4	1.8e4	1.8e4
1e−5	5	1.9e3	1.8e3	2.0e3	1.9e3	5	1.8e4	1.8e4	1.9e4	1.8e4
1e−8	5	2.1e3	2.0e3	2.2e3	2.1e3	5	1.9e4	1.9e4	2.0e4	1.9e4

Δf	f_3 in 5-D, N=5, mFE=1.83e6 #	ERT	10%	90%	RT_{succ}	f_3 in 20-D, N=5, mFE=6.74e6 #	ERT	10%	90%	RT_{succ}
10	5	1.3e3	7.3e2	1.8e3	1.3e3	5	9.2e4	4.0e4	1.3e5	9.2e4
1	5	7.8e3	4.6e3	1.1e4	7.8e3	0	*40e−1*	*30e−1*	*60e−1*	2.5e5
1e−1	4	5.9e5	1.1e5	1.4e6	1.3e5	.	.	.	.	.
1e−3	4	5.9e5	1.1e5	1.4e6	1.3e5	.	.	.	.	.
1e−5	4	5.9e5	5.3e4	1.4e6	1.3e5	.	.	.	.	.
1e−8	4	5.9e5	9.0e4	1.4e6	1.4e5	.	.	.	.	.

Δf	f_4 in 5-D, N=5, mFE=1.89e6 #	ERT	10%	90%	RT_{succ}	f_4 in 20-D, N=5, mFE=5.73e6 #	ERT	10%	90%	RT_{succ}
10	5	9.6e2	4.8e2	1.4e3	9.6e2	1	2.4e7	8.2e6	>2e7	2.3e6
1	1	7.1e6	1.2e6	>7e6	7.8e2	0	*13e+0*	*99e−1*	*14e+0*	6.8e5
1e−1	0	*29e−1*	*09e−2*	*27e−1*	3.3e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_5 in 5-D, N=5, mFE=83 #	ERT	10%	90%	RT_{succ}	f_5 in 20-D, N=5, mFE=327 #	ERT	10%	90%	RT_{succ}
10	5	4.3e1	3.6e1	5.0e1	4.3e1	5	2.1e2	1.9e2	2.3e2	2.1e2
1	5	6.3e1	5.3e1	7.2e1	6.3e1	5	2.6e2	2.4e2	2.8e2	2.6e2
1e−1	5	6.3e1	5.3e1	7.0e1	6.3e1	5	2.7e2	2.5e2	2.9e2	2.7e2
1e−3	5	6.3e1	5.4e1	7.2e1	6.3e1	5	2.7e2	2.5e2	2.9e2	2.7e2
1e−5	5	6.3e1	5.8e1	7.3e1	6.3e1	5	2.7e2	2.5e2	2.9e2	2.7e2
1e−8	5	6.3e1	5.8e1	7.0e1	6.3e1	5	2.7e2	2.4e2	2.9e2	2.7e2

Δf	f_6 in 5-D, N=5, mFE=2243 #	ERT	10%	90%	RT_{succ}	f_6 in 20-D, N=5, mFE=12651 #	ERT	10%	90%	RT_{succ}
10	5	2.3e2	1.8e2	2.8e2	2.3e2	5	2.4e3	2.0e3	2.8e3	2.4e3
1	5	3.9e2	3.4e2	4.3e2	3.9e2	5	3.3e3	2.9e3	3.7e3	3.3e3
1e−1	5	5.9e2	5.6e2	6.2e2	5.9e2	5	4.3e3	3.8e3	4.7e3	4.3e3
1e−3	5	9.0e2	8.2e2	9.7e2	9.0e2	5	6.2e3	5.8e3	6.7e3	6.2e3
1e−5	5	1.3e3	1.2e3	1.4e3	1.3e3	5	8.1e3	7.7e3	8.5e3	8.1e3
1e−8	5	1.9e3	1.7e3	2.0e3	1.9e3	5	1.1e4	1.1e4	1.2e4	1.1e4

Δf	f_7 in 5-D, N=5, mFE=4653 #	ERT	10%	90%	RT_{succ}	f_7 in 20-D, N=5, mFE=39069 #	ERT	10%	90%	RT_{succ}
10	5	7.6e1	5.4e1	1.0e2	7.6e1	5	3.5e3	2.6e3	4.5e3	3.5e3
1	5	3.6e2	2.1e2	5.5e2	3.6e2	5	2.0e4	1.6e4	2.6e4	2.0e4
1e−1	5	1.9e3	1.5e3	2.5e3	1.9e3	5	2.9e4	2.5e4	3.4e4	2.9e4
1e−3	5	2.1e3	1.5e3	3.0e3	2.1e3	5	3.1e4	2.6e4	3.4e4	3.1e4
1e−5	5	2.1e3	1.5e3	2.7e3	2.1e3	5	3.1e4	2.6e4	3.6e4	3.1e4
1e−8	5	2.3e3	1.7e3	3.0e3	2.3e3	5	3.2e4	2.7e4	3.6e4	3.2e4

Δf	f_8 in 5-D, N=5, mFE=2611 #	ERT	10%	90%	RT_{succ}	f_8 in 20-D, N=5, mFE=22215 #	ERT	10%	90%	RT_{succ}
10	5	2.1e2	1.7e2	2.4e2	2.1e2	5	7.3e3	6.7e3	7.9e3	7.3e3
1	5	9.1e2	7.3e2	1.1e3	9.1e2	5	1.5e4	1.5e4	1.6e4	1.5e4
1e−1	5	1.4e3	1.1e3	1.7e3	1.4e3	5	1.7e4	1.6e4	1.8e4	1.7e4
1e−3	5	1.8e3	1.5e3	2.0e3	1.8e3	5	1.9e4	1.8e4	2.0e4	1.9e4
1e−5	5	2.0e3	1.7e3	2.2e3	2.0e3	5	2.0e4	1.8e4	2.0e4	2.0e4
1e−8	5	2.2e3	1.9e3	2.5e3	2.2e3	5	2.1e4	2.0e4	2.1e4	2.1e4

Δf	f_9 in 5-D, N=5, mFE=2507 #	ERT	10%	90%	RT_{succ}	f_9 in 20-D, N=5, mFE=49613 #	ERT	10%	90%	RT_{succ}
10	5	2.0e2	1.7e2	2.3e2	2.0e2	5	8.0e3	7.3e3	8.7e3	8.0e3
1	5	9.0e2	7.6e2	1.0e3	9.0e2	5	2.1e4	1.5e4	2.6e4	2.1e4
1e−1	5	1.4e3	1.3e3	1.5e3	1.4e3	5	2.3e4	1.7e4	2.9e4	2.3e4
1e−3	5	1.8e3	1.7e3	1.9e3	1.8e3	5	2.4e4	1.9e4	3.0e4	2.4e4
1e−5	5	2.0e3	1.8e3	2.1e3	2.0e3	5	2.5e4	1.9e4	3.1e4	2.5e4
1e−8	5	2.2e3	2.1e3	2.3e3	2.2e3	5	2.7e4	2.1e4	3.3e4	2.7e4

Δf	f_{10} in 5-D, N=5, mFE=2251 #	ERT	10%	90%	RT_{succ}	f_{10} in 20-D, N=5, mFE=19839 #	ERT	10%	90%	RT_{succ}
10	5	1.0e3	8.4e2	1.2e3	1.0e3	5	1.4e4	1.3e4	1.5e4	1.4e4
1	5	1.3e3	1.2e3	1.4e3	1.3e3	5	1.5e4	1.5e4	1.6e4	1.5e4
1e−1	5	1.5e3	1.5e3	1.5e3	1.5e3	5	1.7e4	1.6e4	1.7e4	1.7e4
1e−3	5	1.7e3	1.7e3	1.7e3	1.7e3	5	1.8e4	1.8e4	1.8e4	1.8e4
1e−5	5	1.9e3	1.8e3	1.9e3	1.9e3	5	1.8e4	1.8e4	1.9e4	1.8e4
1e−8	5	2.1e3	2.1e3	2.1e3	2.1e3	5	1.9e4	1.9e4	1.9e4	1.9e4

Δf	f_{11} in 5-D, N=5, mFE=2395 #	ERT	10%	90%	RT_{succ}	f_{11} in 20-D, N=5, mFE=16503 #	ERT	10%	90%	RT_{succ}
10	5	1.3e3	1.2e3	1.4e3	1.3e3	5	1.1e4	1.0e4	1.1e4	1.1e4
1	5	1.4e3	1.3e3	1.6e3	1.4e3	5	1.2e4	1.2e4	1.2e4	1.2e4
1e−1	5	1.6e3	1.5e3	1.7e3	1.6e3	5	1.3e4	1.2e4	1.3e4	1.3e4
1e−3	5	1.8e3	1.7e3	1.9e3	1.8e3	5	1.4e4	1.3e4	1.4e4	1.4e4
1e−5	5	2.0e3	1.9e3	2.0e3	2.0e3	5	1.5e4	1.4e4	1.5e4	1.5e4
1e−8	5	2.2e3	2.1e3	2.3e3	2.2e3	5	1.6e4	1.5e4	1.6e4	1.6e4

Δf	f_{12} in 5-D, N=5, mFE=7403 #	ERT	10%	90%	RT_{succ}	f_{12} in 20-D, N=5, mFE=36471 #	ERT	10%	90%	RT_{succ}
10	5	6.7e2	5.3e2	8.1e2	6.7e2	5	2.2e3	2.1e3	2.3e3	2.2e3
1	5	1.6e3	1.3e3	1.9e3	1.6e3	5	5.9e3	3.5e3	9.3e3	5.9e3
1e−1	5	2.2e3	1.7e3	2.6e3	2.2e3	5	1.1e4	7.1e3	1.5e4	1.1e4
1e−3	5	2.8e3	2.1e3	3.7e3	2.8e3	5	1.7e4	1.3e4	2.0e4	1.7e4
1e−5	5	3.3e3	2.0e3	4.1e3	3.3e3	5	2.2e4	1.9e4	2.6e4	2.2e4
1e−8	5	3.9e3	2.9e3	5.3e3	3.9e3	5	2.8e4	2.5e4	3.1e4	2.8e4

Δf	f_{13} in 5-D, N=5, mFE=7381 #	ERT	10%	90%	RT_{succ}	f_{13} in 20-D, N=5, mFE=189296 #	ERT	10%	90%	RT_{succ}
10	5	5.5e2	3.5e2	7.2e2	5.5e2	5	2.5e3	1.7e3	3.4e3	2.5e3
1	5	1.2e3	9.3e2	1.4e3	1.2e3	5	1.6e4	1.0e4	2.2e4	1.6e4
1e−1	5	1.6e3	1.5e3	1.7e3	1.6e3	5	2.1e4	1.5e4	2.6e4	2.1e4
1e−3	5	2.1e3	2.1e3	2.1e3	2.1e3	5	2.3e4	1.7e4	2.7e4	2.3e4
1e−5	5	2.9e3	2.7e3	3.2e3	2.9e3	5	4.8e4	3.8e4	5.8e4	4.8e4
1e−8	5	4.6e3	3.8e3	5.5e3	4.6e3	5	9.6e4	5.9e4	1.2e5	9.6e4

Δf	f_{14} in 5-D, N=5, mFE=2595 #	ERT	10%	90%	RT_{succ}	f_{14} in 20-D, N=5, mFE=24435 #	ERT	10%	90%	RT_{succ}
10	5	1.3e1	6.0e0	2.0e1	1.3e1	5	2.7e2	2.5e2	3.0e2	2.7e2
1	5	1.2e2	1.1e2	1.4e2	1.2e2	5	6.7e2	6.4e2	7.1e2	6.7e2
1e−1	5	2.3e2	2.2e2	2.5e2	2.3e2	5	1.1e3	1.0e3	1.1e3	1.1e3
1e−3	5	6.4e2	6.1e2	6.8e2	6.4e2	5	3.8e3	3.7e3	4.0e3	3.8e3
1e−5	5	1.4e3	1.3e3	1.4e3	1.4e3	5	9.7e3	9.1e3	1.0e4	9.7e3
1e−8	5	2.4e3	2.2e3	2.5e3	2.4e3	5	2.3e4	2.1e4	2.4e4	2.3e4

Δf	f_{15} in 5-D, N=5, mFE=54419 #	ERT	10%	90%	RT_{succ}	f_{15} in 20-D, N=5, mFE=186159 #	ERT	10%	90%	RT_{succ}
10	5	1.0e3	5.6e2	1.5e3	1.0e3	5	2.7e4	2.6e4	2.7e4	2.7e4
1	5	7.6e3	5.0e3	9.3e3	7.6e3	5	2.7e5	2.3e5	3.1e5	2.7e5
1e−1	5	2.3e4	1.2e4	3.1e4	2.3e4	5	3.1e5	2.5e5	3.7e5	3.1e5
1e−3	5	2.3e4	1.4e4	3.4e4	2.3e4	5	3.1e5	2.5e5	3.7e5	3.1e5
1e−5	5	2.4e4	1.3e4	3.5e4	2.4e4	5	3.2e5	2.6e5	3.8e5	3.2e5
1e−8	5	2.5e4	1.4e4	3.7e4	2.5e4	5	3.3e5	2.7e5	3.9e5	3.3e5

Δf	f_{16} in 5-D, N=5, mFE=11735 #	ERT	10%	90%	RT_{succ}	f_{16} in 20-D, N=5, mFE=816537 #	ERT	10%	90%	RT_{succ}
10	5	3.3e2	2.6e2	4.1e2	3.3e2	5	5.9e3	2.6e3	1.1e4	5.9e3
1	5	2.7e3	9.5e2	4.4e3	2.7e3	5	3.8e4	2.4e4	5.7e4	3.8e4
1e−1	5	8.8e3	8.2e3	9.6e3	8.8e3	5	8.3e4	7.5e4	9.1e4	8.3e4
1e−3	5	9.3e3	8.6e3	9.9e3	9.3e3	5	2.7e5	2.3e5	3.1e5	2.7e5
1e−5	5	9.8e3	9.1e3	1.0e4	9.8e3	5	3.7e5	2.4e5	5.0e5	3.7e5
1e−8	5	1.0e4	9.6e3	1.1e4	1.0e4	5	3.8e5	2.4e5	5.1e5	3.8e5

Δf	f_{17} in 5-D, N=5, mFE=12567 #	ERT	10%	90%	RT_{succ}	f_{17} in 20-D, N=5, mFE=225555 #	ERT	10%	90%	RT_{succ}
10	5	1.8e1	8.4e0	3.2e1	1.8e1	5	1.6e2	1.3e2	1.9e2	1.6e2
1	5	1.8e2	1.6e2	2.0e2	1.8e2	5	1.2e3	1.1e3	1.3e3	1.2e3
1e−1	5	4.0e2	3.7e2	4.2e2	4.0e2	5	8.1e3	2.6e3	1.4e4	8.1e3
1e−3	5	2.4e3	8.8e2	4.0e3	2.4e3	5	2.8e4	2.1e4	3.4e4	2.8e4
1e−5	5	4.6e3	3.0e3	6.4e3	4.6e3	5	9.5e4	6.2e4	1.3e5	9.5e4
1e−8	5	8.4e3	6.0e3	1.1e4	8.4e3	5	1.3e5	1.0e5	1.5e5	1.3e5

Δf	f_{18} in 5-D, N=5, mFE=27313 #	ERT	10%	90%	RT_{succ}	f_{18} in 20-D, N=5, mFE=262843 #	ERT	10%	90%	RT_{succ}
10	5	9.9e1	9.2e1	1.1e2	9.9e1	5	6.2e2	5.7e2	6.5e2	6.2e2
1	5	2.7e2	2.3e2	3.3e2	2.7e2	5	4.7e3	1.7e3	7.6e3	4.7e3
1e−1	5	2.4e3	6.7e2	4.1e3	2.4e3	5	1.9e4	1.8e4	2.1e4	1.9e4
1e−3	5	5.6e3	2.9e3	7.6e3	5.6e3	5	6.8e4	5.4e4	8.2e4	6.8e4
1e−5	5	1.5e4	1.2e4	1.8e4	1.5e4	5	2.1e5	1.8e5	2.3e5	2.1e5
1e−8	5	1.6e4	1.4e4	1.9e4	1.6e4	5	2.3e5	2.0e5	2.5e5	2.3e5

Δf	f_{19} in 5-D, N=5, mFE=334622 #	ERT	10%	90%	RT_{succ}	f_{19} in 20-D, N=5, mFE=4.56e6 #	ERT	10%	90%	RT_{succ}
10	5	1.0e0	1.0e0	1.0e0	1.0e0	5	1.0e0	1.0e0	1.0e0	1.0e0
1	5	1.0e0	1.0e0	1.0e0	1.0e0	5	1.0e0	1.0e0	1.0e0	1.0e0
1e−1	5	1.8e4	1.1e4	2.7e4	1.8e4	5	4.9e5	3.5e5	7.1e5	4.9e5
1e−3	5	1.9e5	1.5e5	2.5e5	1.9e5	5	4.3e6	4.3e6	4.4e6	4.3e6
1e−5	5	1.9e5	1.5e5	2.6e5	1.9e5	5	4.4e6	4.3e6	4.4e6	4.4e6
1e−8	5	2.0e5	1.4e5	2.5e5	2.0e5	5	4.4e6	4.4e6	4.5e6	4.4e6

Δf	f_{20} in 5-D, N=5, mFE=178331 #	ERT	10%	90%	RT_{succ}	f_{20} in 20-D, N=5, mFE=6.98e6 #	ERT	10%	90%	RT_{succ}
10	5	6.1e1	4.7e1	7.4e1	6.1e1	5	4.3e2	3.9e2	4.7e2	4.3e2
1	5	1.2e4	5.8e3	1.8e4	1.2e4	5	4.0e5	3.4e5	4.6e5	4.0e5
1e−1	5	1.4e5	1.3e5	1.5e5	1.4e5	5	4.1e6	3.3e6	5.1e6	4.1e6
1e−3	5	1.5e5	1.4e5	1.6e5	1.5e5	5	5.5e6	4.7e6	6.2e6	5.5e6
1e−5	5	1.5e5	1.4e5	1.6e5	1.5e5	5	5.5e6	4.7e6	6.3e6	5.5e6
1e−8	5	1.5e5	1.4e5	1.6e5	1.5e5	5	5.6e6	4.8e6	6.3e6	5.6e6

Δf	f_{21} in 5-D, N=5, mFE=23789 #	ERT	10%	90%	RT_{succ}	f_{21} in 20-D, N=5, mFE=2.24e6 #	ERT	10%	90%	RT_{succ}
10	5	1.2e2	8.1e1	1.5e2	1.2e2	5	8.8e3	3.9e3	1.5e4	8.8e3
1	5	9.0e3	5.0e3	1.5e4	9.0e3	5	4.4e5	2.3e5	6.6e5	4.4e5
1e−1	5	1.3e4	7.0e3	1.7e4	1.3e4	4	1.0e6	3.0e5	2.2e6	4.6e5
1e−3	5	1.3e4	9.2e3	1.8e4	1.3e4	4	1.0e6	3.7e5	2.2e6	4.6e5
1e−5	5	1.4e4	9.0e3	1.9e4	1.4e4	4	1.0e6	3.7e5	2.2e6	4.6e5
1e−8	5	1.4e4	9.8e3	1.9e4	1.4e4	4	1.0e6	3.2e5	2.2e6	4.7e5

Δf	f_{22} in 5-D, N=5, mFE=183959 #	ERT	10%	90%	RT_{succ}	f_{22} in 20-D, N=5, mFE=2.43e6 #	ERT	10%	90%	RT_{succ}
10	5	7.4e1	5.6e1	8.8e1	7.4e1	5	6.6e3	4.5e2	1.9e4	6.6e3
1	5	7.9e2	1.4e2	1.4e3	7.9e2	5	1.3e5	4.2e4	2.2e5	1.3e5
1e−1	5	1.2e4	1.2e3	2.2e4	1.2e4	0	*69e−2*	*69e−2*	*69e−2*	7.1e4
1e−3	5	4.7e4	9.5e3	8.3e4	4.7e4	.	.	.	.	.
1e−5	5	4.7e4	9.8e3	1.1e5	4.7e4	.	.	.	.	.
1e−8	5	4.7e4	6.2e3	8.4e4	4.7e4	.	.	.	.	.

