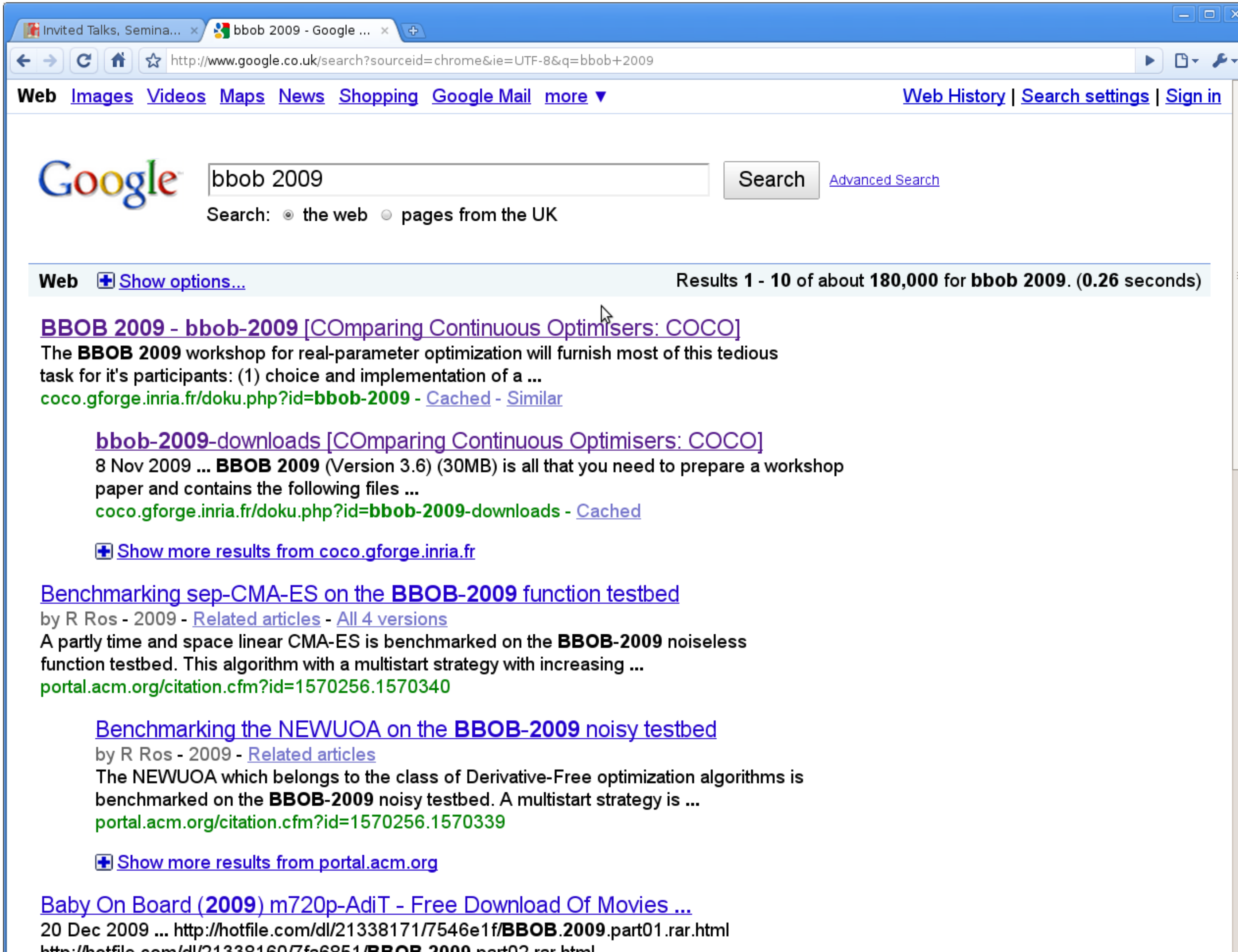
COCO: objectives

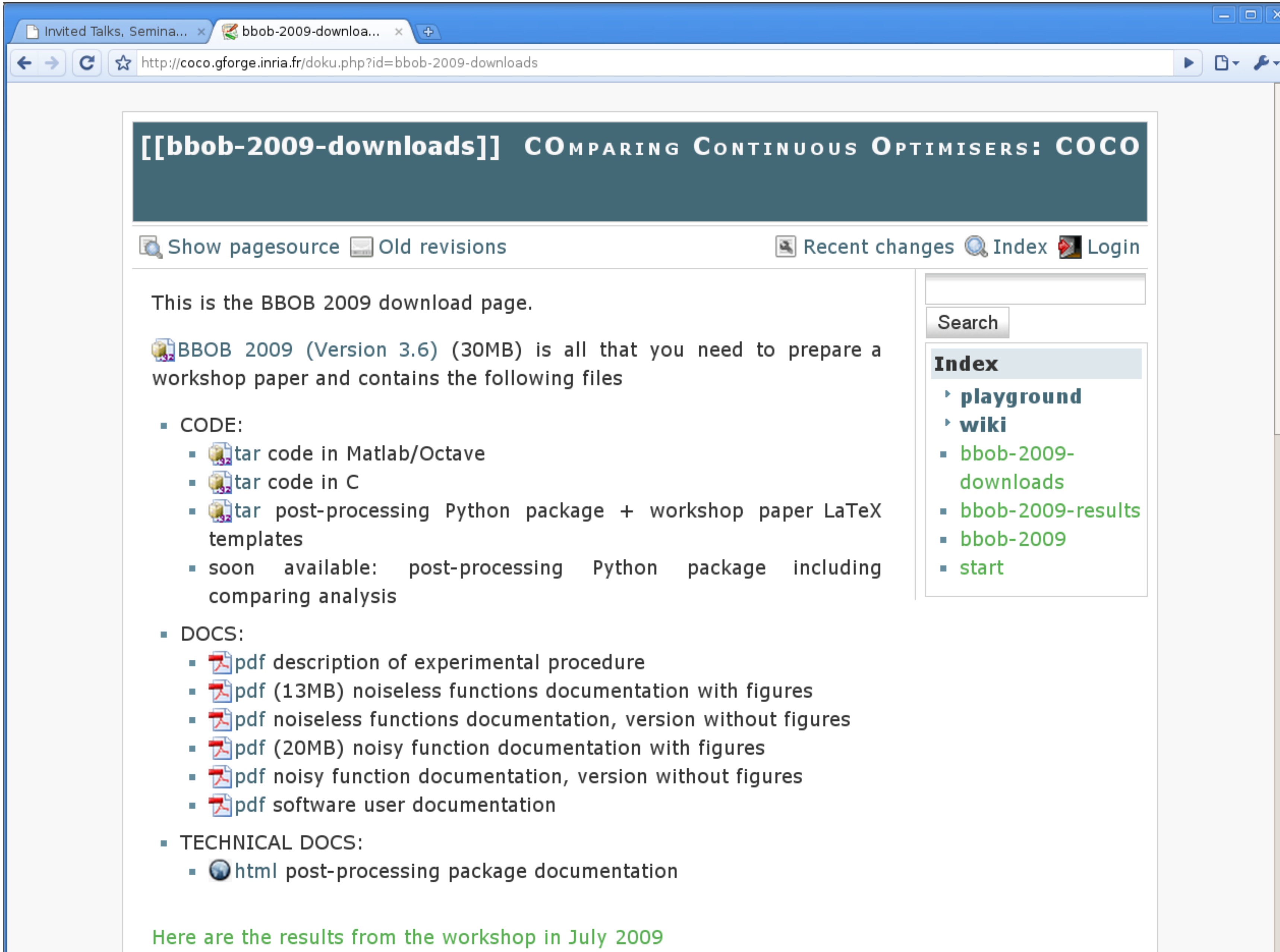
- function testbed:
  - should “reflect reality”
  - mainly non-convex and non-separable
  - scalable with the search space dimension
  - not too easy to solve, but yet comprehensible
- provide data acquisition at the interface of solver and objective function

  lean but sufficient data for quantitative analyses
- data presentation yields quantitative assessment, stratified by function properties...