Δf	f_{23} in 5-D, N=5, mFE=88264 #	ERT	10%	90%	RT_{succ}	f_{23} in 20-D, N=5, mFE=2.56e6 #	ERT	10%	90%	RT_{succ}
10	5	1.1e1	6.7e0	1.5e1	1.1e1	5	3.6e0	2.3e0	5.0e0	3.6e0
1	5	8.0e3	5.7e3	1.1e4	8.0e3	5	3.5e4	2.7e4	4.4e4	3.5e4
1e−1	5	2.2e4	1.2e4	3.1e4	2.2e4	5	4.1e4	2.9e4	5.3e4	4.1e4
1e−3	5	2.3e4	1.5e4	3.1e4	2.3e4	5	1.4e6	1.1e6	1.7e6	1.4e6
1e−5	5	2.4e4	1.2e4	3.2e4	2.4e4	5	1.5e6	1.1e6	1.8e6	1.5e6
1e−8	5	3.9e4	2.3e4	5.6e4	3.9e4	5	1.5e6	1.2e6	1.9e6	1.5e6

Δf	f_{24} in 5-D, N=5, mFE=2.70e6 #	ERT	10%	90%	RT_{succ}	f_{24} in 20-D, N=5, mFE=1.17e7 #	ERT	10%	90%	RT_{succ}
10	5	4.4e3	3.2e3	5.7e3	4.4e3	5	8.9e5	4.2e5	1.5e6	8.9e5
1	5	3.5e5	1.3e5	5.9e5	3.5e5	4	7.0e6	3.9e6	1.2e7	4.4e6
1e−1	3	3.4e6	2.0e6	5.8e6	1.6e6	1	5.4e7	2.7e7	>5e7	1.2e7
1e−3	2	5.8e6	2.9e6	∞	1.8e6	1	5.4e7	2.7e7	>5e7	1.2e7
1e−5	2	5.9e6	3.4e6	1.3e7	1.8e6	1	5.4e7	2.7e7	>5e7	1.2e7
1e−8	2	5.9e6	3.1e6	∞	1.9e6	1	5.4e7	2.7e7	>5e7	1.2e7

Table 1: Shown are, for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\rm opt}+\Delta f$ (ERT, see Figure 1); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value ($\mathrm{RT}_{\rm succ}$). If $f_{\rm opt}+\Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 1 for the names of functions.

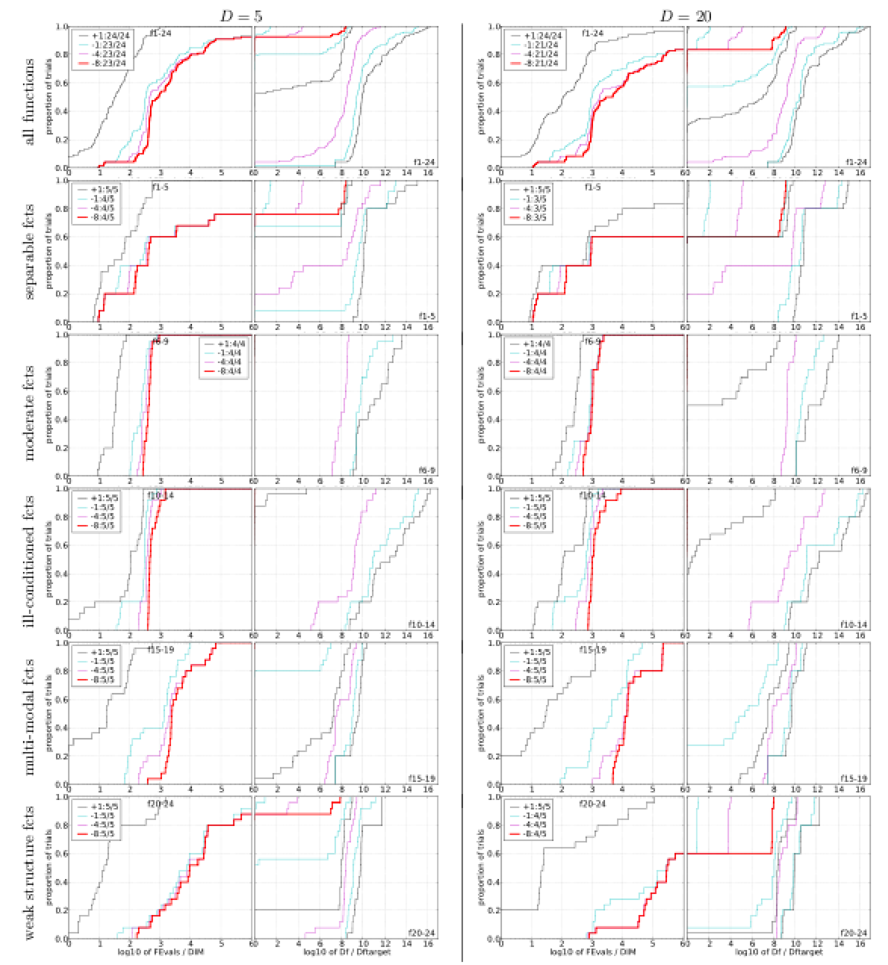


Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\rm opt}+\Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. Right subplots: ECDF of the best achieved Δf divided by 10^k (upper left lines in continuation of the left subplot), and best achieved Δf divided by 10^{-8} for running times of $D, 10\,D, 100\,D \ldots$ function evaluations (from right to left cycling black-cyan-magenta). Top row: all functions; second row: separable functions; third row: misc. moderate functions; fourth row: ill-conditioned functions; fifth row: multi-modal functions with adequate structure; last row: multi-modal functions with weak structure. The legends indicate the number of functions that were solved in at least one trial. FEvals denotes number of function evaluations, D and DIM denote search space dimension, and Δf and Df denote the difference to the optimal function value.

Submitted Data Sets

- 2009: 31 noiseless and 21 noisy “data sets”
- 2010: 24 noiseless and 16 noisy “data sets”
- 2012: 30 noiseless and 4 noisy “data sets”
- Algorithms: RCGAs (e.g. plain, PCX), EDAs (e.g. IDEA), BFGS & (many) other “classical” methods, ESs (e.g. CMA), PSO, DE, Ant-Stigmergy Alg, Bee Colony, EGS, SPSA, Meta-Strategies...

Components of CoCO

- BBO function testbeds (currently two)

  determine the "scientific question"

- experimental protocol

  important in the details, major future changes are unlikely

- data writing/storage protocol

  to be adapted/extended for noisy, constraint & MO case
  long-term data format needs to be determined

- data post-processing and presentation

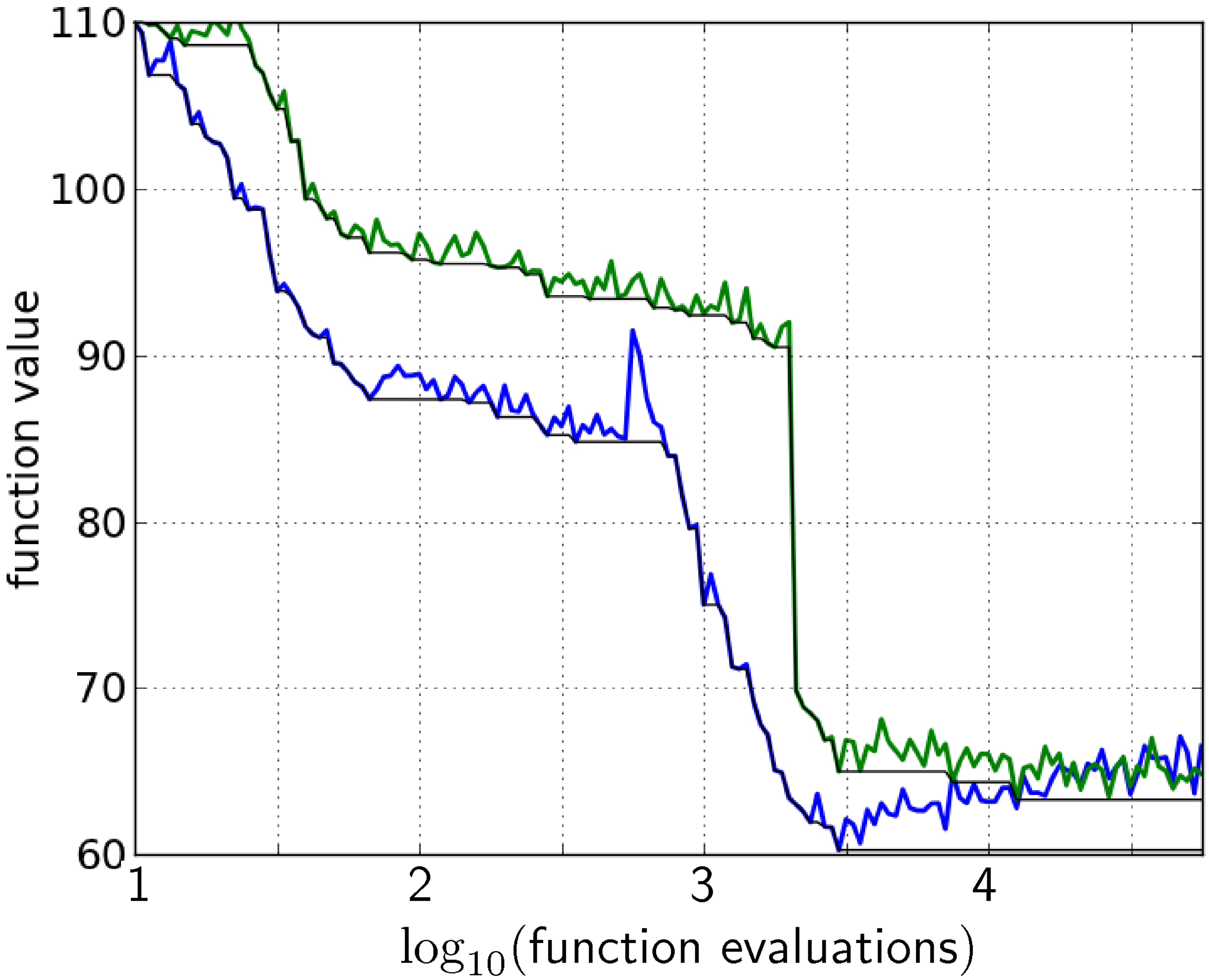
  continually evolving and improving
  to be adapted/extended for noisy, constraint & MO case

How do we measure performance?

Measuring Performance

convergence graphs is
all we have to start with

Two Convergence Graphs



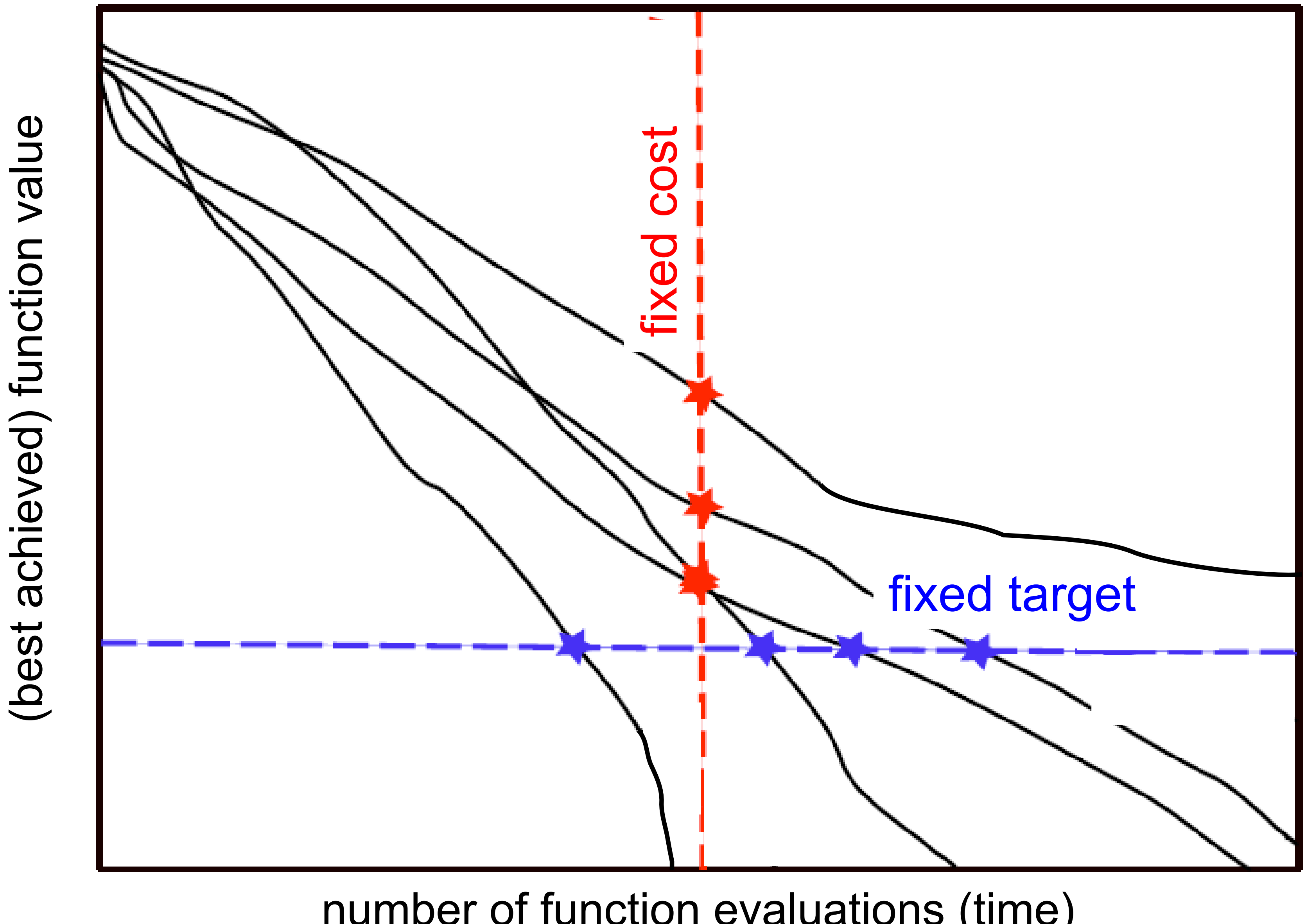
(recall) Black-Box Optimization

Two objectives:

- Find solution with a smallest possible function value
- With the least possible search costs (number of function evaluations)
- For measuring performance: fix one and measure the other