BBOB in practice



BBOB in practice



Invited Talks, Semina... | bbob-2009-downloa...

http://coco.gforge.inria.fr/doku.php?id=bbob-2009-downloads

[[bbob-2009-downloads]] COMPARING CONTINUOUS OPTIMISERS: COCO

Show pagesource Old revisions Recent changes Index Login

This is the BBOB 2009 download page.

BBOB 2009 (Version 3.6) (30MB) is all that you need to prepare a workshop paper and contains the following files

- CODE:
  - tar code in Matlab/Octave
  - tar code in C
  - tar post-processing Python package + workshop paper LaTeX templates
  - soon available: post-processing Python package including comparing analysis
- DOCS:
  - pdf description of experimental procedure
  - pdf (13MB) noiseless functions documentation with figures
  - pdf noiseless functions documentation, version without figures
  - pdf (20MB) noisy function documentation with figures
  - pdf noisy function documentation, version without figures
  - pdf software user documentation
- TECHNICAL DOCS:
  - html post-processing package documentation

Here are the results from the workshop in July 2009

Search

Index

- **playground**
- **wiki**
- bbob-2009-downloads
- bbob-2009-results
- bbob-2009
- start

BBOB in practice

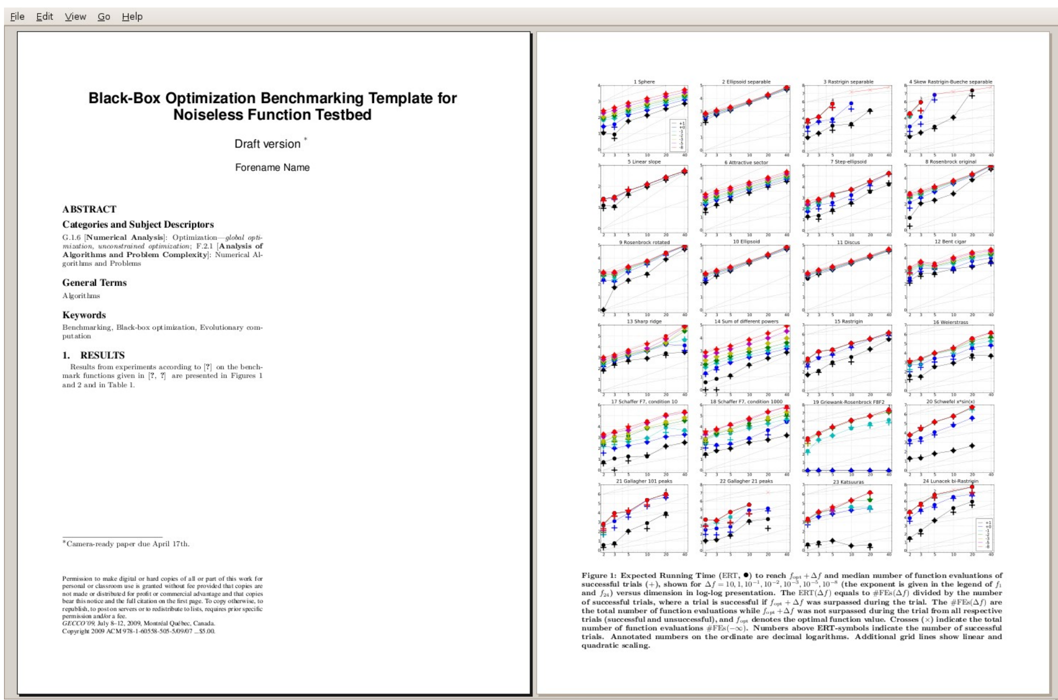
Matlab script:

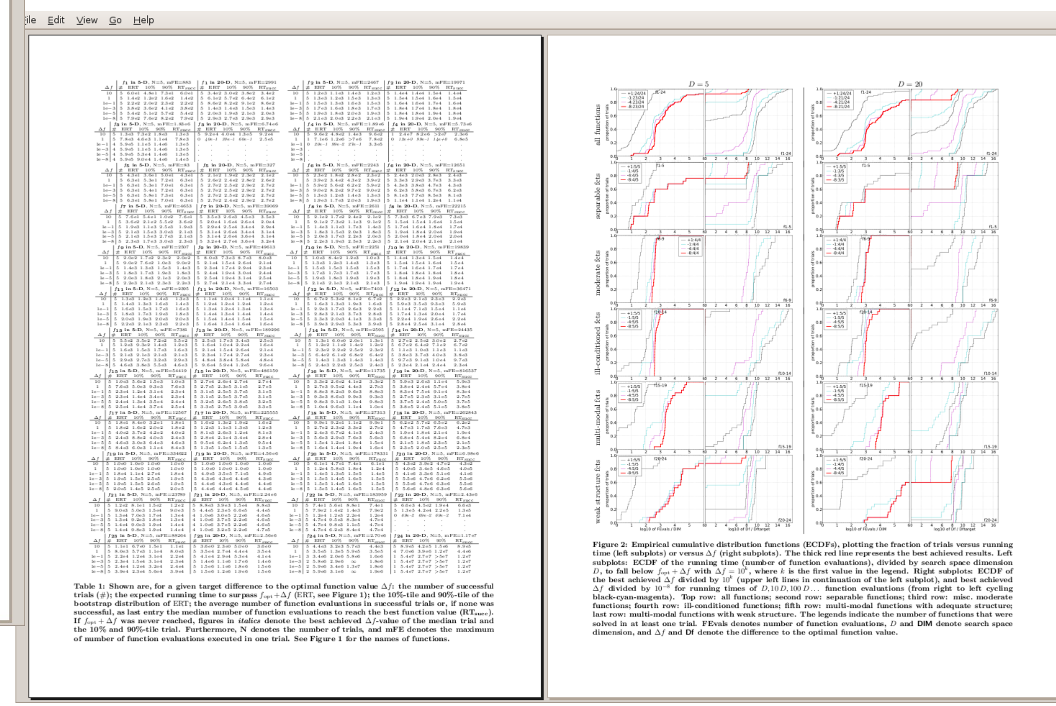
```
for dim = [2,3,5,10,20,40]  % small dimensions first, for CPU reasons
  for ifun = benchmarks('FunctionIndices')  % or benchmarksnoisy(...)
    for iinstance = [1:5, 1:5, 1:5]  % first 5 fct instances, three times
      fgeneric('initialize', ifun, iinstance, datapath);

      MY_OPTIMIZER('fgeneric', dim, ...   % necessary parameters
                   fgeneric('ftarget'));  % optional termination parameter

      fgeneric('finalize');
    end
    disp(['      date and time: ' num2str(clock, ' %.0f')]);
  end
  disp(sprintf('---- dimension %d-D done ----', dim));
end
```

BBOB in practice

Post-processing at the OS shell:

```
python codepath/bbob_pproc/run.py datapath
pdflatex templateACMarticle.tex
```



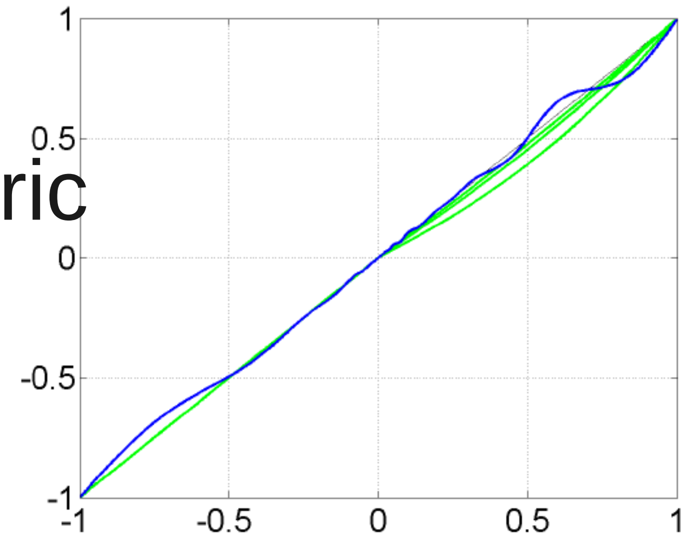


COCO: the noiseless functions

24 functions within five sub-groups

- Separable functions
- Essential unimodal functions
- Ill-conditioned unimodal functions
- Multimodal structured functions
- Multimodal functions with weak or without structure

functions are not perfectly symmetric and are locally deformed



COCO: the noisy functions

three noise-”models”, so-called:

- Gauss, Uniform (severe), Cauchy (outliers)
- Utility-free noise

$$E(f(x)) \leq E(f(y)) \Rightarrow U(f(x)) \leq U(f(y)) \ \forall x, y, U$$

30 functions with three sub-groups

- 2x3 functions with weak noise
- 5x3 unimodal functions
- 3x3 multimodal functions

How should we measure performance?

Evaluation of Search Algorithms

needs

- Meaningful quantitative measure on benchmark functions or real world problems
- Account for meta-parameter tuning

tuning to specific problems can be quite expensive

- Account for invariance properties

prediction of performance is based on "similarity", ideally equivalence classes of functions

- Account for algorithm internal costs

often negligible, depending on the objective function cost

A performance measure

should be

- quantitative, with a ratio scale
- well-interpretable with a meaning
- relevant in the “real world”
- simple

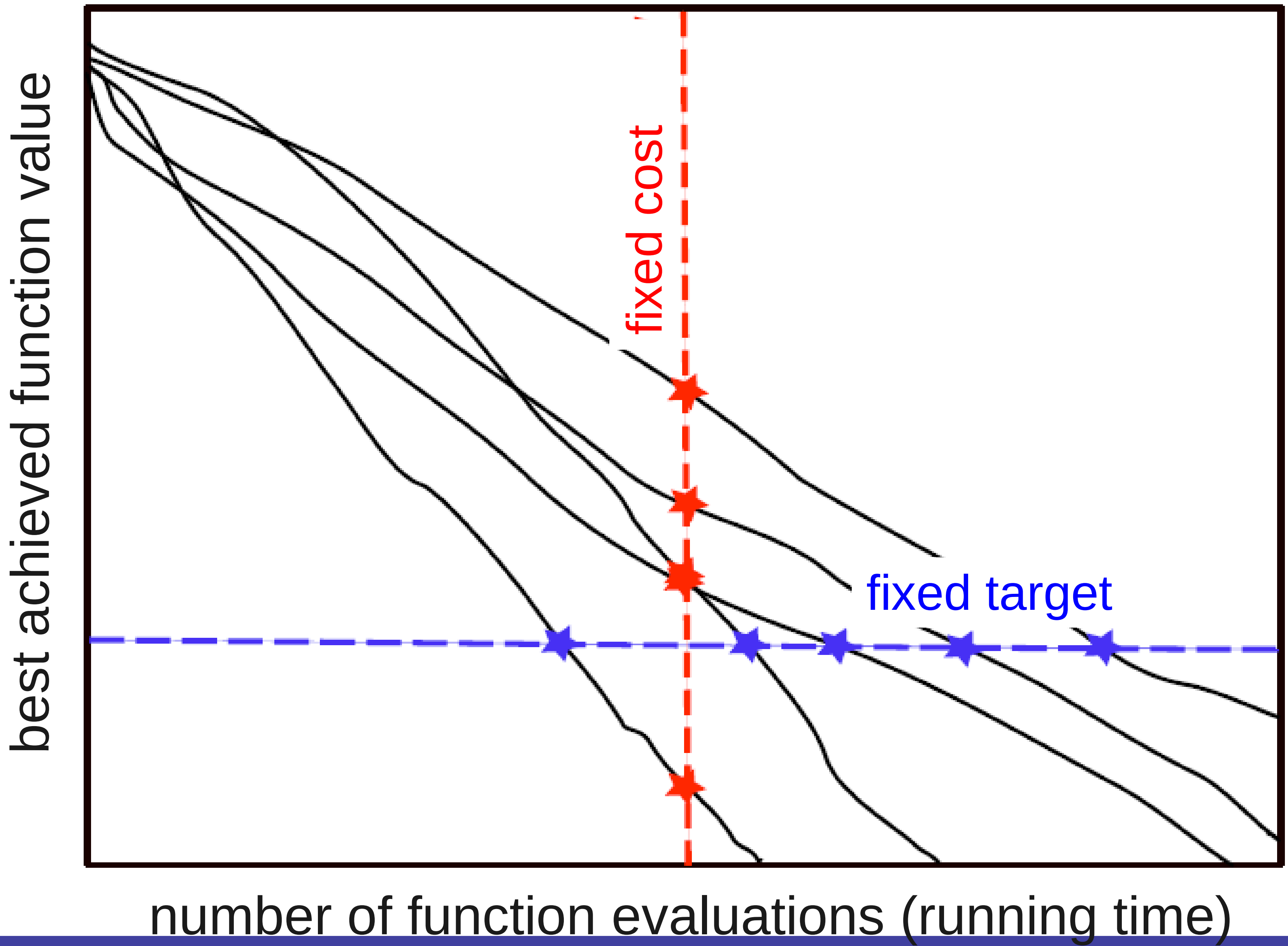
(recall) Black-Box Optimization

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

How should we measure performance?

fixed-cost versus fixed-target



A performance measure

should be

- quantitative, with a ratio scale
- well-interpretable with a meaning
- relevant in the “real world”
- simple

running time

- empirical distribution [Hoos & Stützle 1998]
- expectation, median, ...