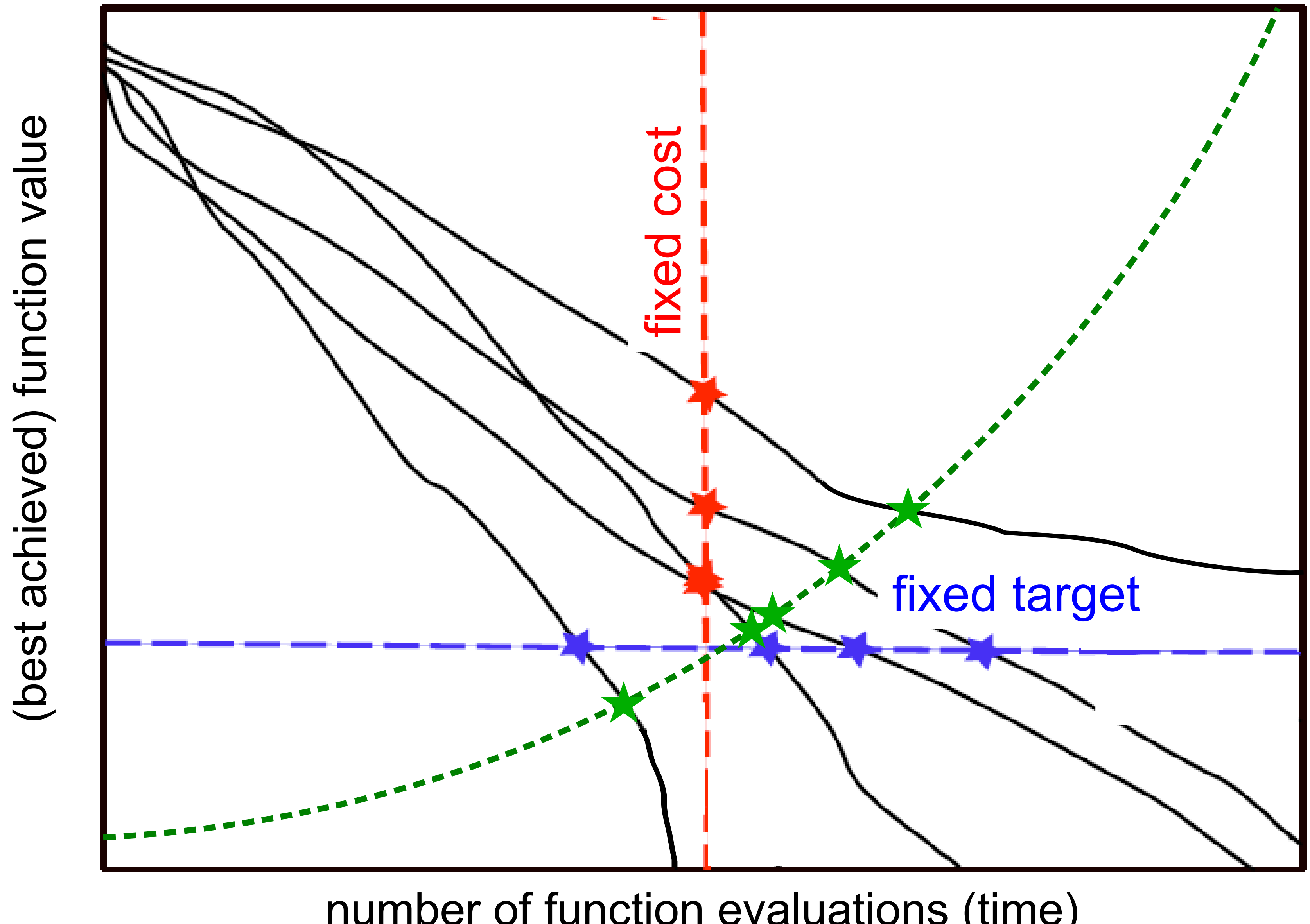
Measuring Performance from Convergence Graphs

fixed-cost versus fixed-target



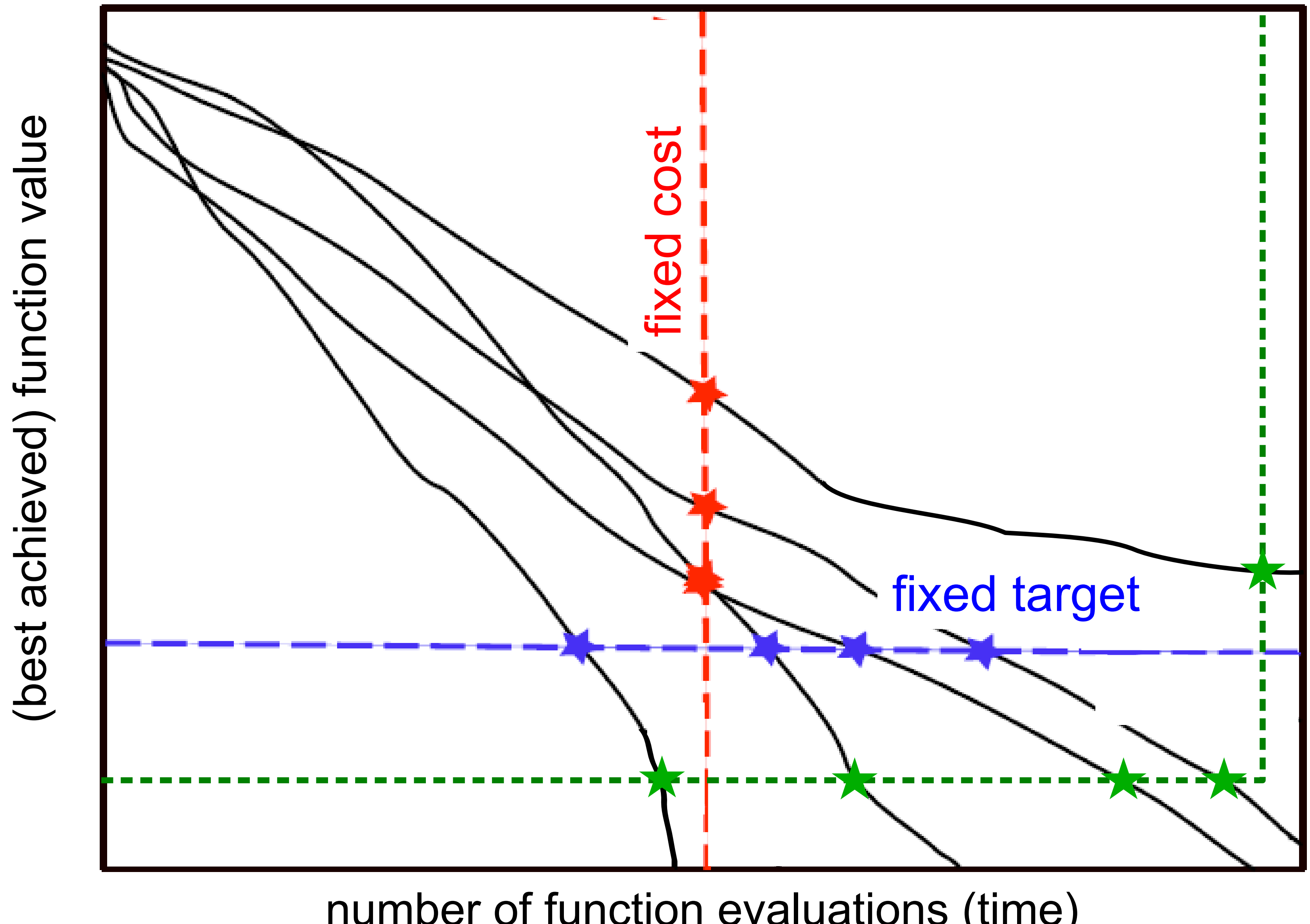
Measuring Performance from Convergence Graphs

fixed-cost versus fixed-target



Measuring Performance from Convergence Graphs

fixed-cost versus fixed-target



Evaluation of Search Algorithms

Behind the scene

a performance should be

- quantitative on the ratio scale (highest possible)
  - “algorithm A is two *times* better than algorithm B” is a meaningful statement
  - can assume a wide range of values

- meaningful (interpretable) with regard to the real world

possible to transfer from benchmarking to real world

runtime is the prime candidate (we don't have many choices anyway)

Fixed-target: Measuring Runtime

We measure runtime in number of function evaluations

- as a distribution of runtimes
- as expected runtime ERT

For success probability $0 < p < 1$: (simulated) restarts until a successful run is observed.

$$\mathrm{RT} = \mathrm{RT}_{\mathrm{succ}} + \sum \mathrm{RT}_{\mathrm{unsucc}}$$

$$\approx E(\mathrm{RT}_{\mathrm{succ}}) + \frac{1-p}{p} E(\mathrm{RT}_{\mathrm{unsucc}})$$

Feature&drawback: termination method for unsuccessful trials can be critical

The data we use

- Currently: samples

points that the algorithm evaluates

- In future: recommendations or samples

points the algorithm proposes as solution to the search problem (in each time step)

results in a sequence (or two) of fitness function values

The performance measure we use

- First hitting time to a given target function value in number of fitness function evaluations

equivalent to first hitting time of a sublevel set in search space

Performance Measure: Assumptions

We make three implicit assumptions

- algorithms are any-time

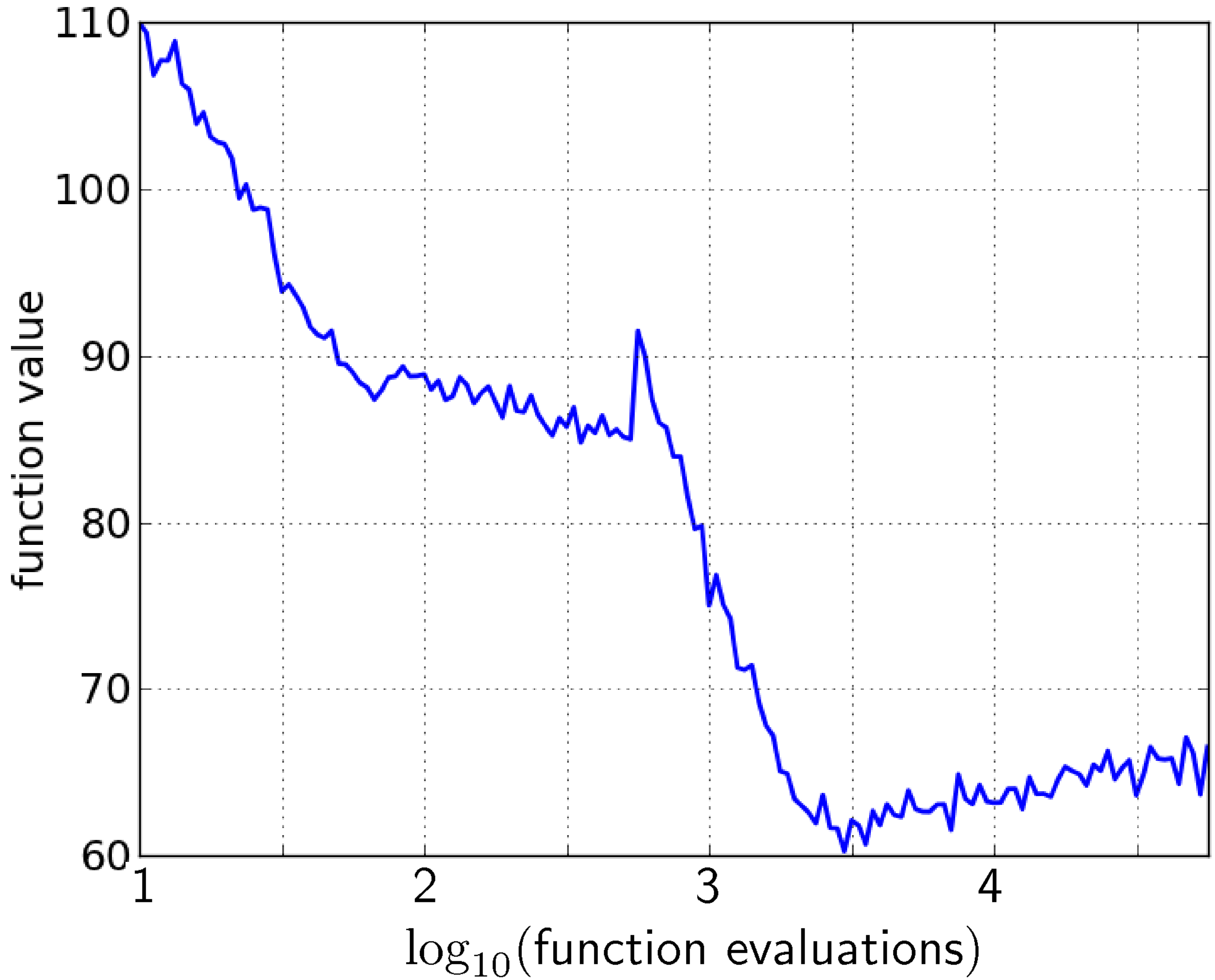
  one run can serve to evaluate performance for each time step and/or each target value

- the “true performance” of an algorithm improves with further evaluations

  is monotonous in the number of function evaluations

- A lucky punch is not "possible"

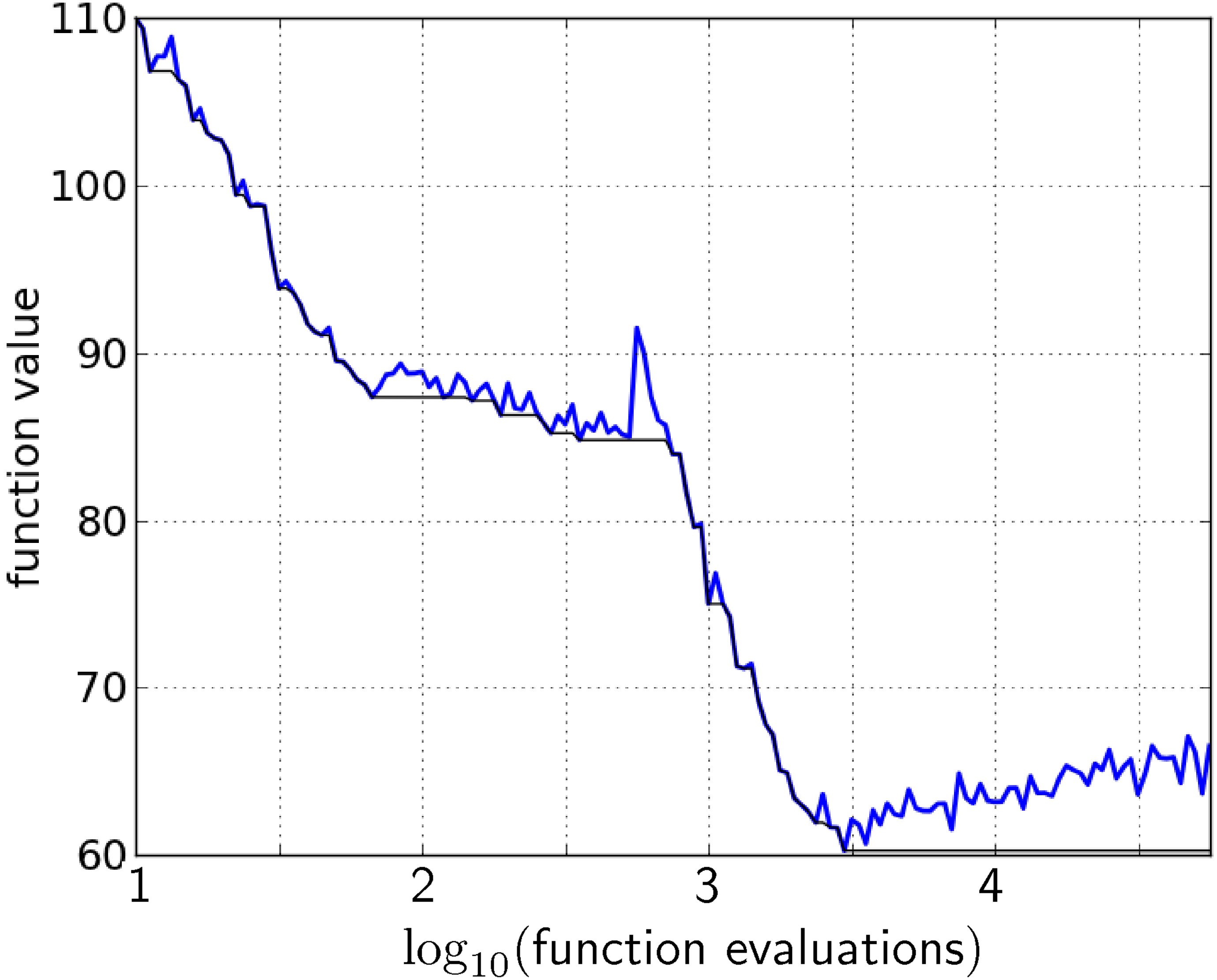
A Convergence Graph



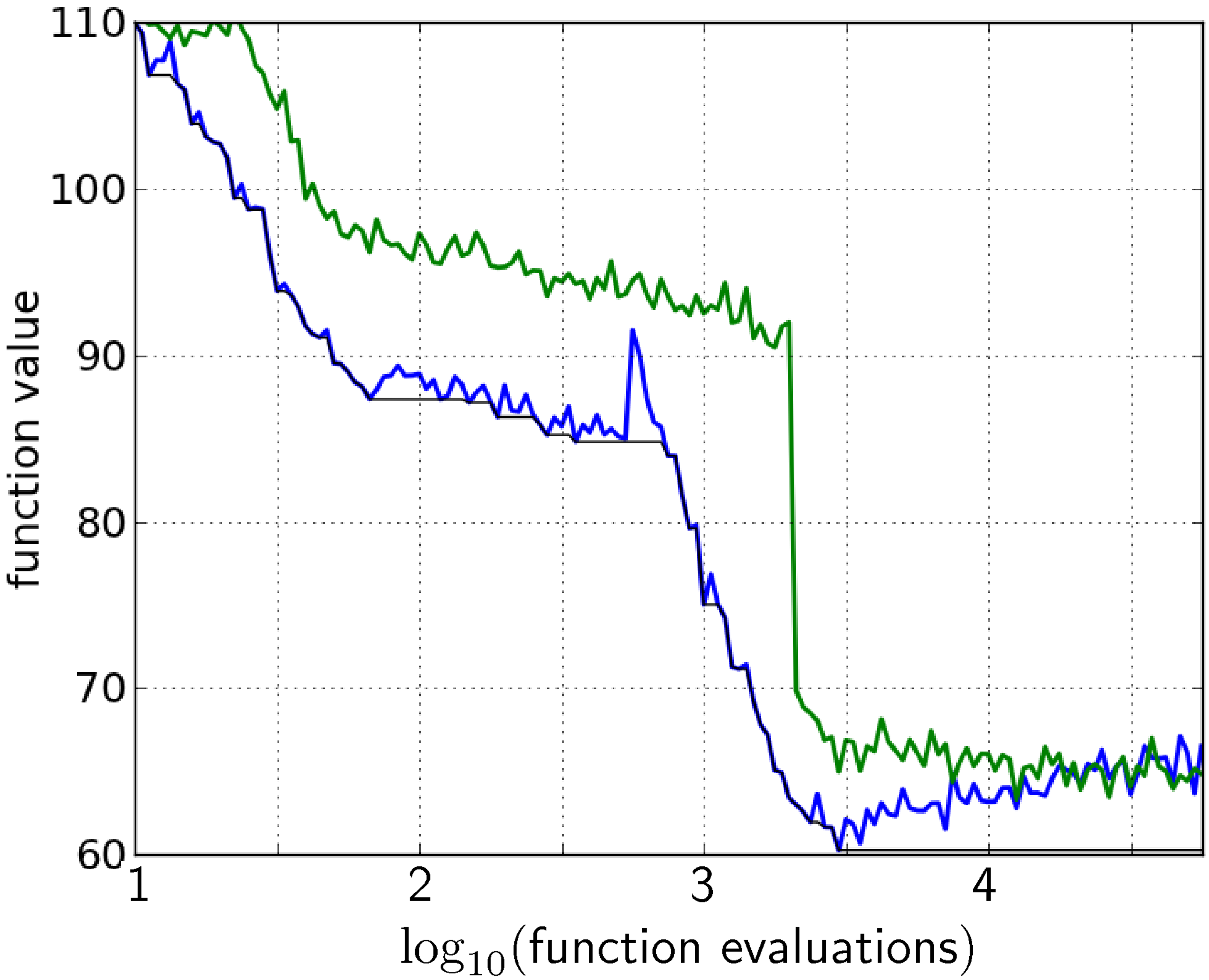
- as a single graph violates the second assumption for >1000 fevals

.