Runtime

We measure runtime in number of function evaluations

- As a distribution of runtimes
- As expected runtime ERT

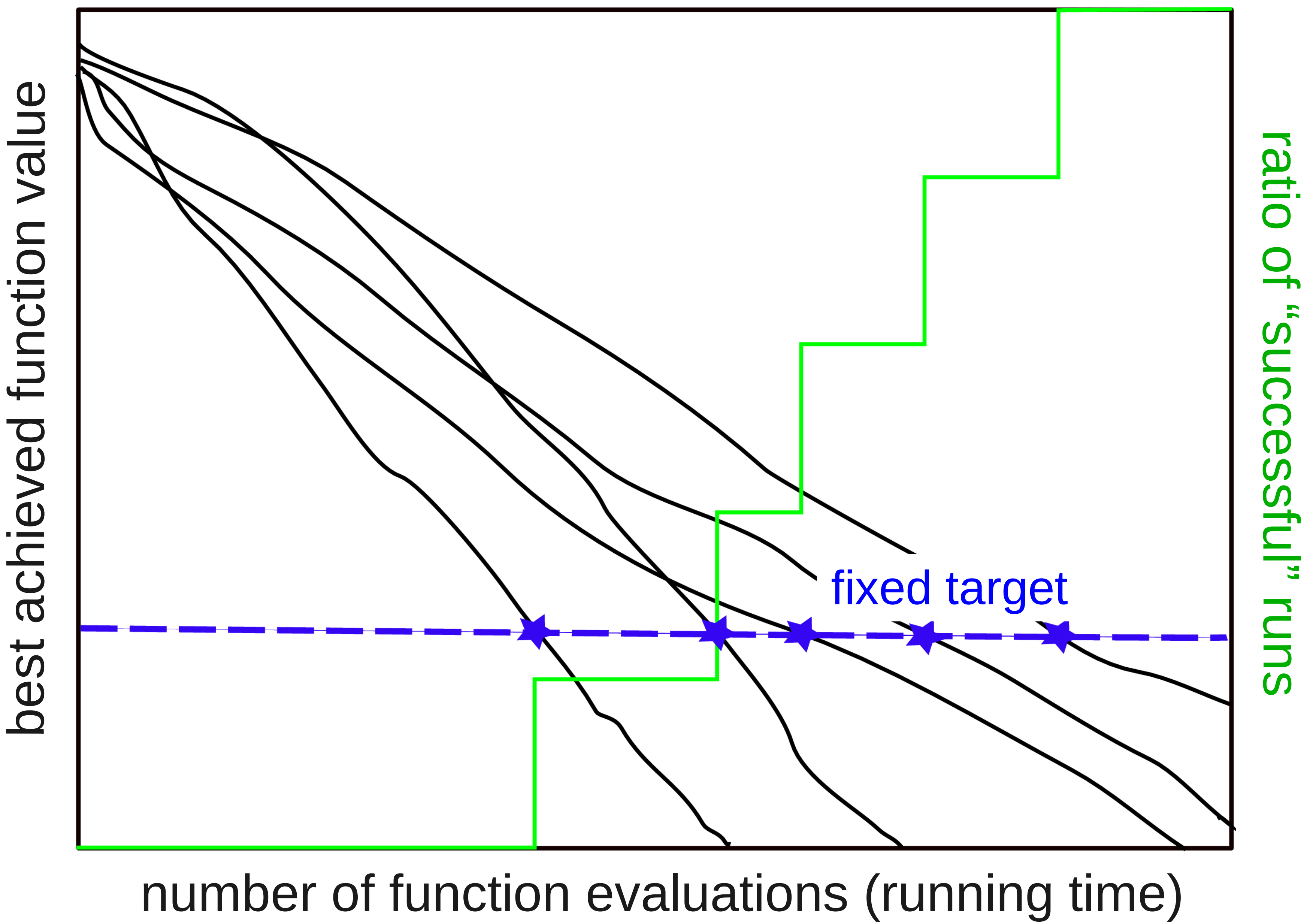
For success probability $0 < p < 1$: (simulated) restarts until a successful run is observed.

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

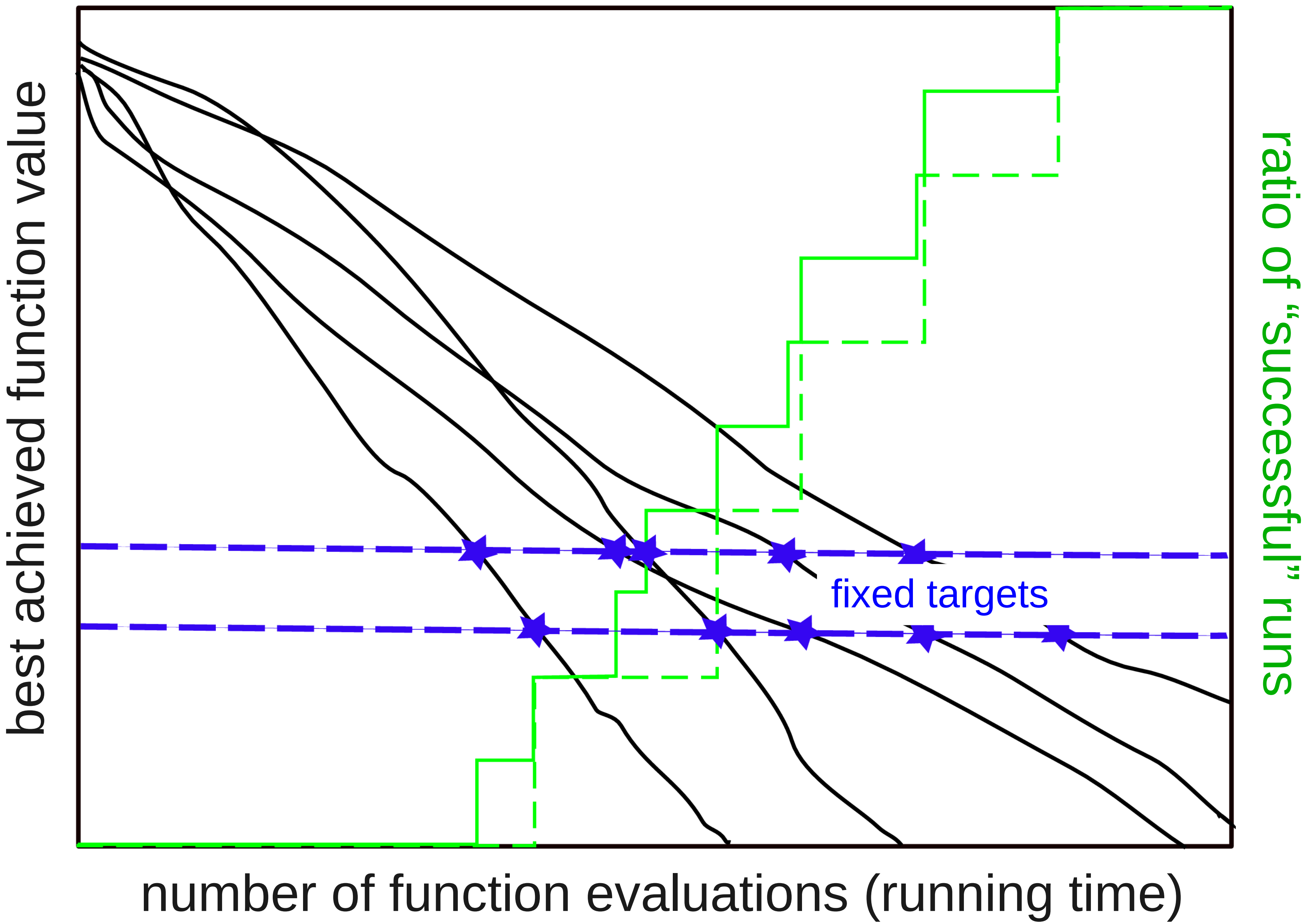
$$\approx E(\mathsf{RT}_{\mathrm{succ}}) + \frac{1-p}{p} E(\mathsf{RT}_{\mathrm{unsucc}})$$

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

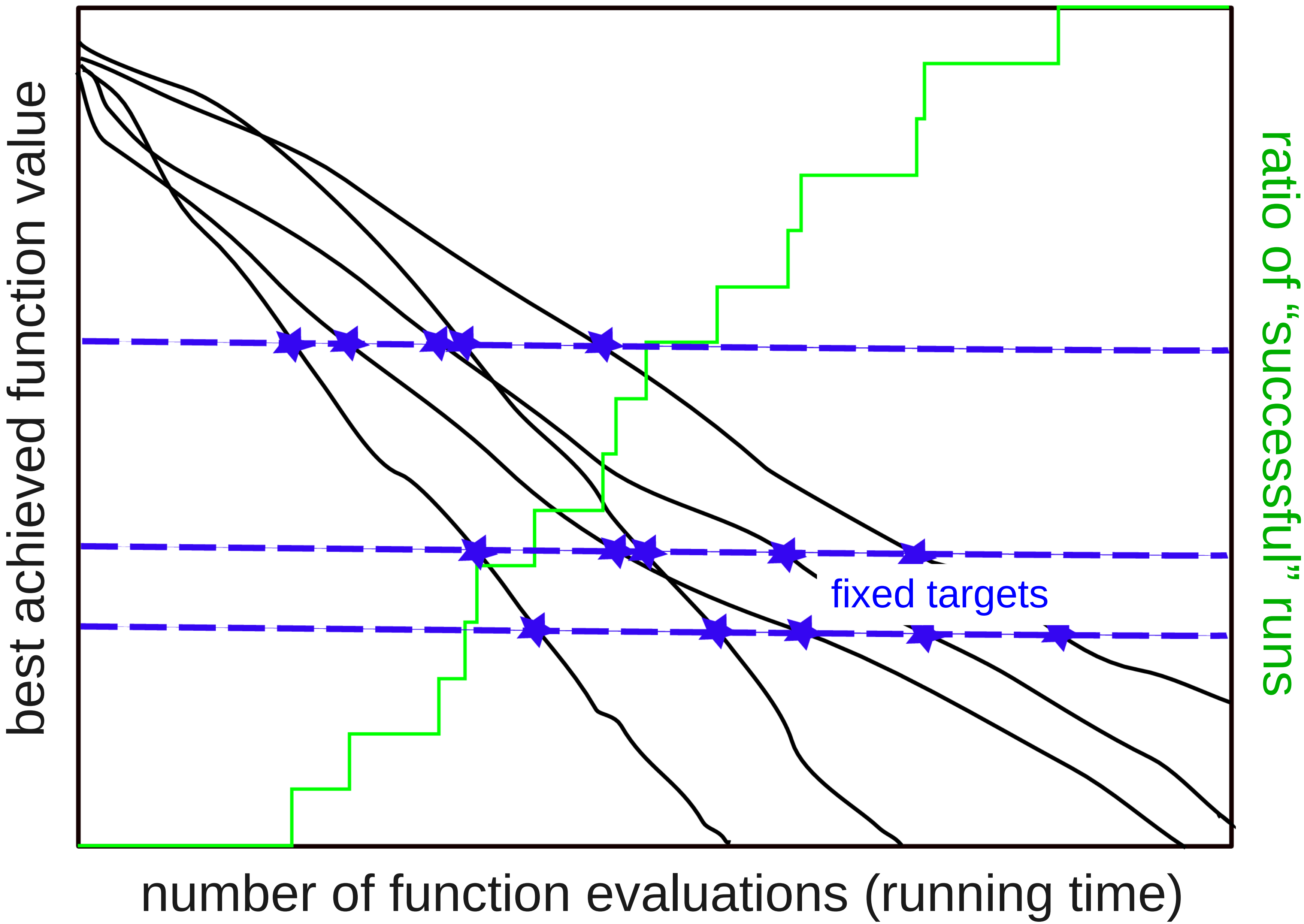
Measuring Performance with given target values



Measuring Performance with given target values



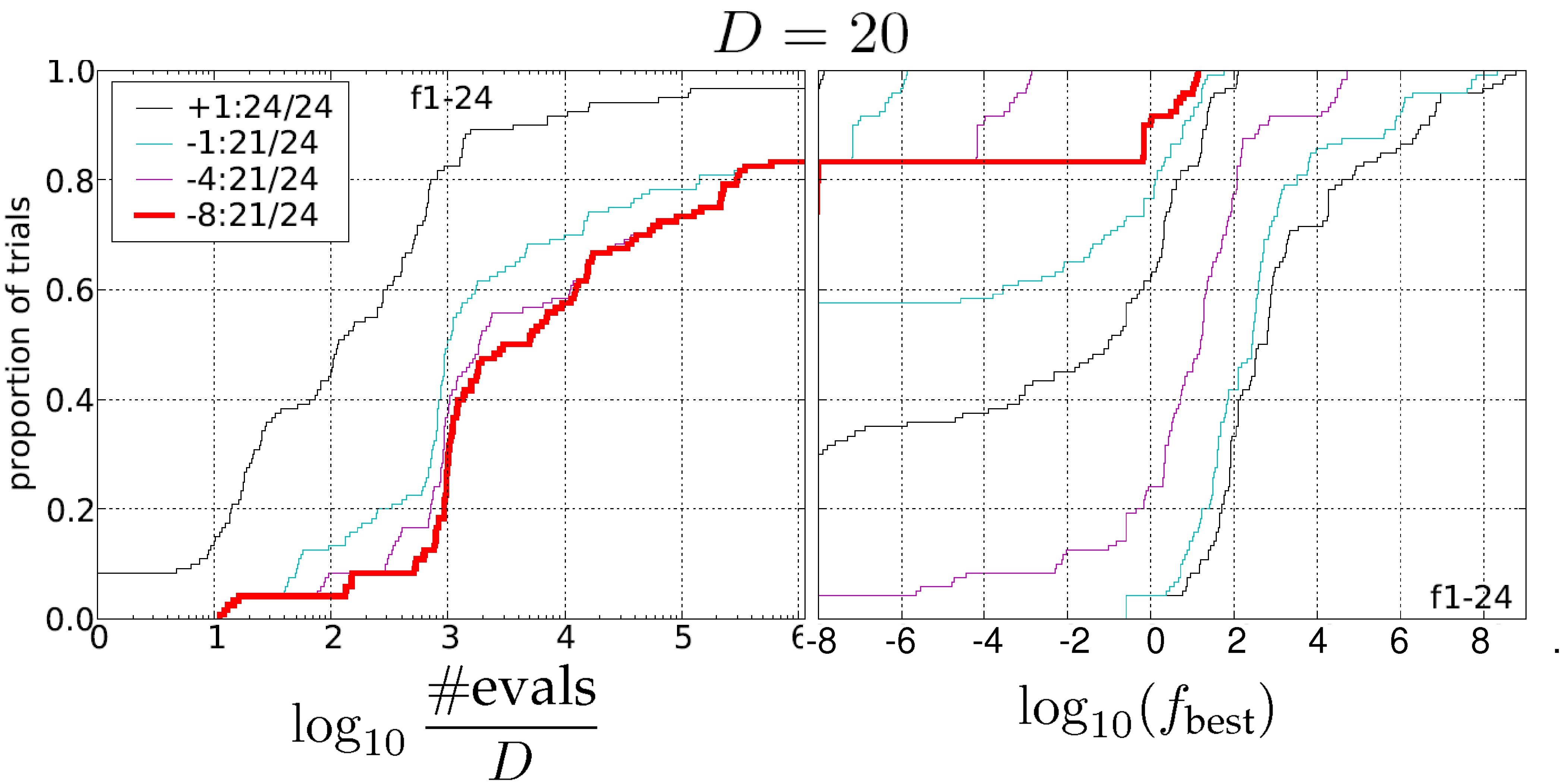
Measuring Performance with given target values