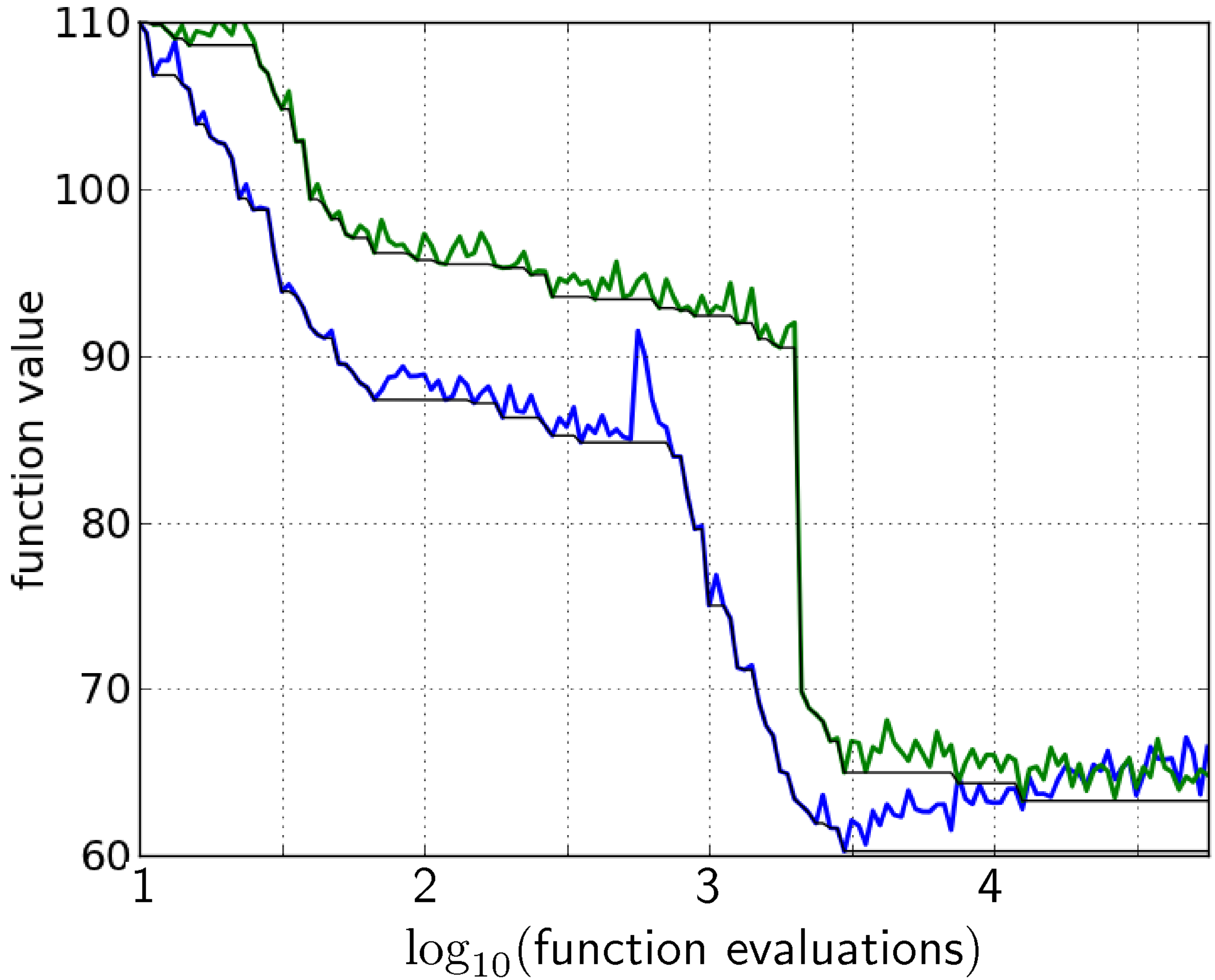
First hitting time is monotonous



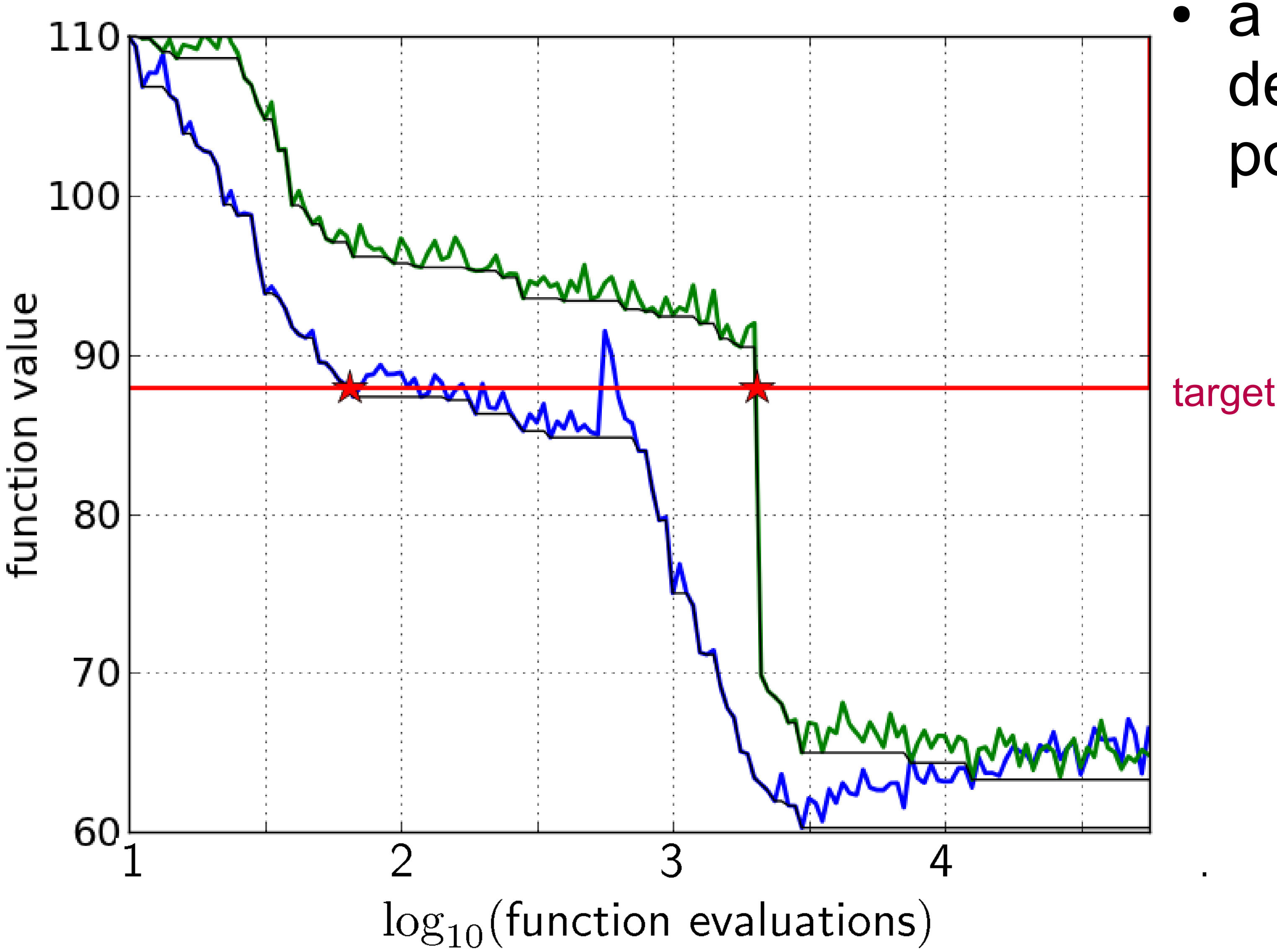
- first hitting time: a monotonous graph



- another convergence graph

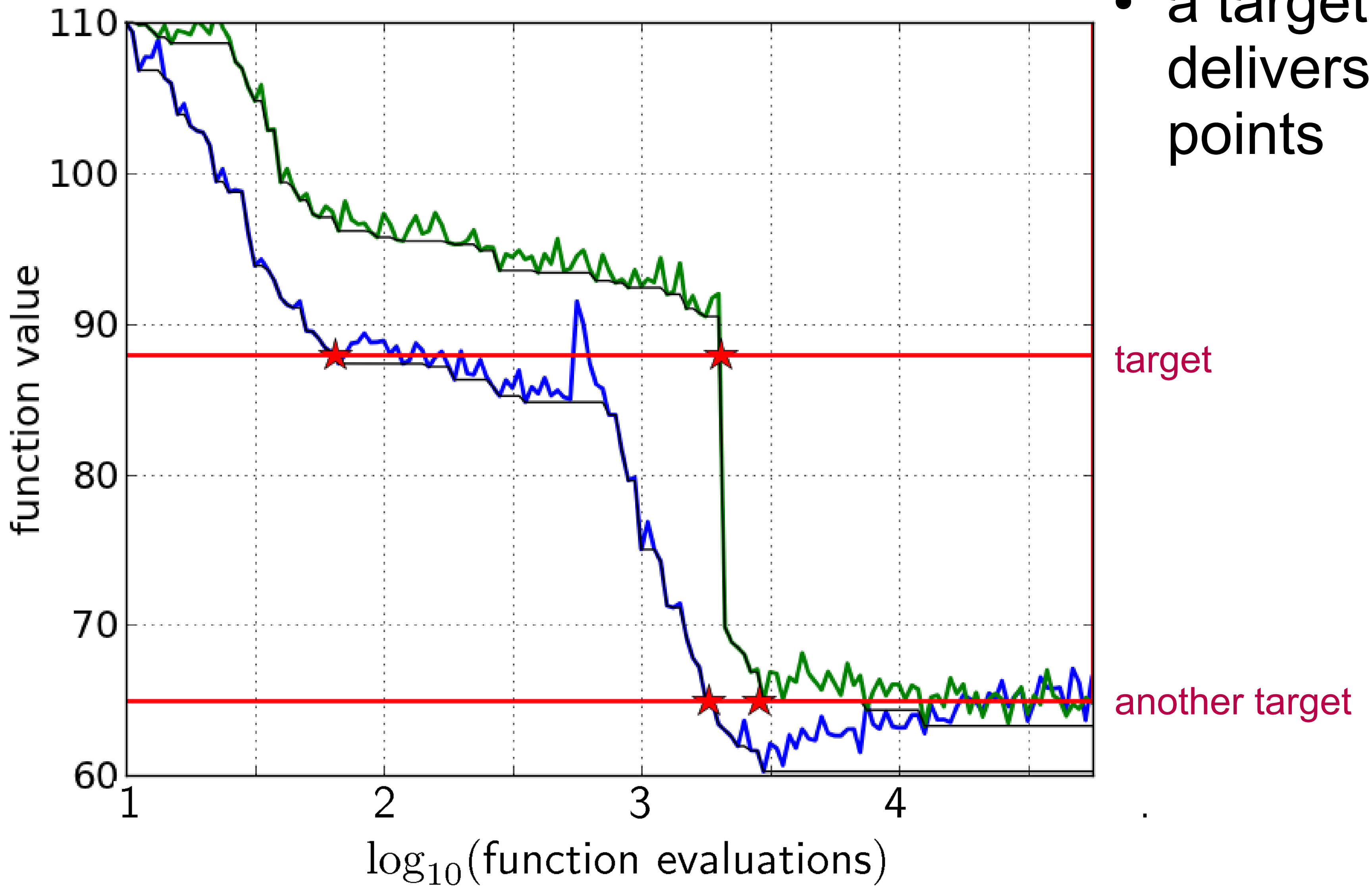


- another convergence graph with hitting time



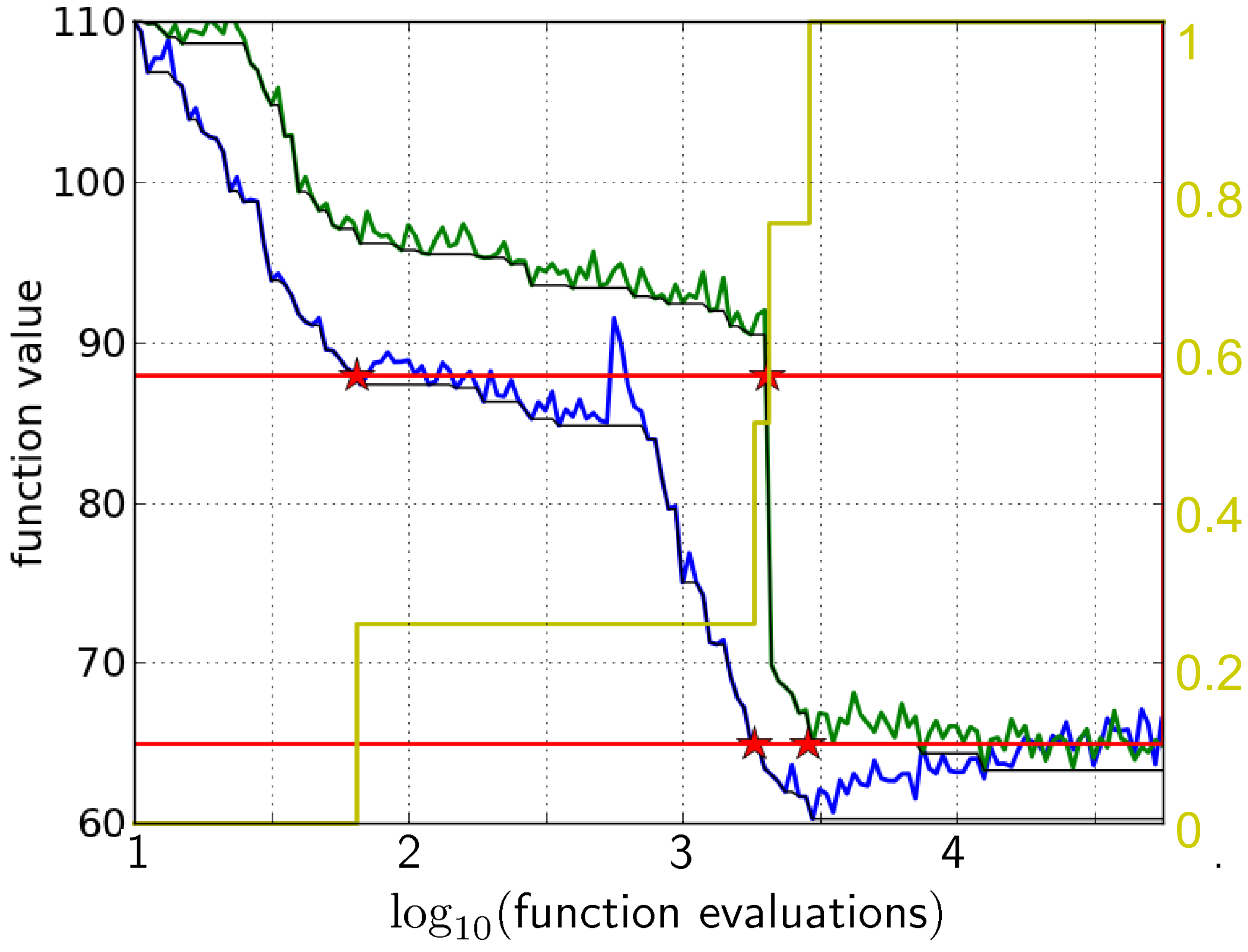
- a target value delivers two data points

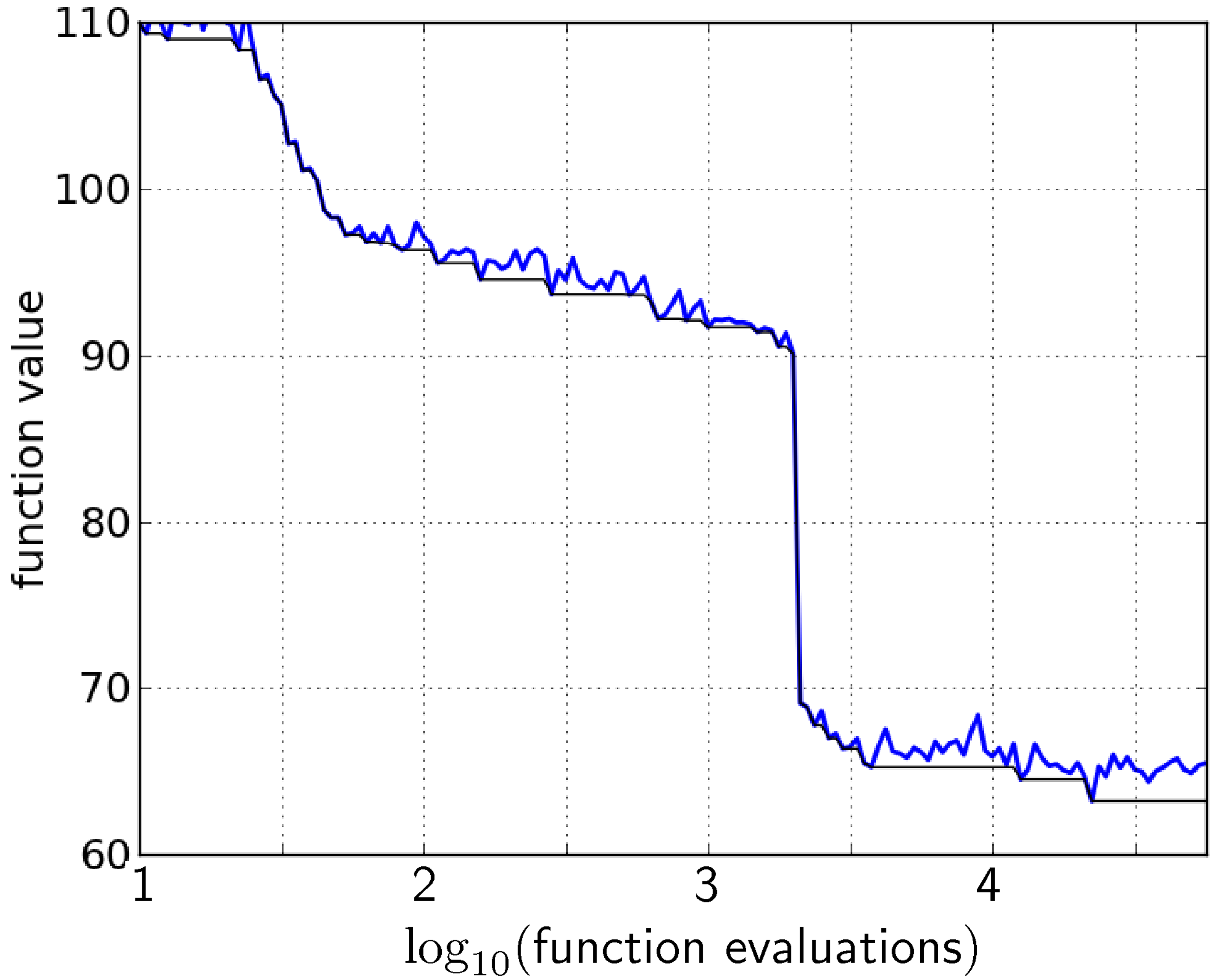
.