Cumulative Distribution of Runtimes

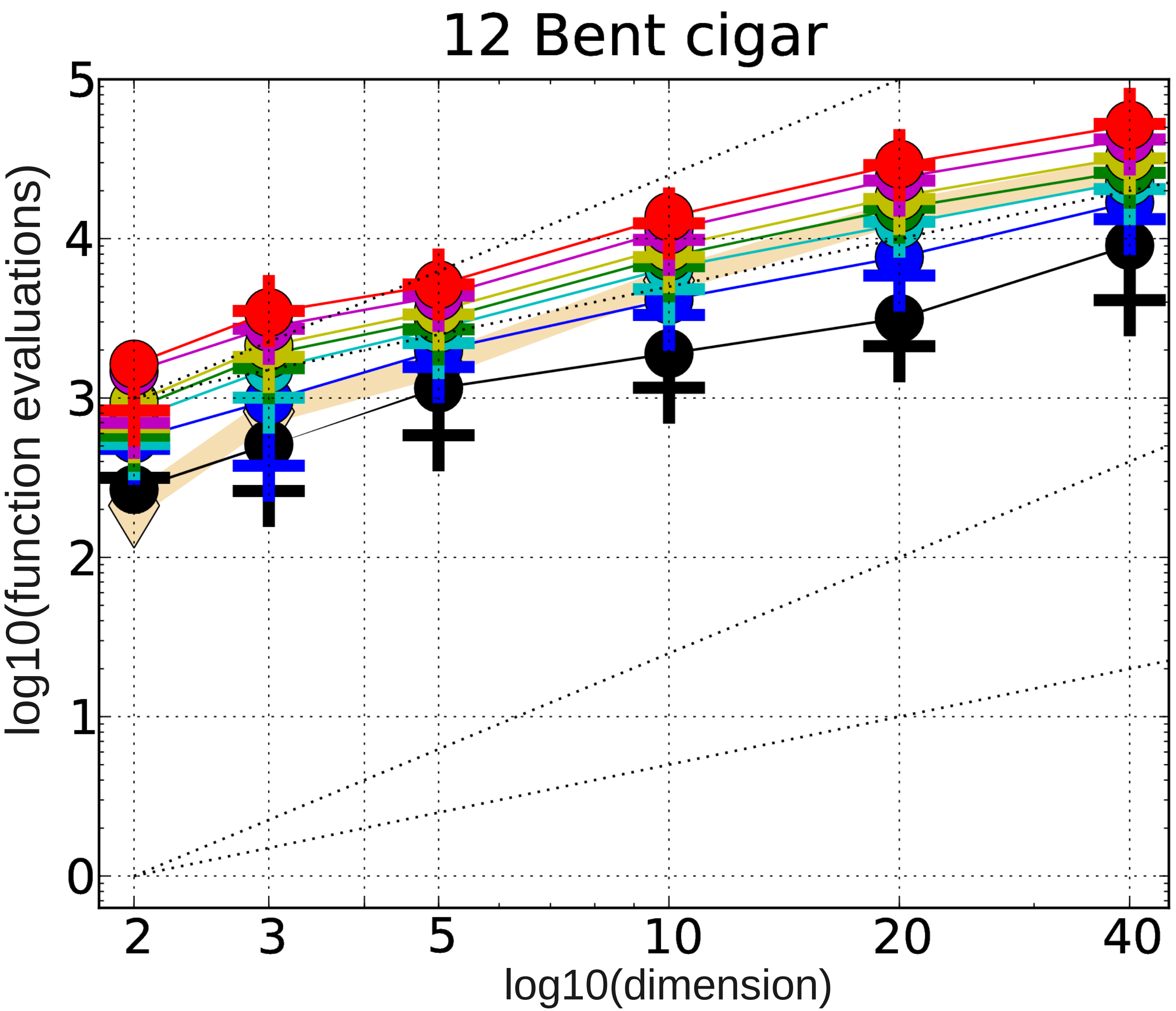
- Given a set of functions and for each function a (weighted) set of target values, the cumulative distribution of (simulated) RTs captures all(?) aspects of the performance in a single graph
- Remark: this performance measure can aggregate over any set of functions and target values
- Here: 50 target values, log-uniform in [1e-8,100] and 15 trials per function

Example for ECDFs



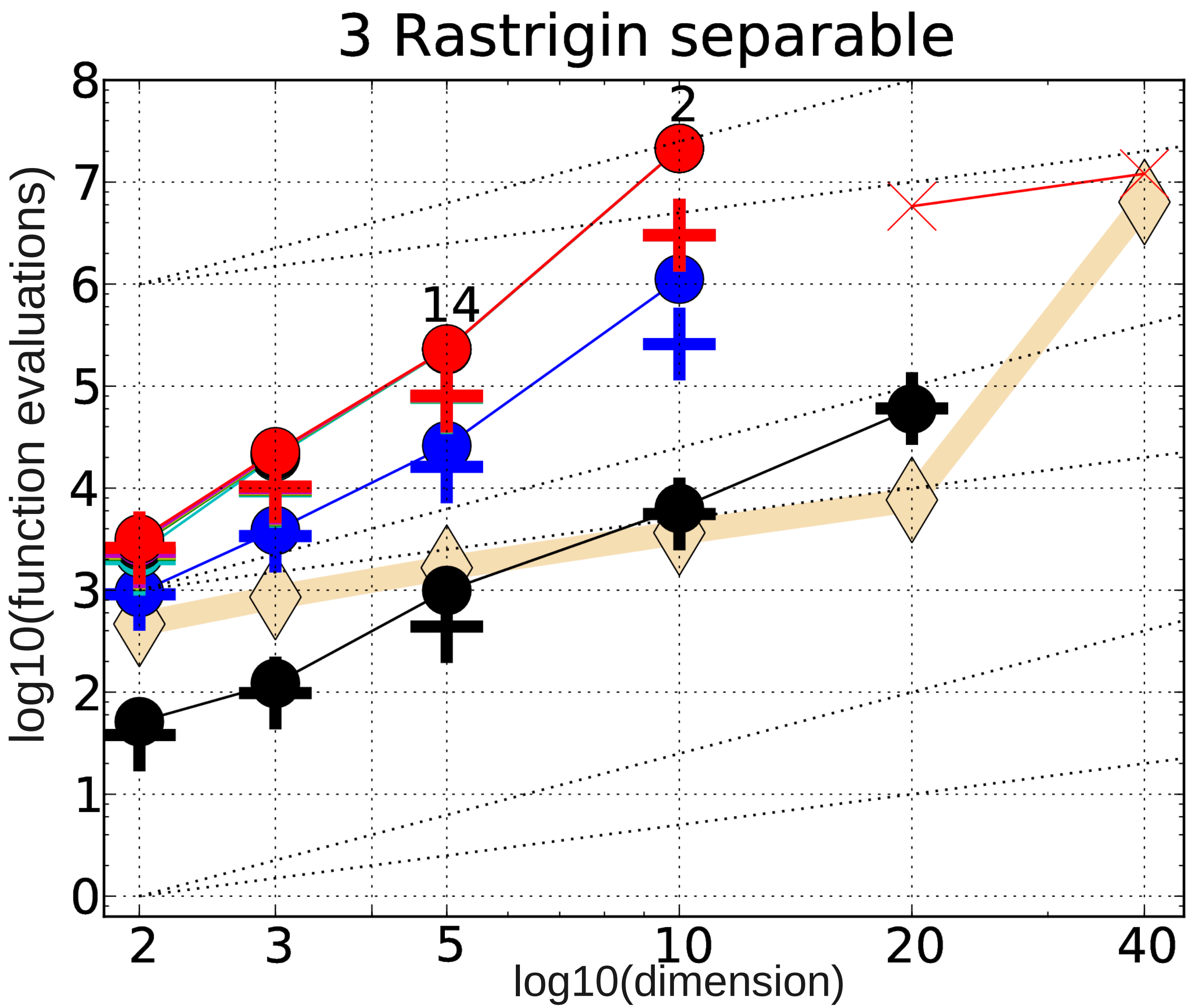
Empirical cumulative distribution functions (ECDFs) of running lengths (left) and function values (right)