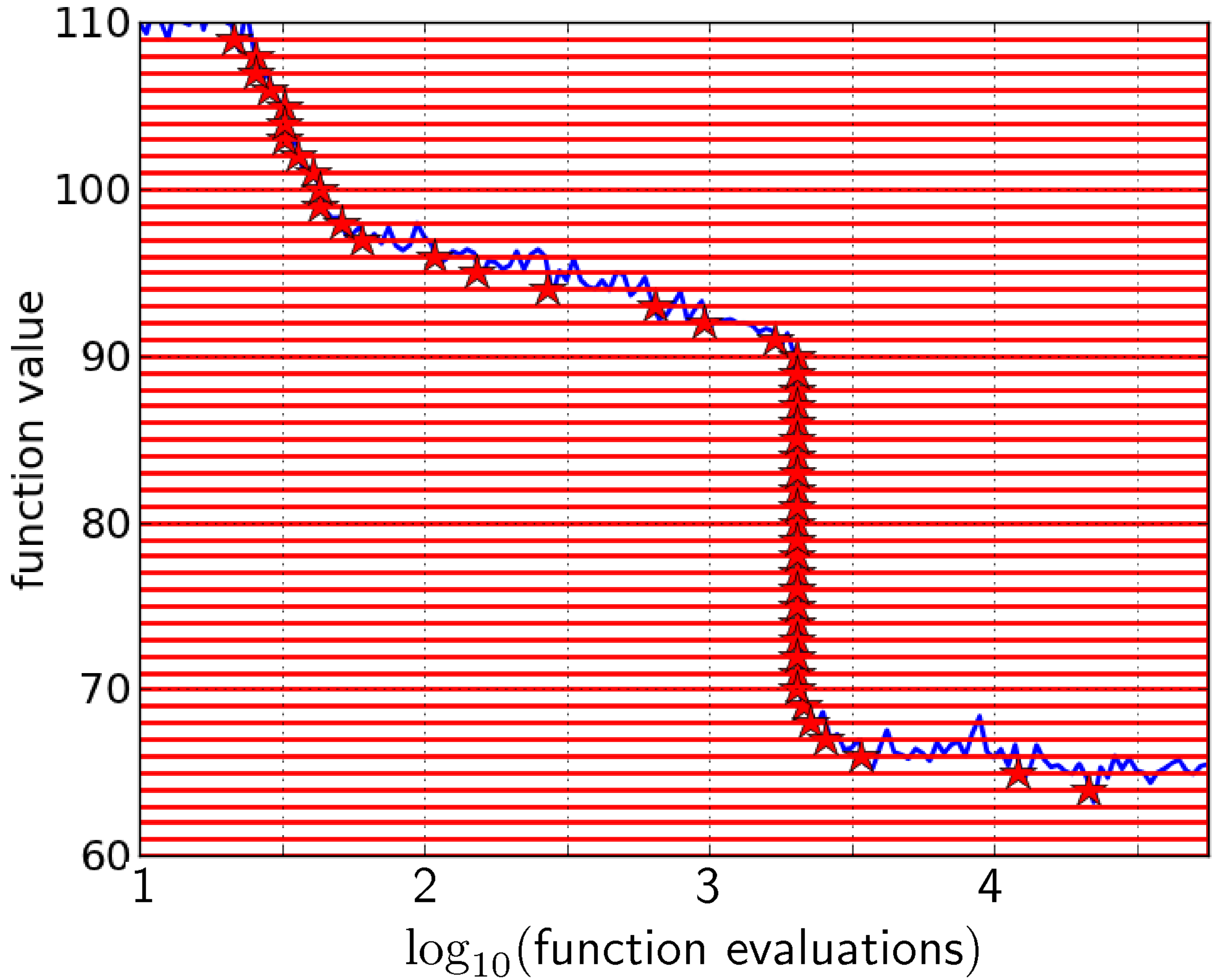
- a target value delivers two data points