Example: Scaling Behaviour



- ERT on f12: linear scaling of BIPOP-CMA-ES

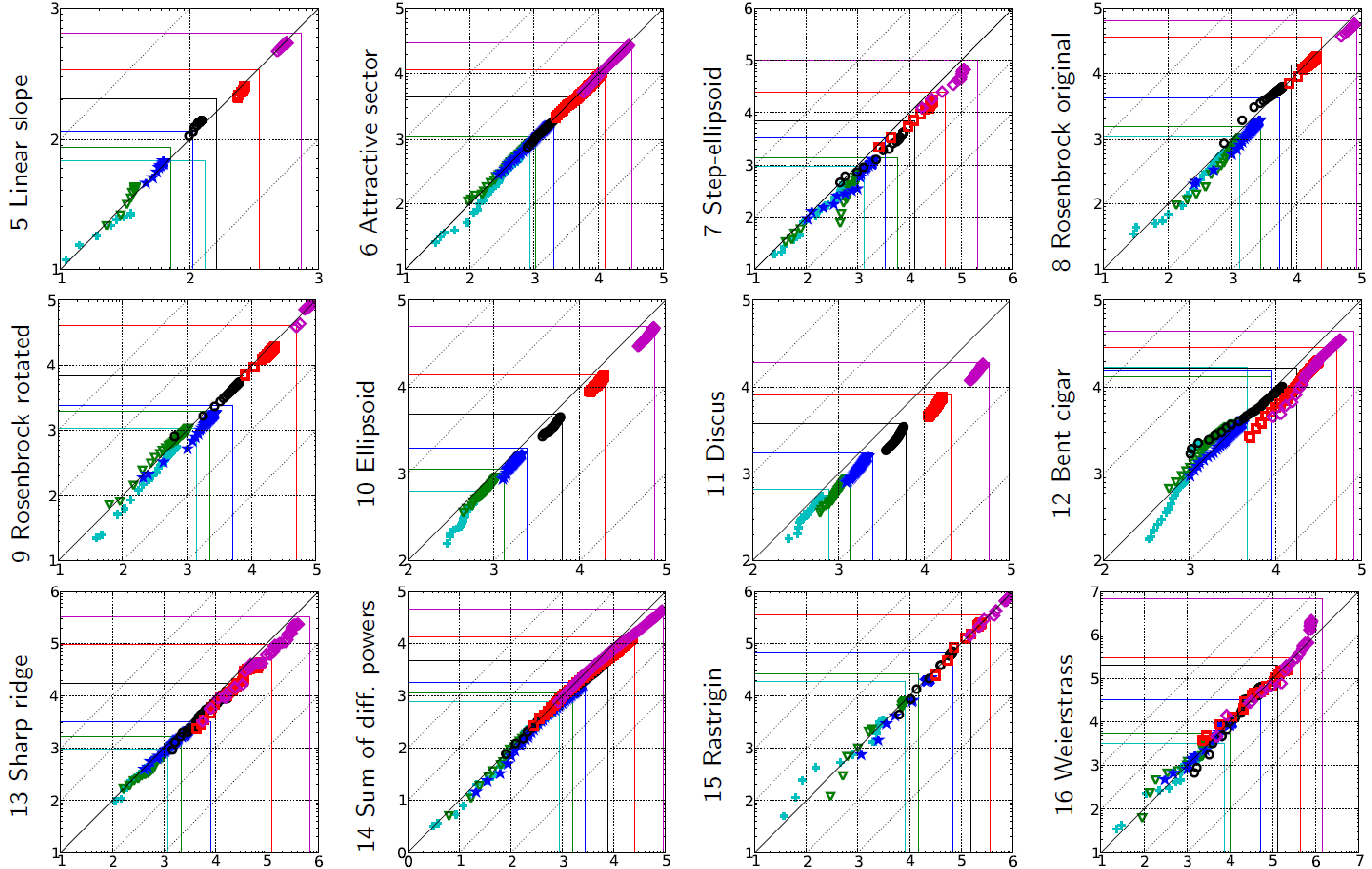
Example: Scaling Behaviour



- Experiments in >100-D are more often than not virtually superfluous

ERT scatter plots comparing two algorithms

all dimensions & targets



Overall Collected Data Sets

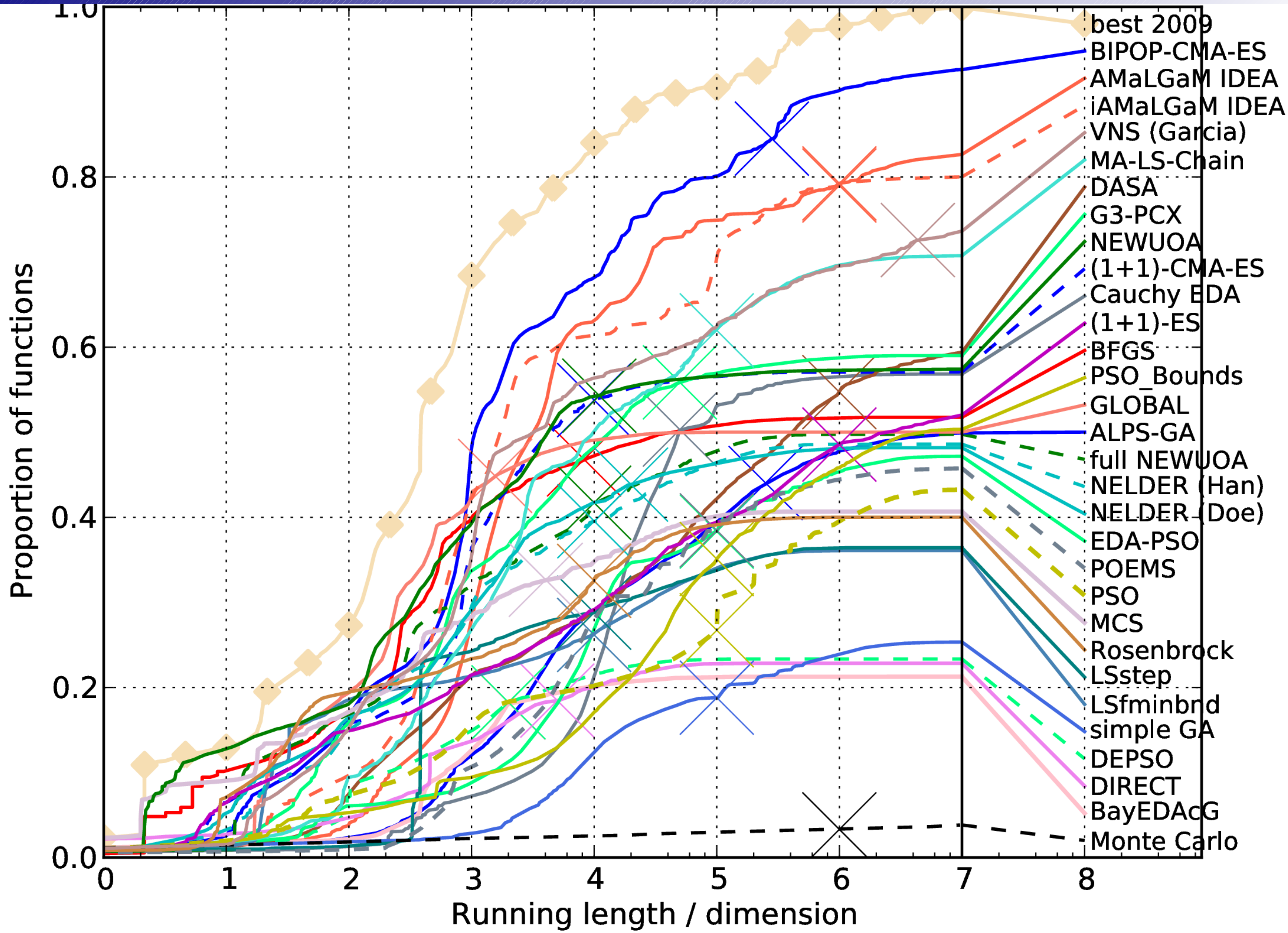
during the *Black-Box Optimization Benchmarking* (BBOB) workshops at the *Genetic and Evolutionary Computation Conference* GECCO

- 2009: 31 noiseless and 21 noisy “data sets”
- 2010: 24 noiseless and 16 noisy “data sets”