ECDF with four data points

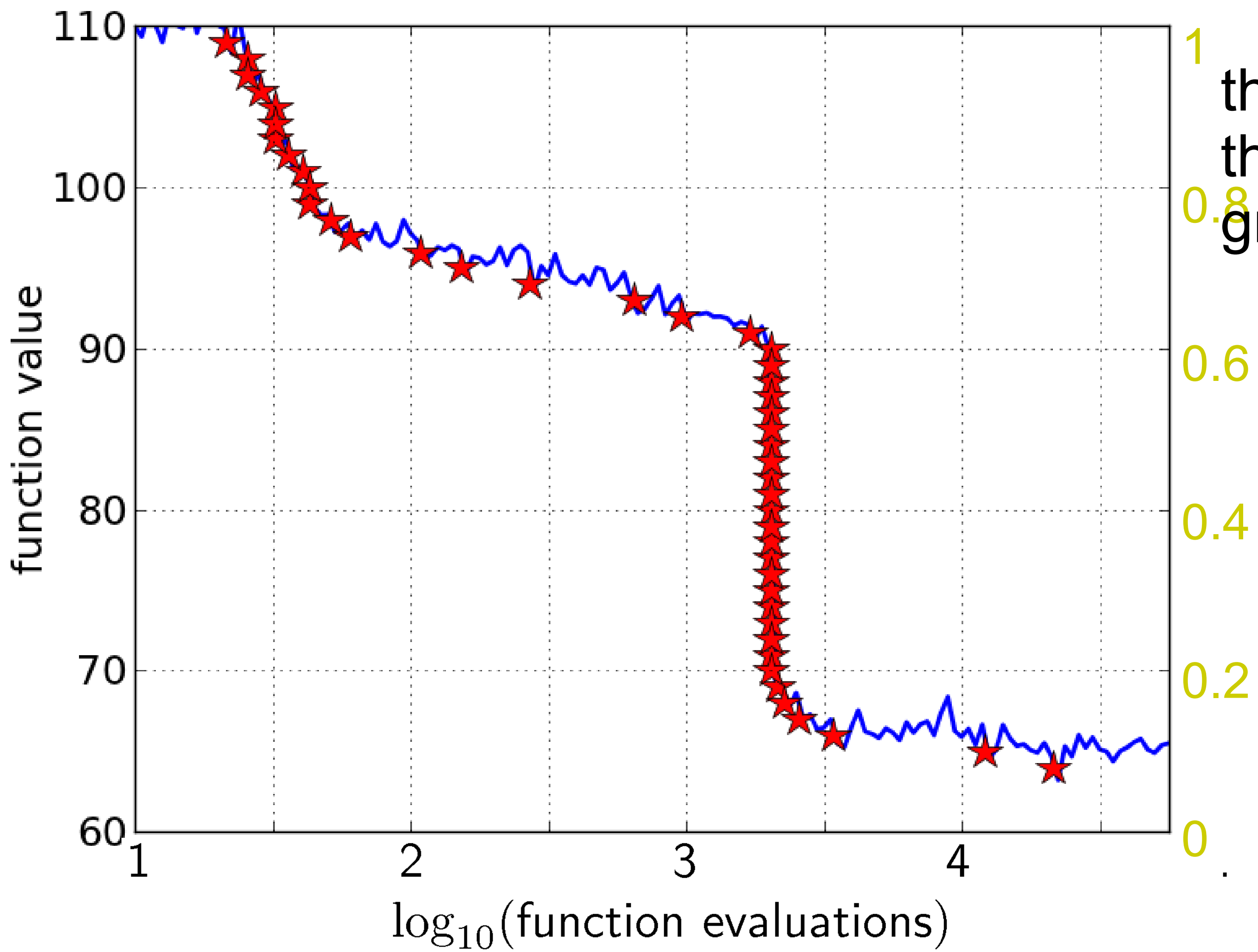




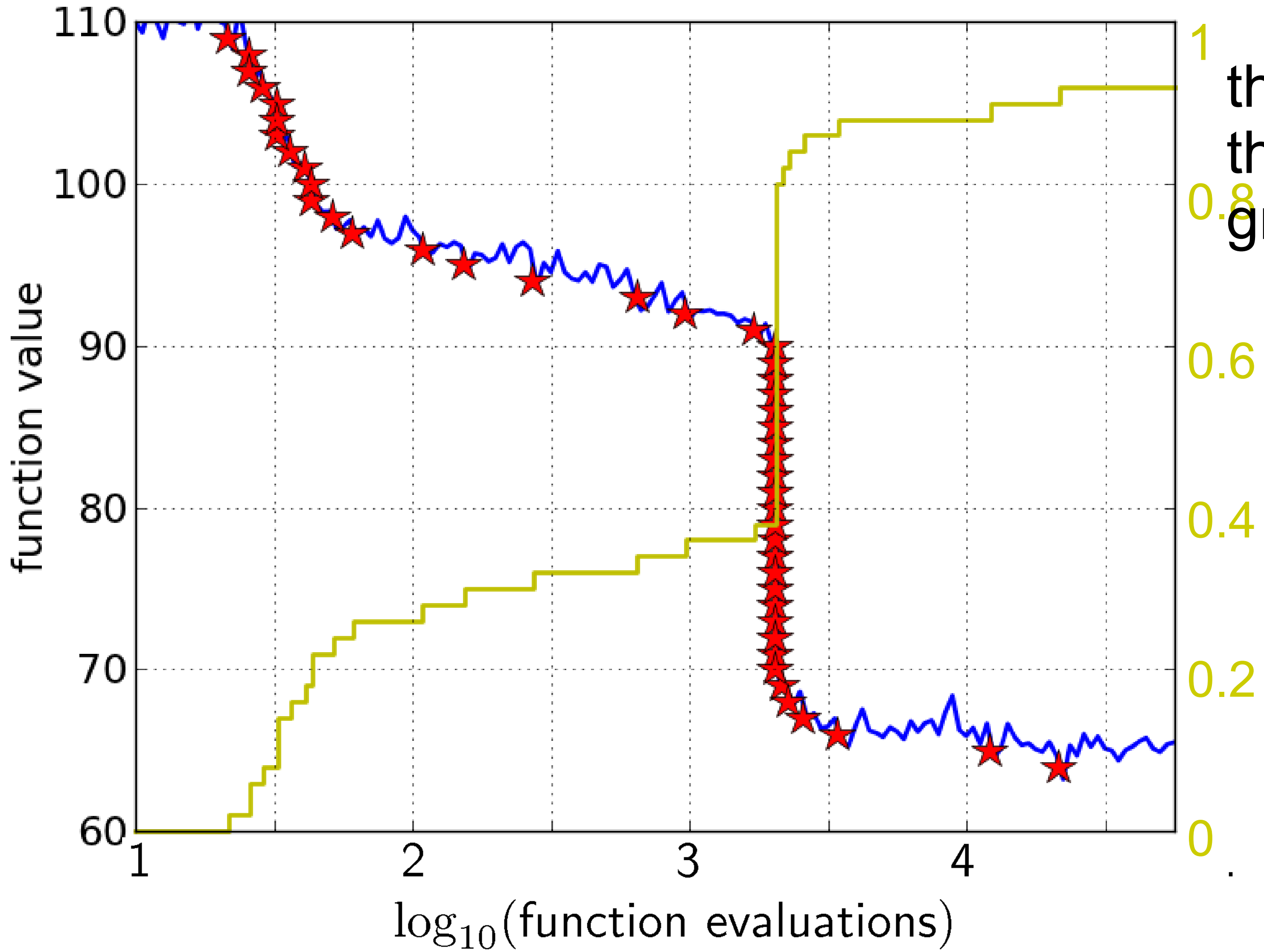
- reconstructing a single run



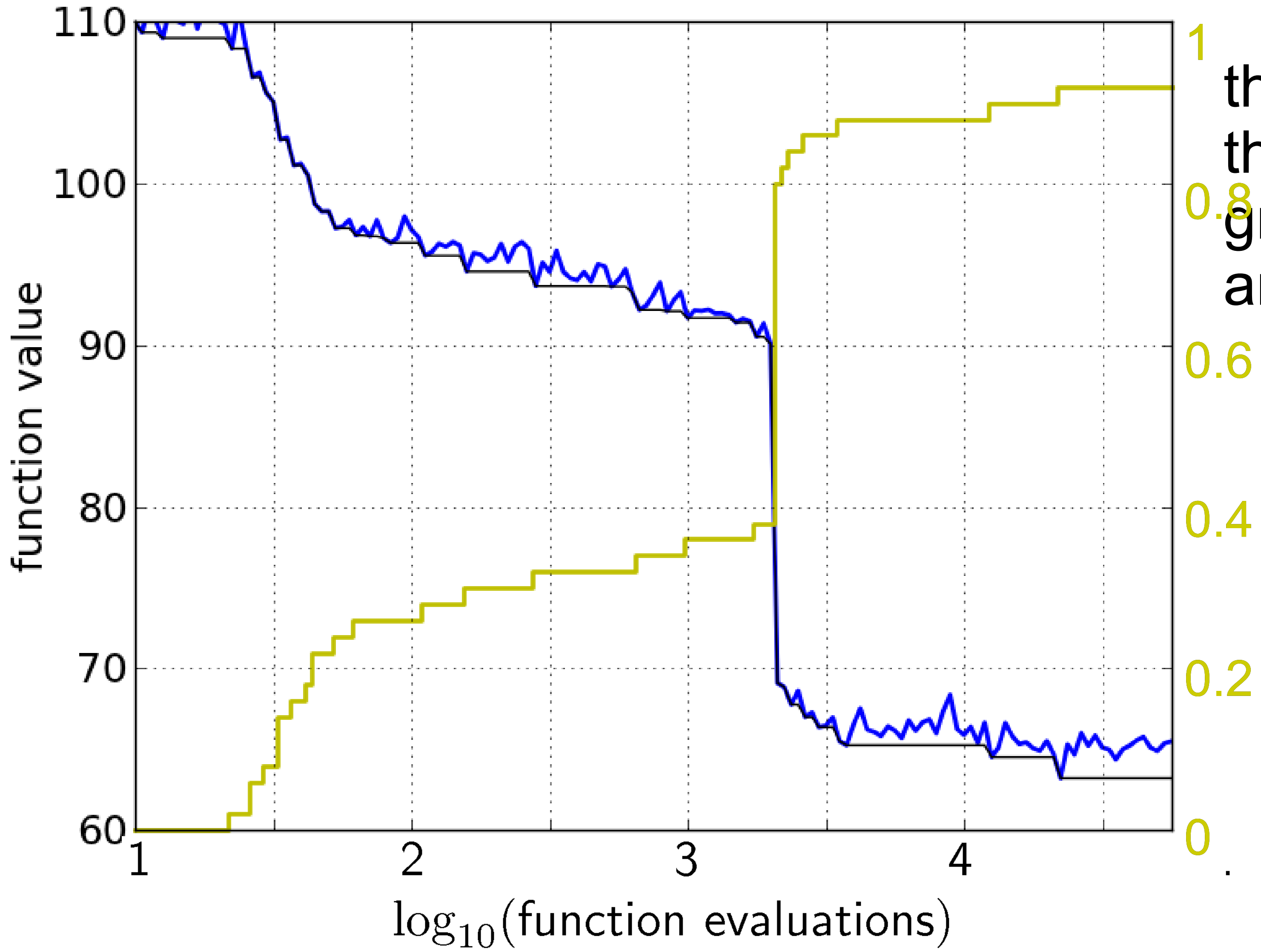
50 equally spaced targets



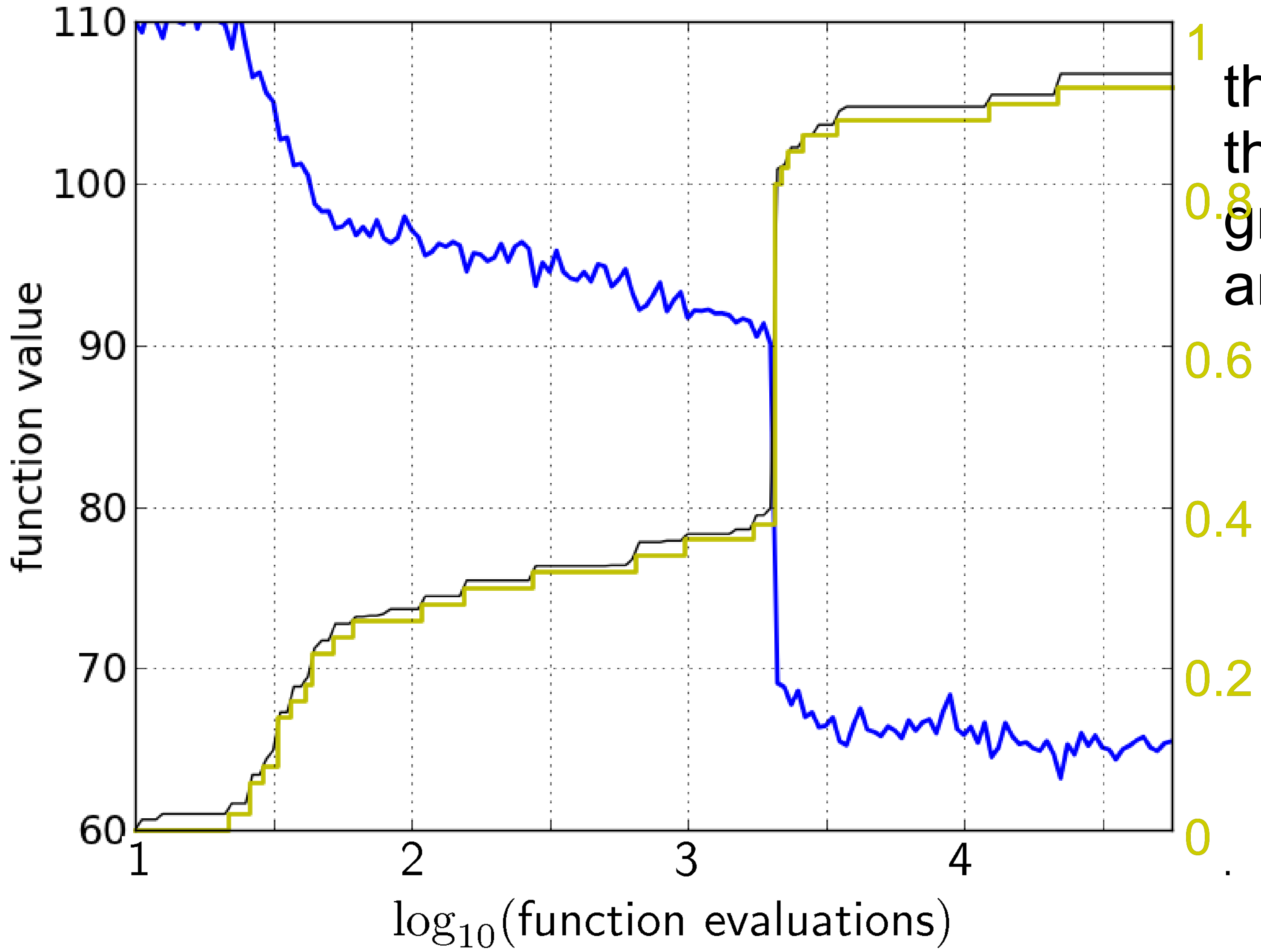
the ECDF recovers the monotonous graph



the ECDF recovers the monotonous graph

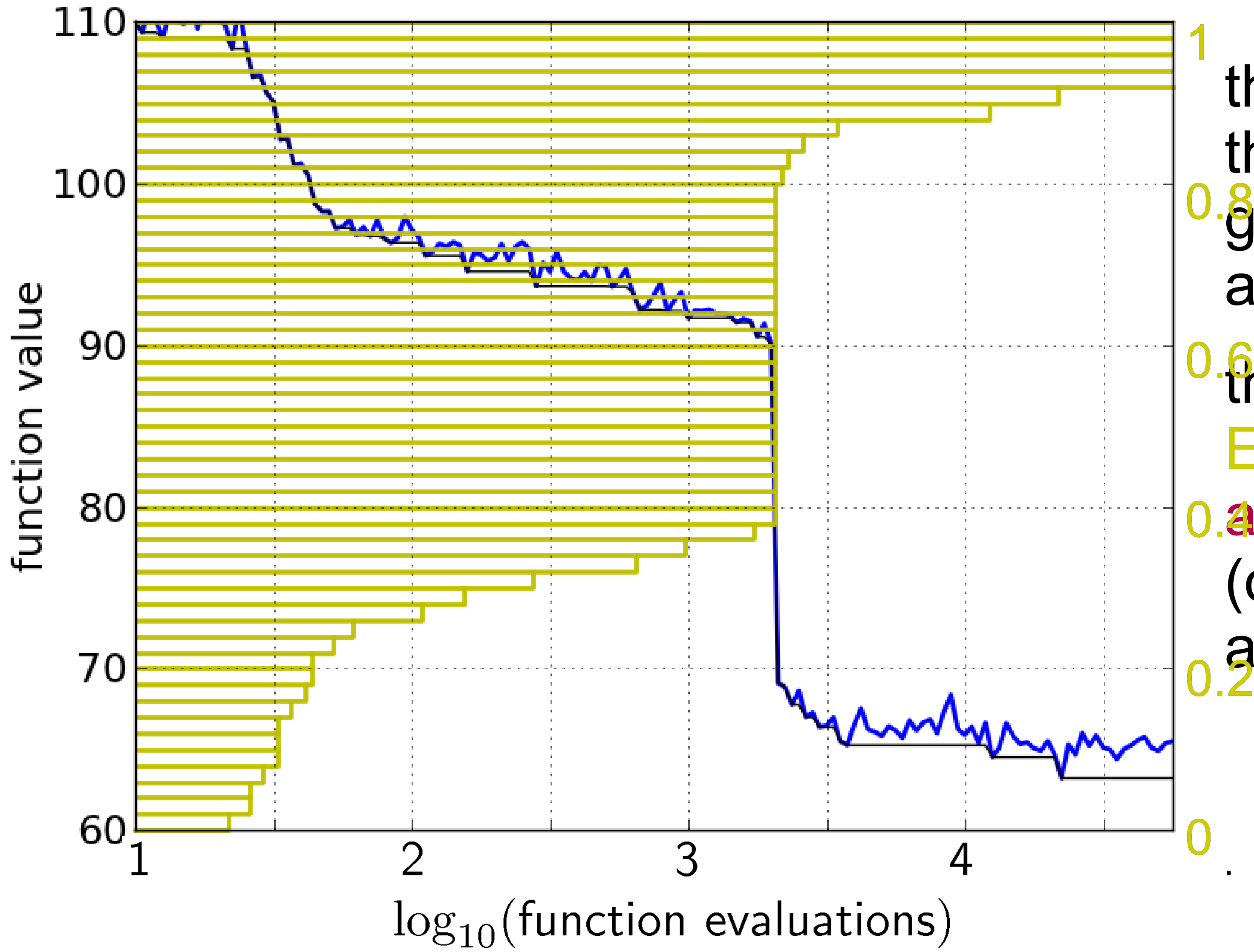


the ECDF recovers the monotonous graph, discretized and flipped



the ECDF recovers the monotonous graph, discretised and flipped

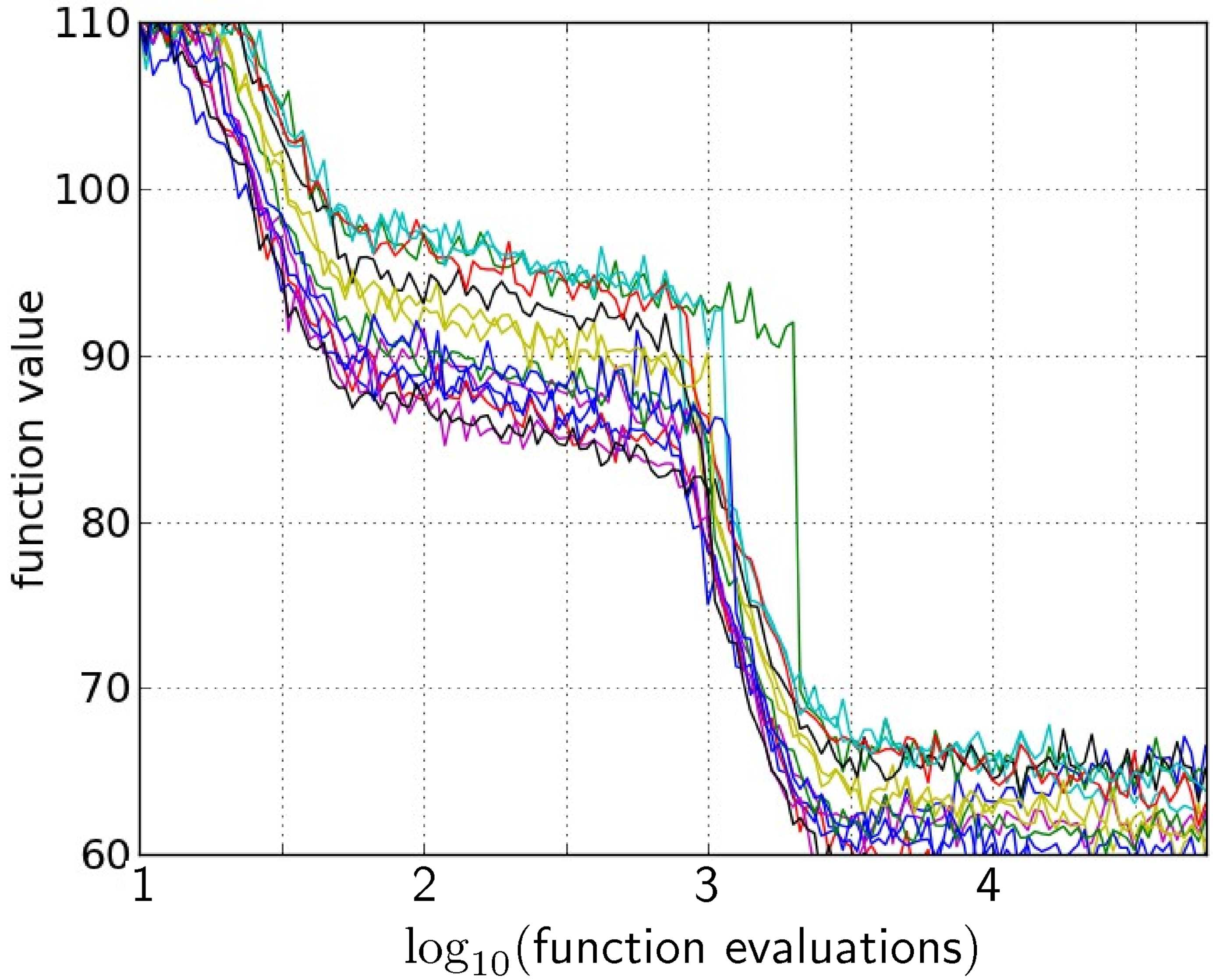
.



the ECDF recovers the monotonous graph, discretized and flipped

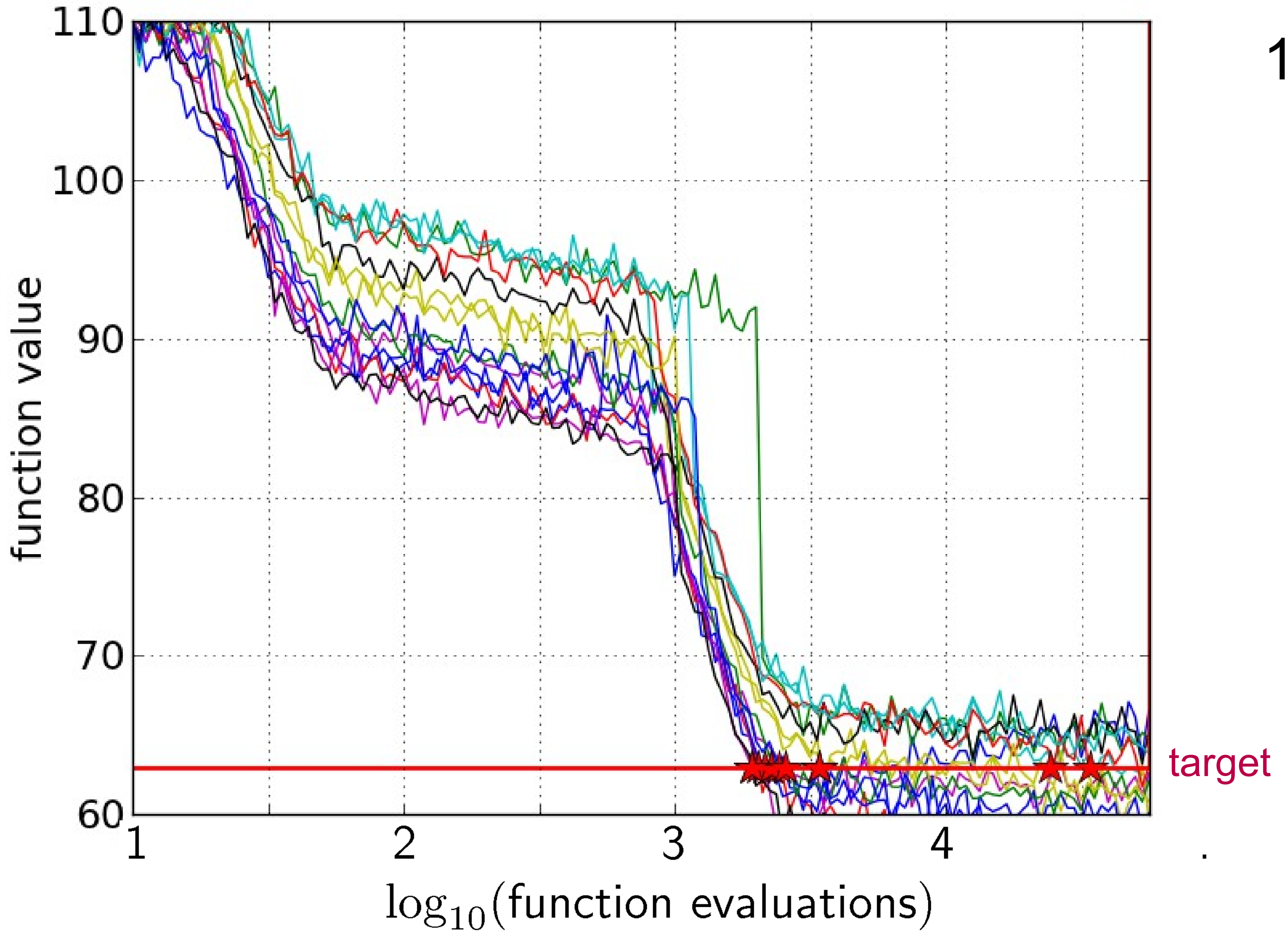
the area over the ECDF curve is the average log runtime (or geometric average runtime)

.

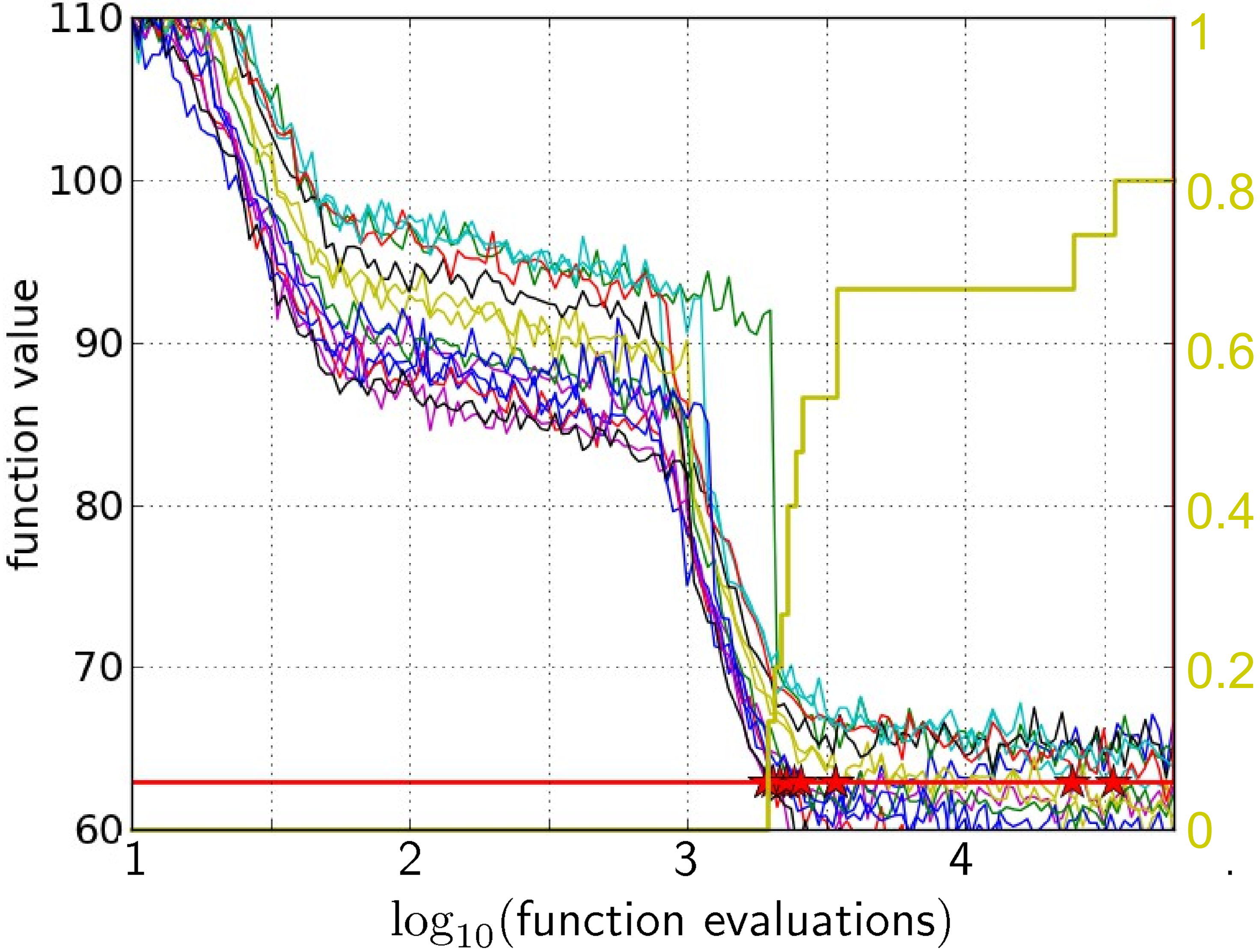


110
100
90
80
70
60
function value
1
2
3
4
log10(function evaluations)

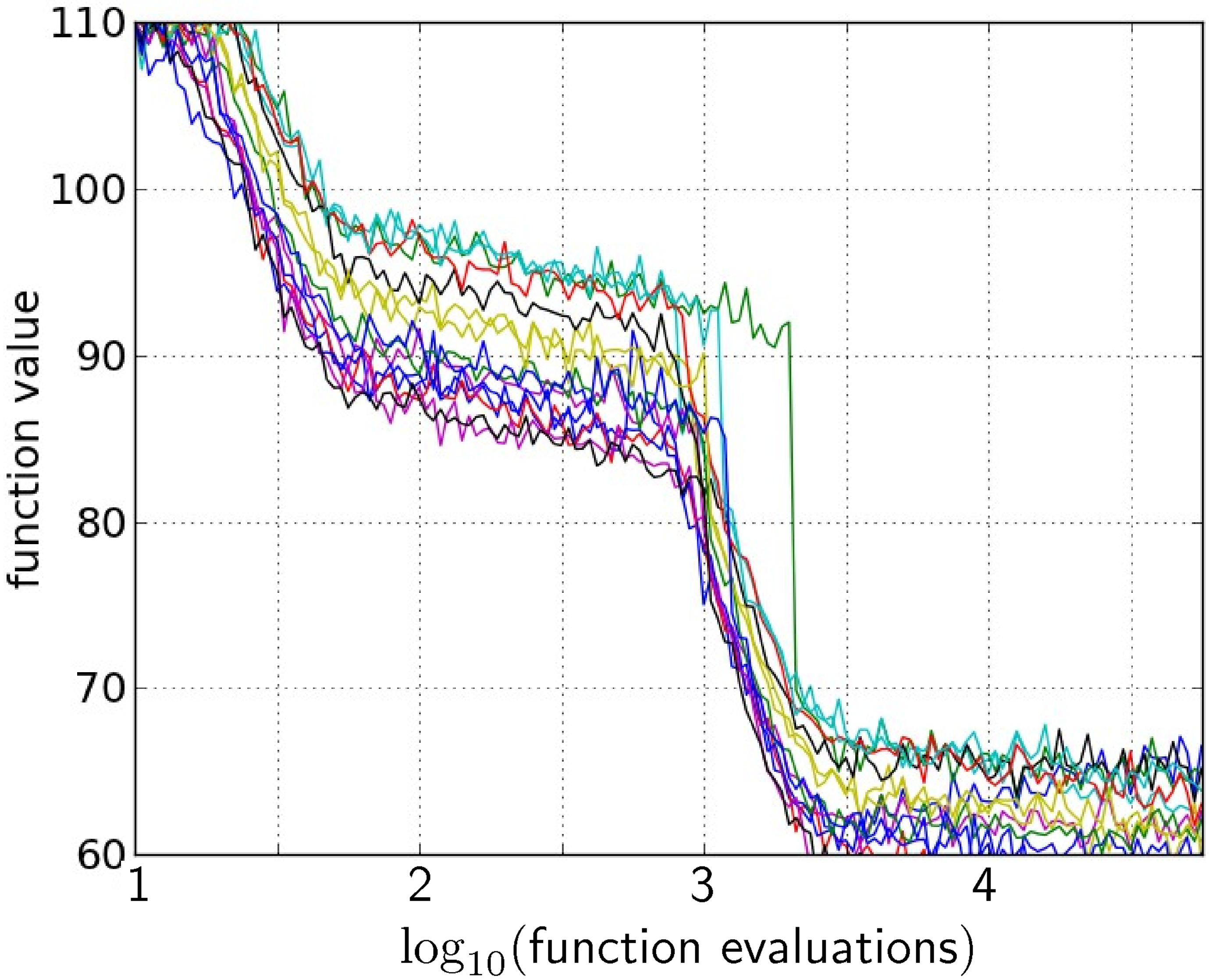
15 runs


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

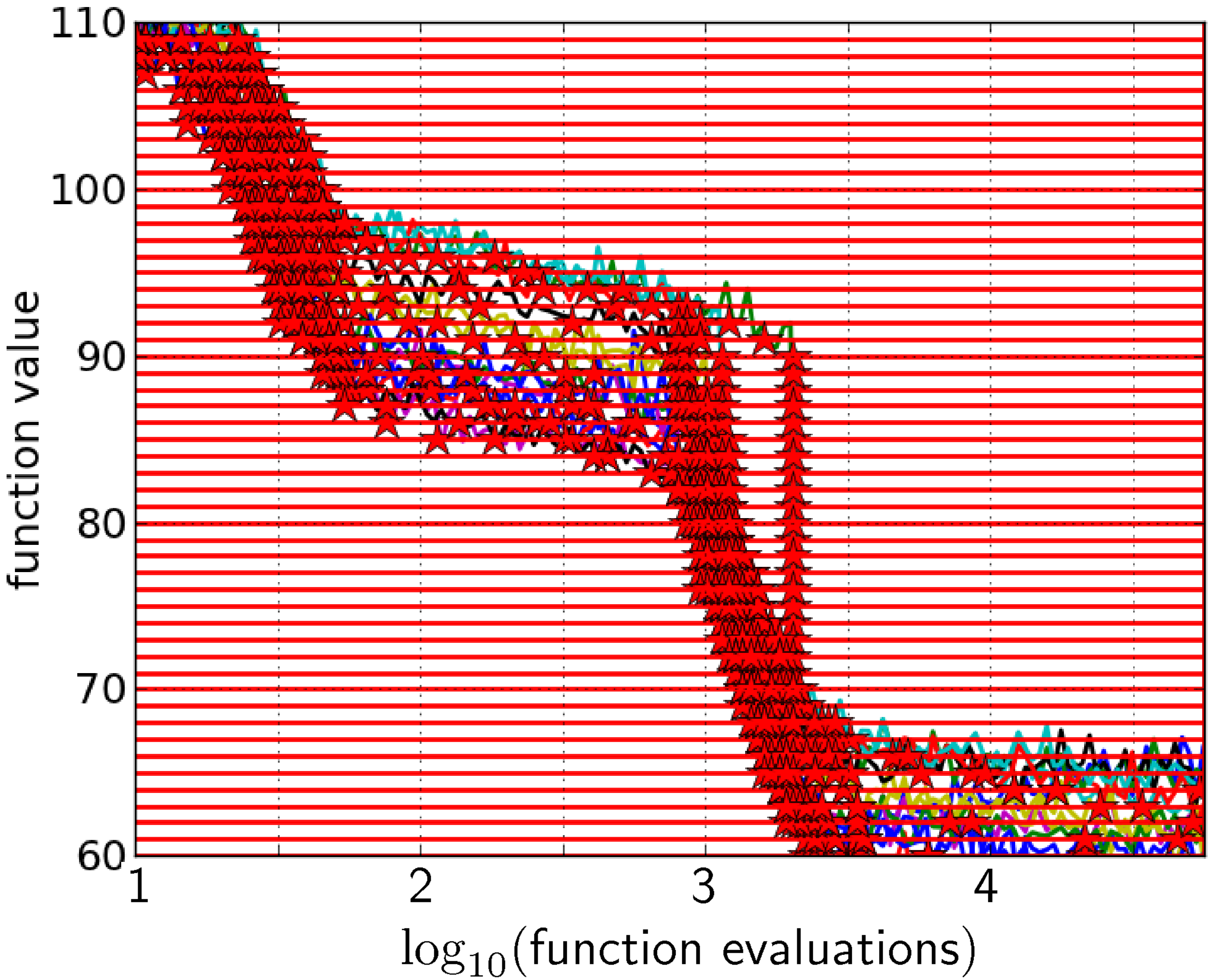
15 runs



the ECDF of run lengths (runtimes)

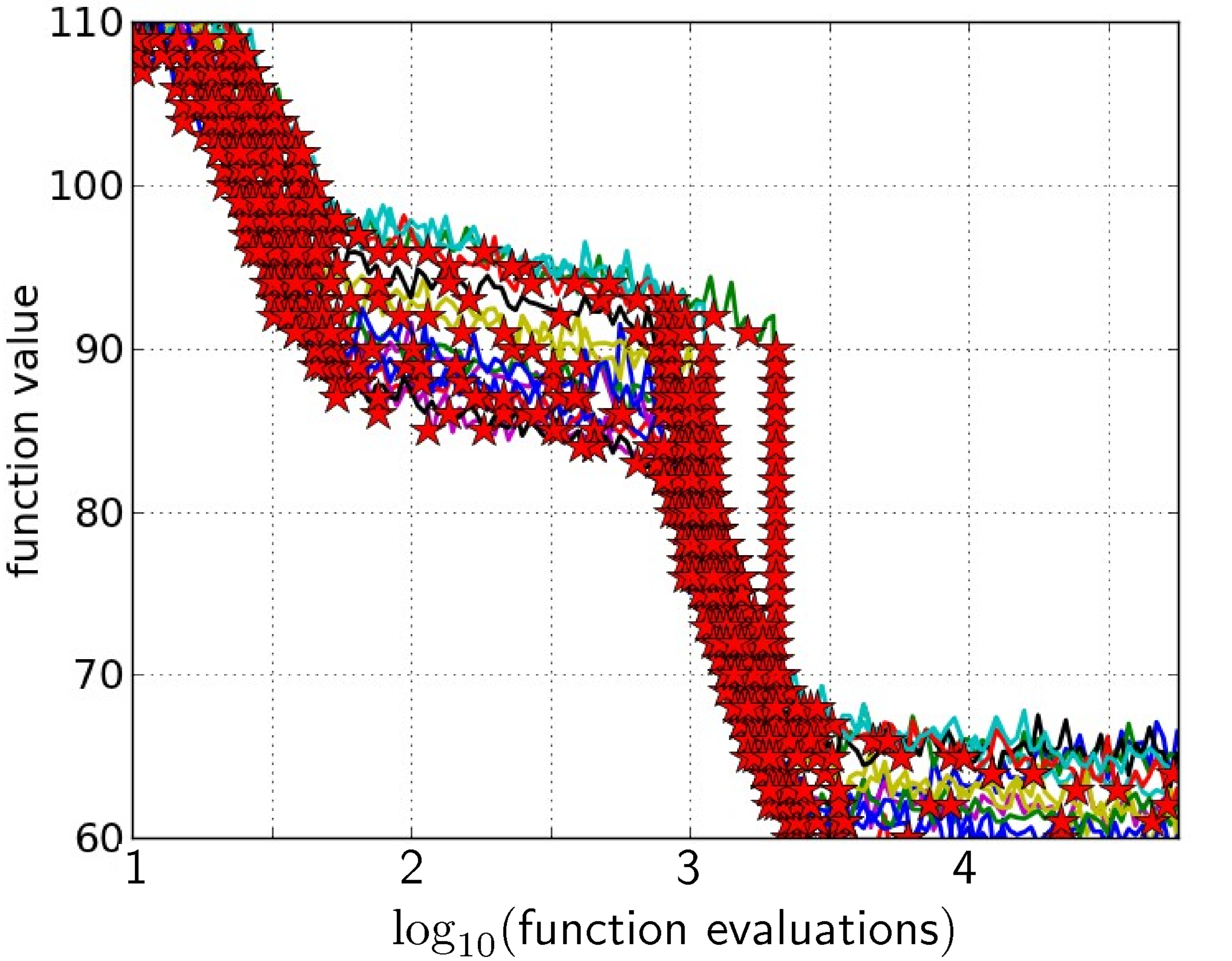

110
100
90
80
70
60
1
2
3
4
function value
$\log_{10}$(function evaluations)

15 runs



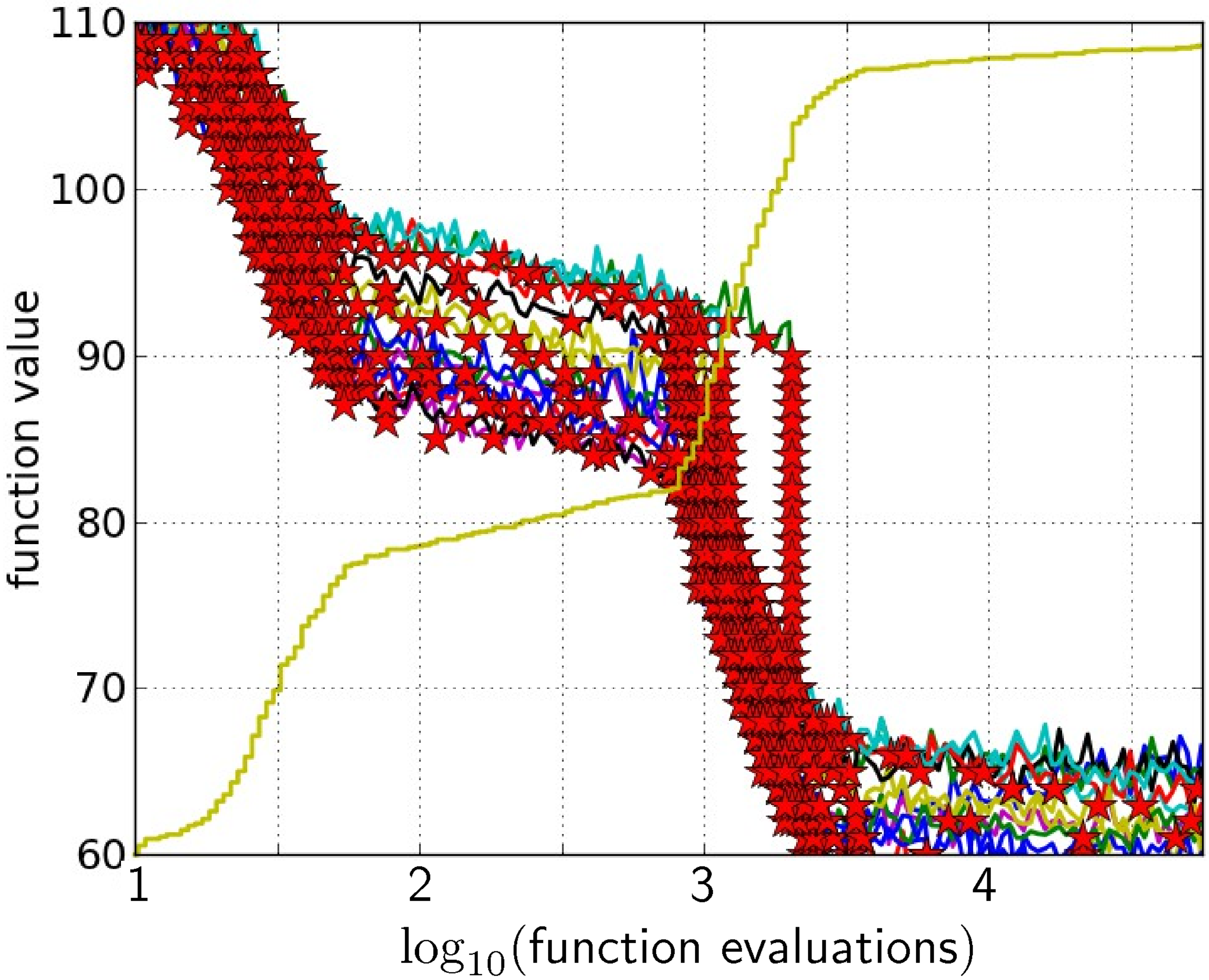
15 runs

50 targets



15 runs

50 targets

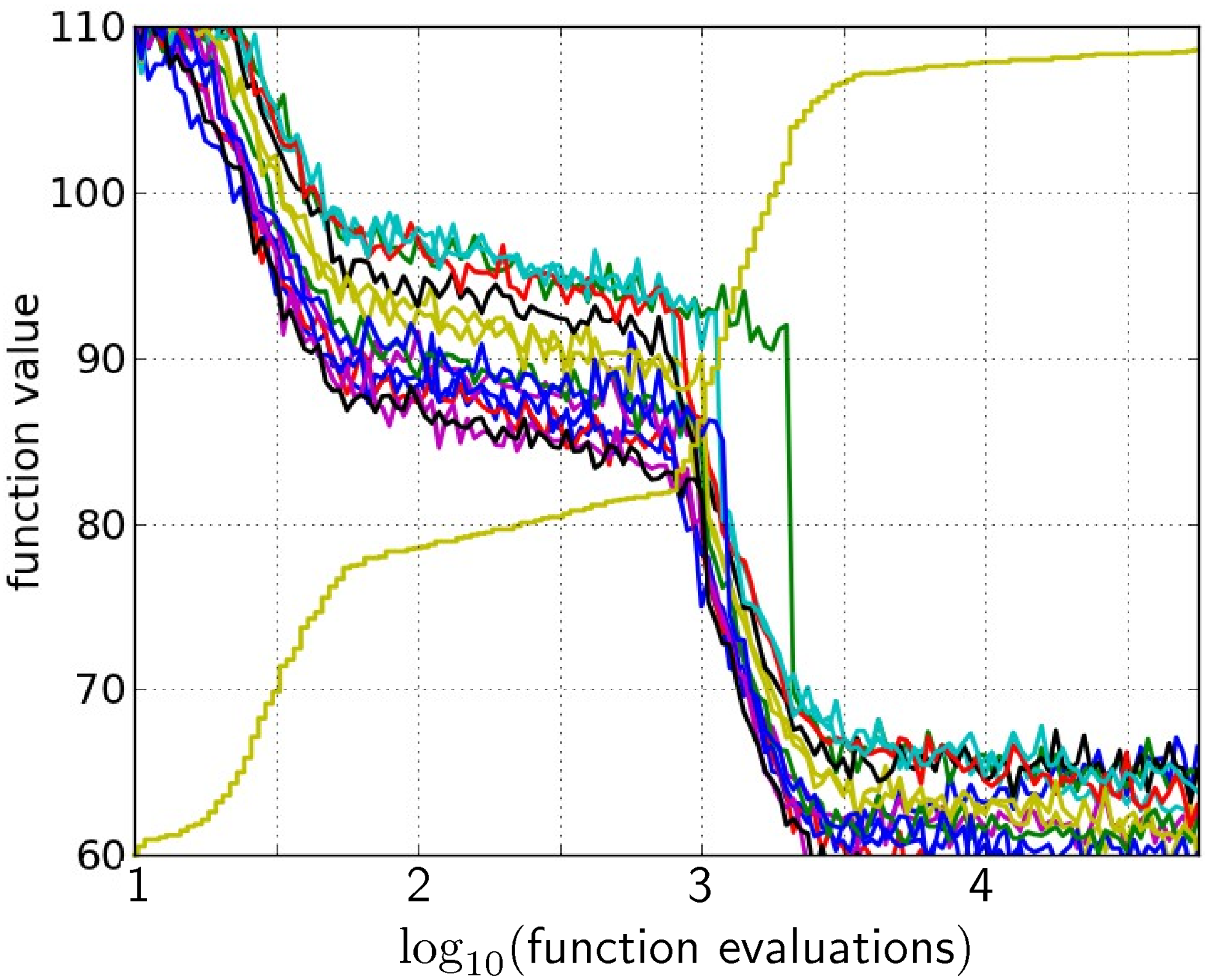


15 runs

50 targets

ECDF

.



15 runs integrated in a single graph

ECDF

empirical cumulative distribution functions

- recover a single convergence graph
- can aggregate over any set of functions and target values

they display a set of run lengths or runtimes (RT)

- for a single problem (function & target value) allow to estimate any statistics of interest from them, like median, expectation (ERT), … in a meaningful way

Critics

- the samples (evaluated solutions) do not reflect the real return value of the algorithm

conjecture: sampling can be consistently done far away from the estimated optimum
resolution: allow recommendations (incumbents)

- in noisy environments one cannot know the best value easily

resolution: allow recommendations (incumbents)

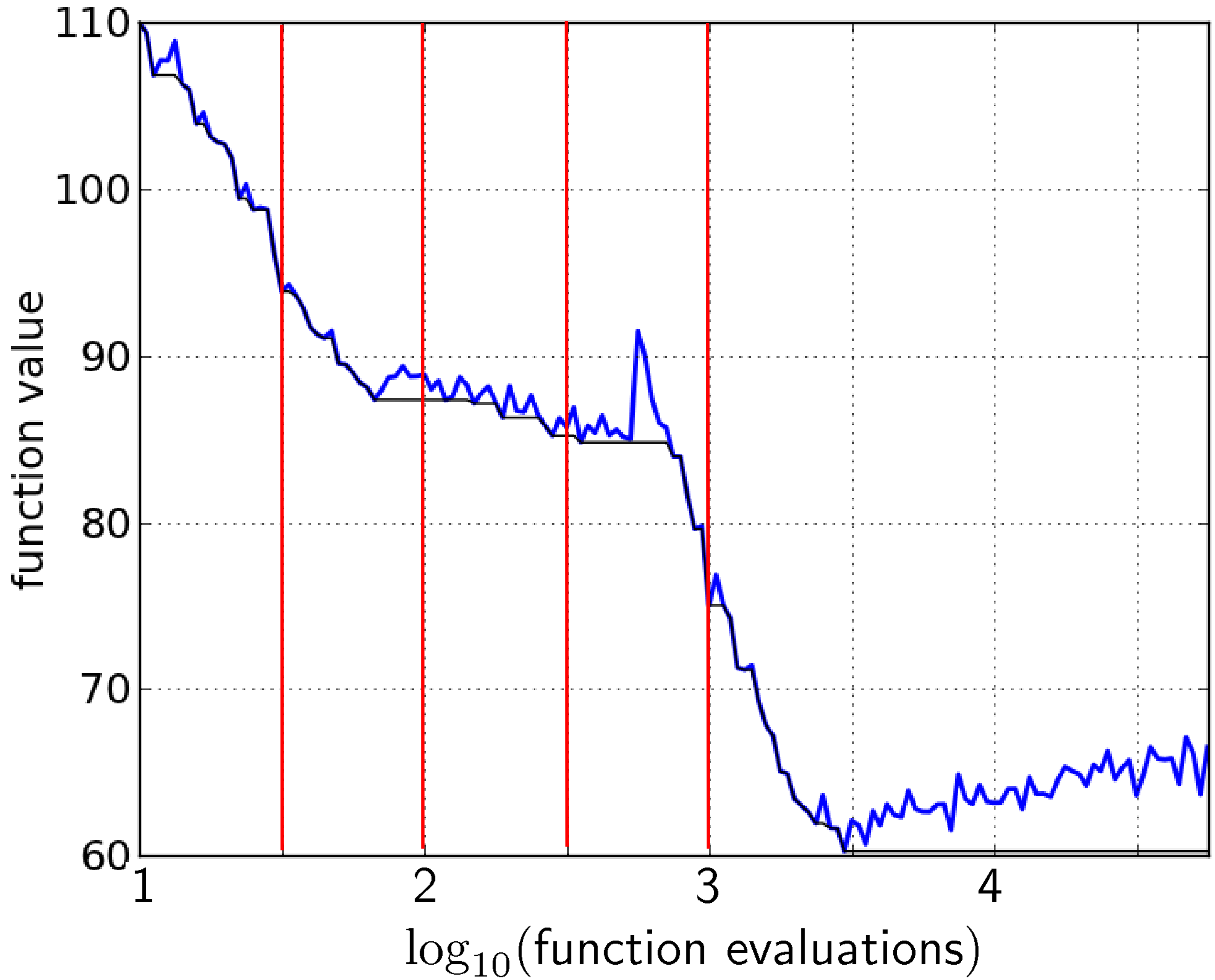
- in noisy environments the first hitting time is unrealistic, because the graph is not monotonous

conjecture: lucky punch is no rare exception

"Expensive" setting

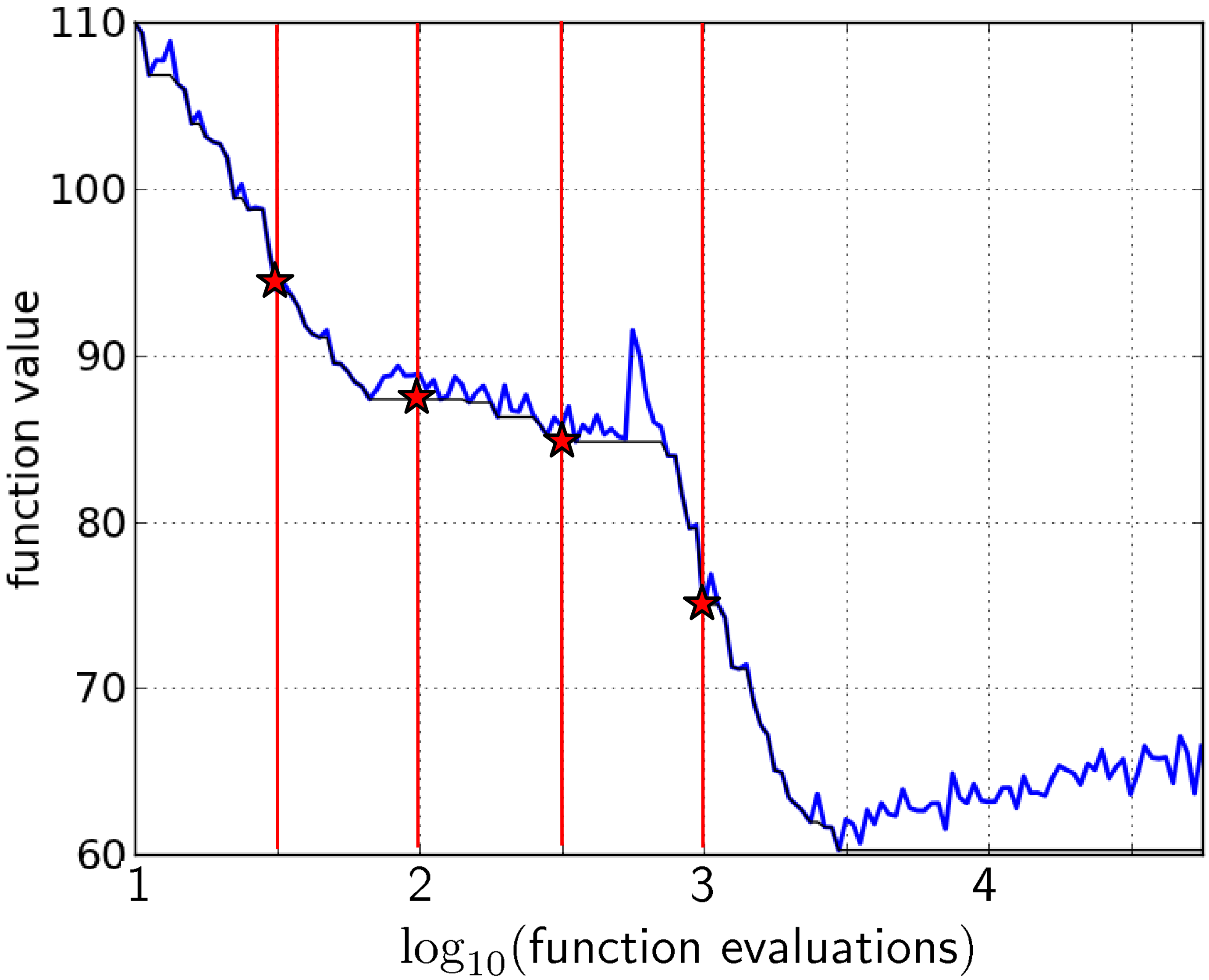
- Changed target values compared to the original setting
- Target values depend on the function and the dimension
- Computation is based on the run length of the GECCO BBOB 2009 best result ==> runlength-based target values

Target budgets/run-lengths



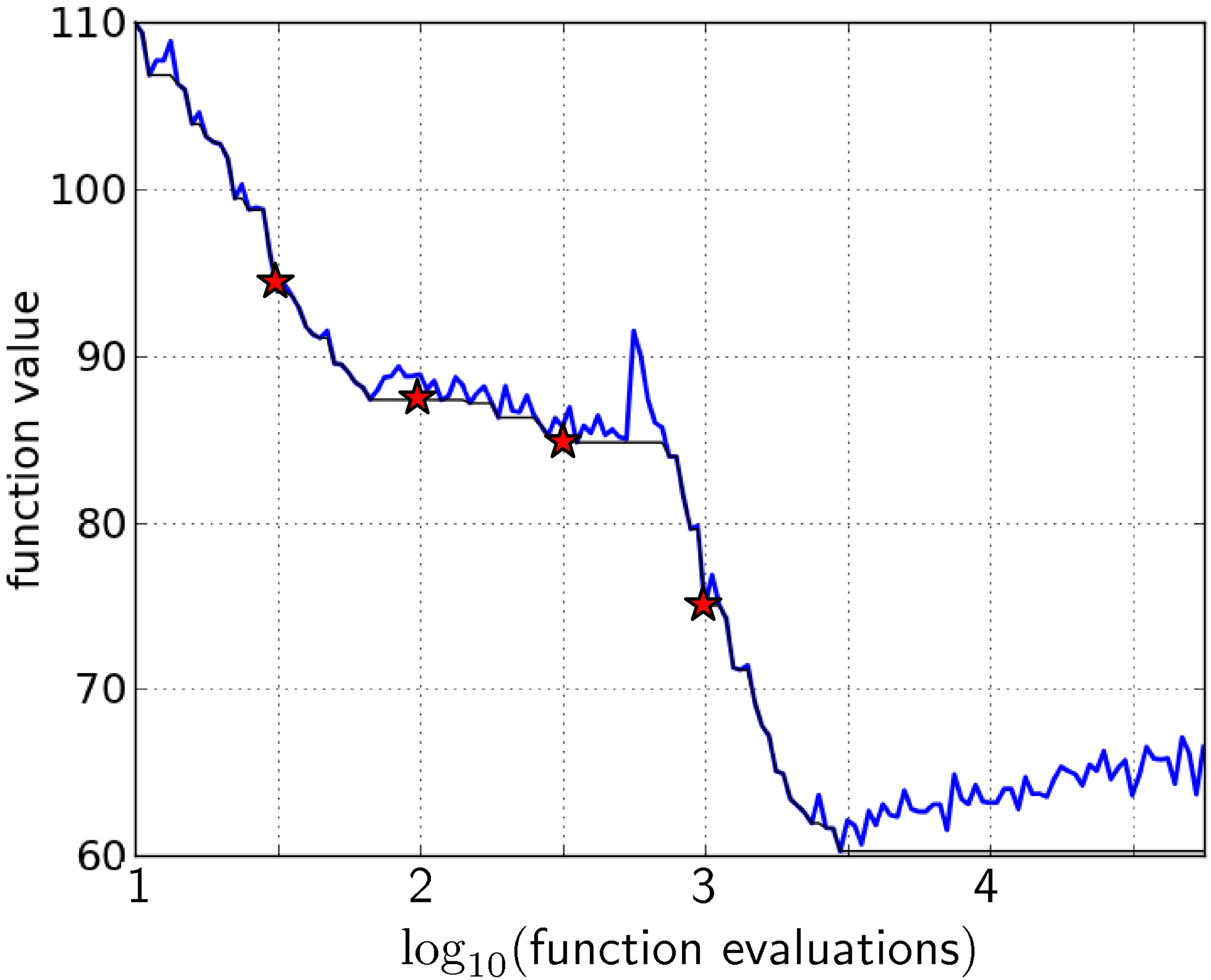
- first hitting time: a monotonous graph

Target budgets on the reference algorithm



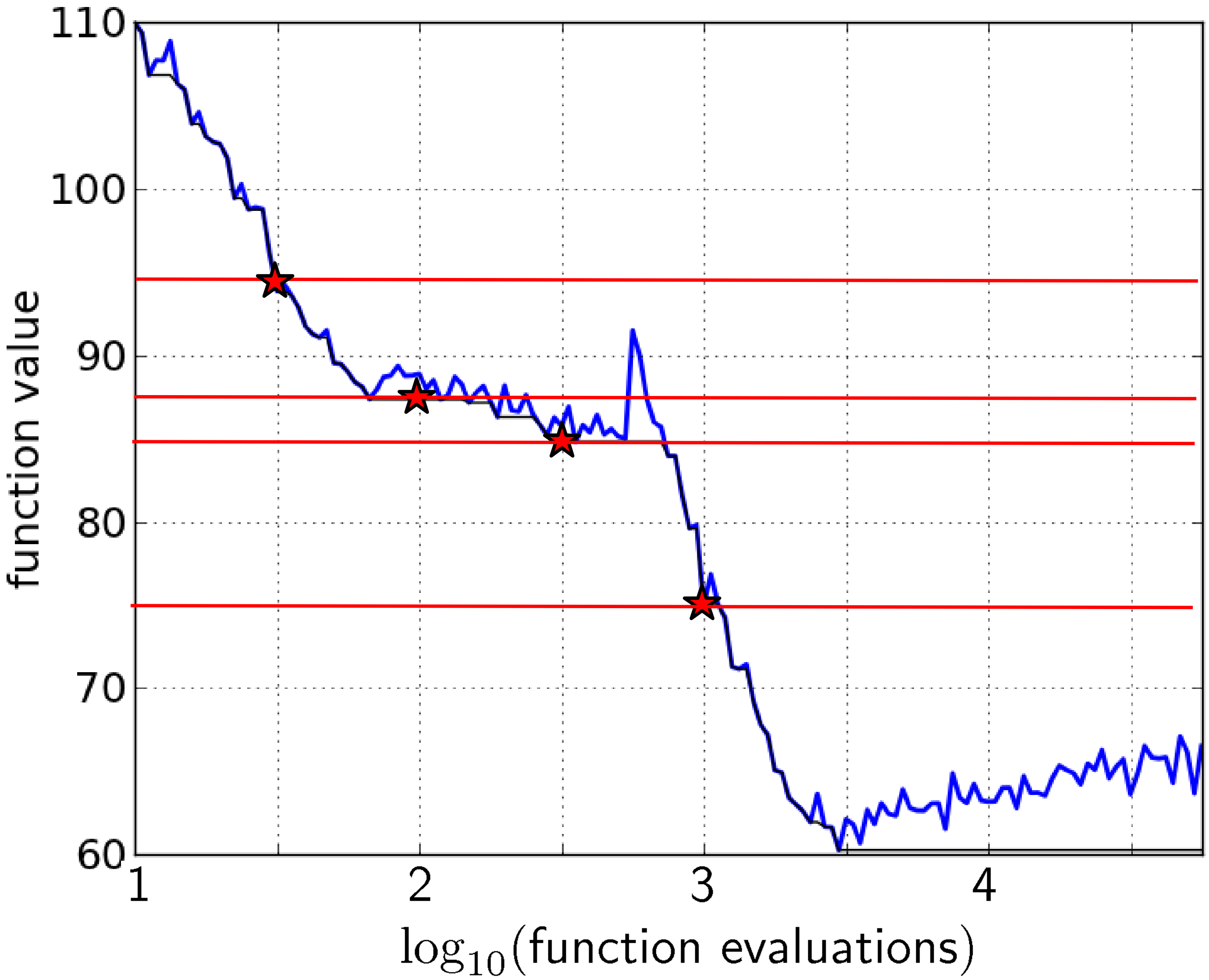
- first hitting time: a monotonous graph

Target budgets on the reference algorithm



- first hitting time: a monotonous graph

Run-length based target f-values



- first hitting time: a monotonous graph

.

Questions?