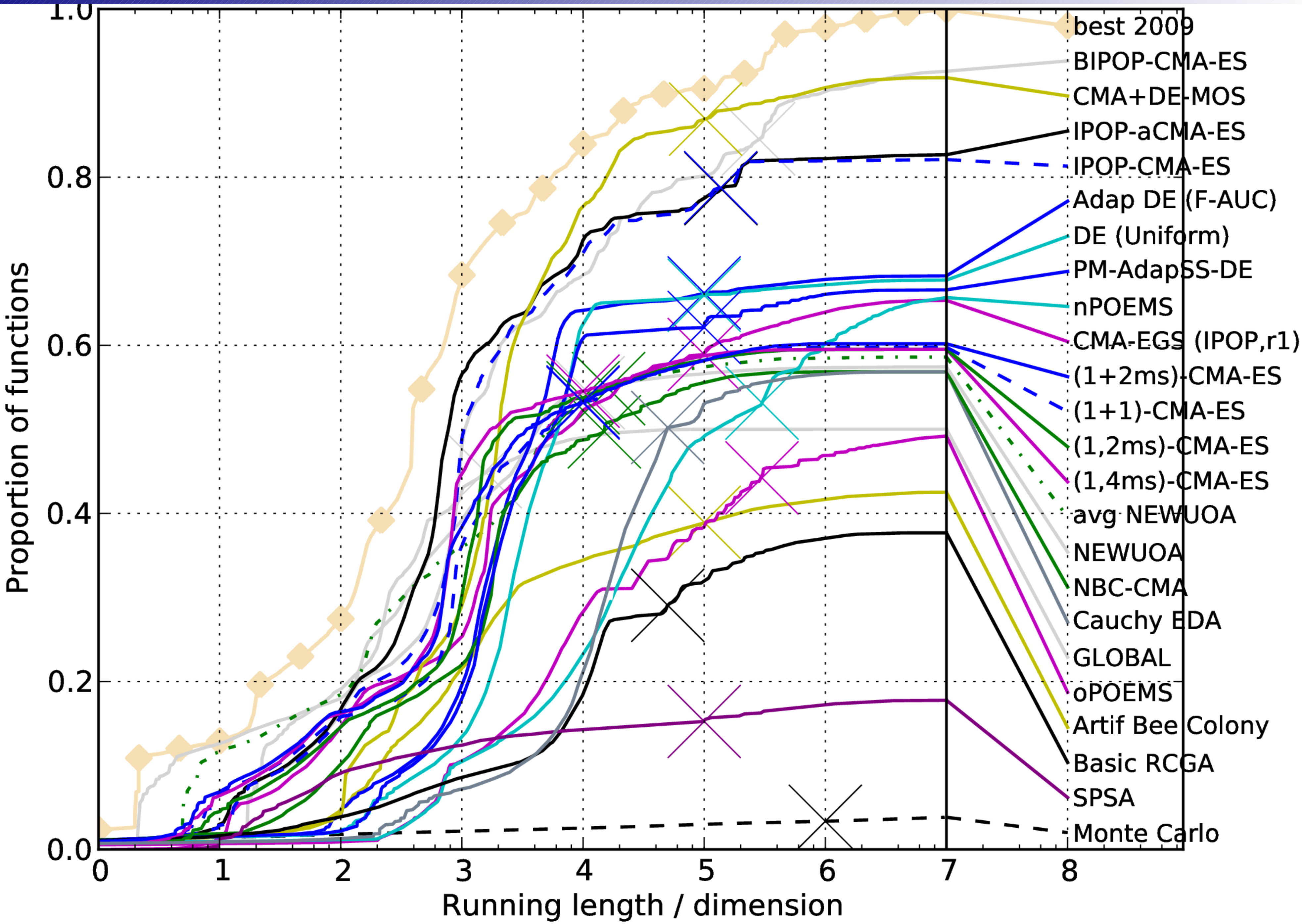
- Algorithms: RCGAs (eg plain, PCX), EDAs (eg IDEA), BFGS & (many) other “classical” methods, ESs (eg CMA), PSO, DE, Ant-Stigmergy Alg, Bee Colony, EGS, SPSA, Meta-Strategies...

Results

Results of 2009 (noisefree, 20-D)

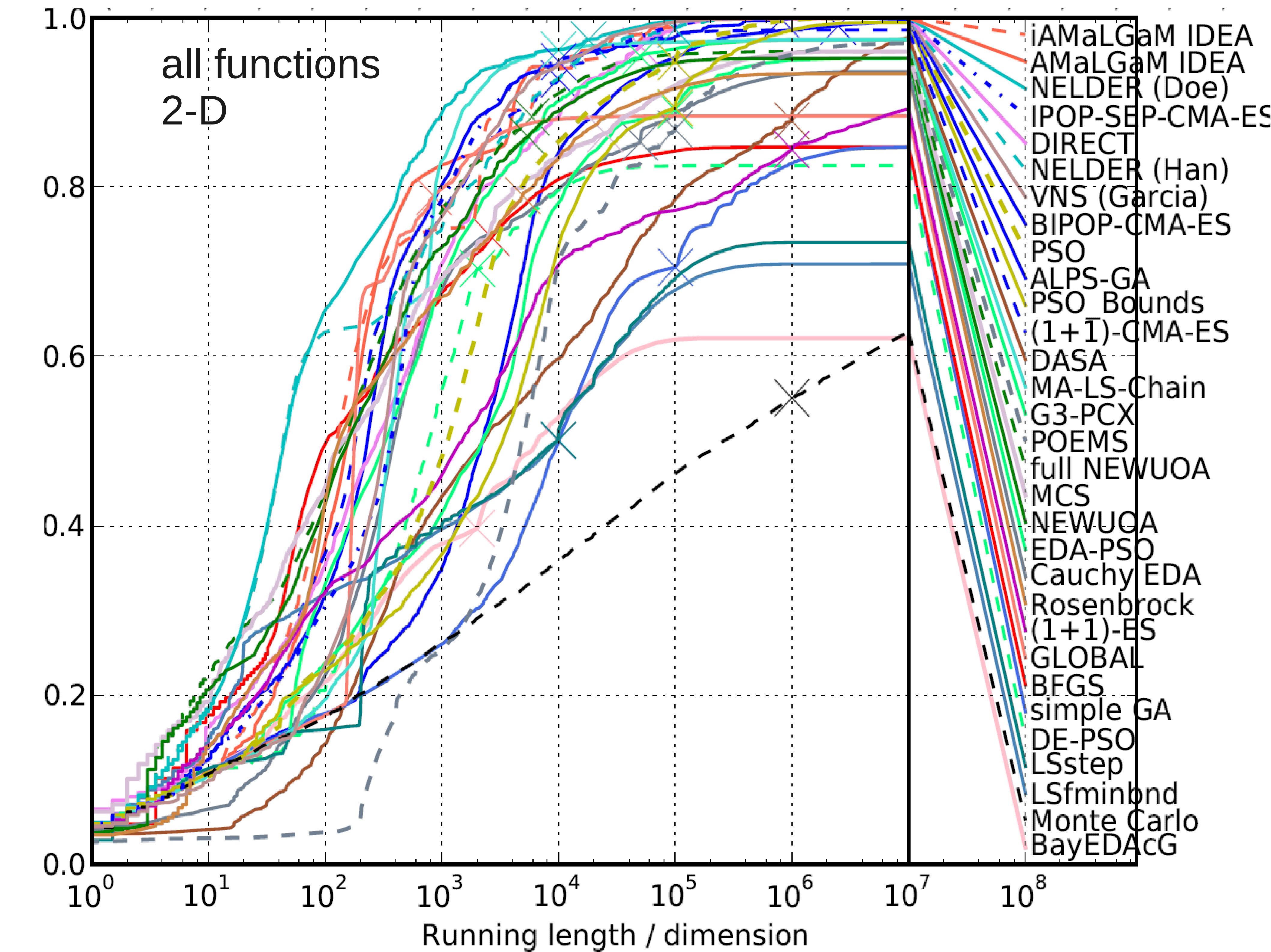


Results of 2010 (noisefree, 20-D)

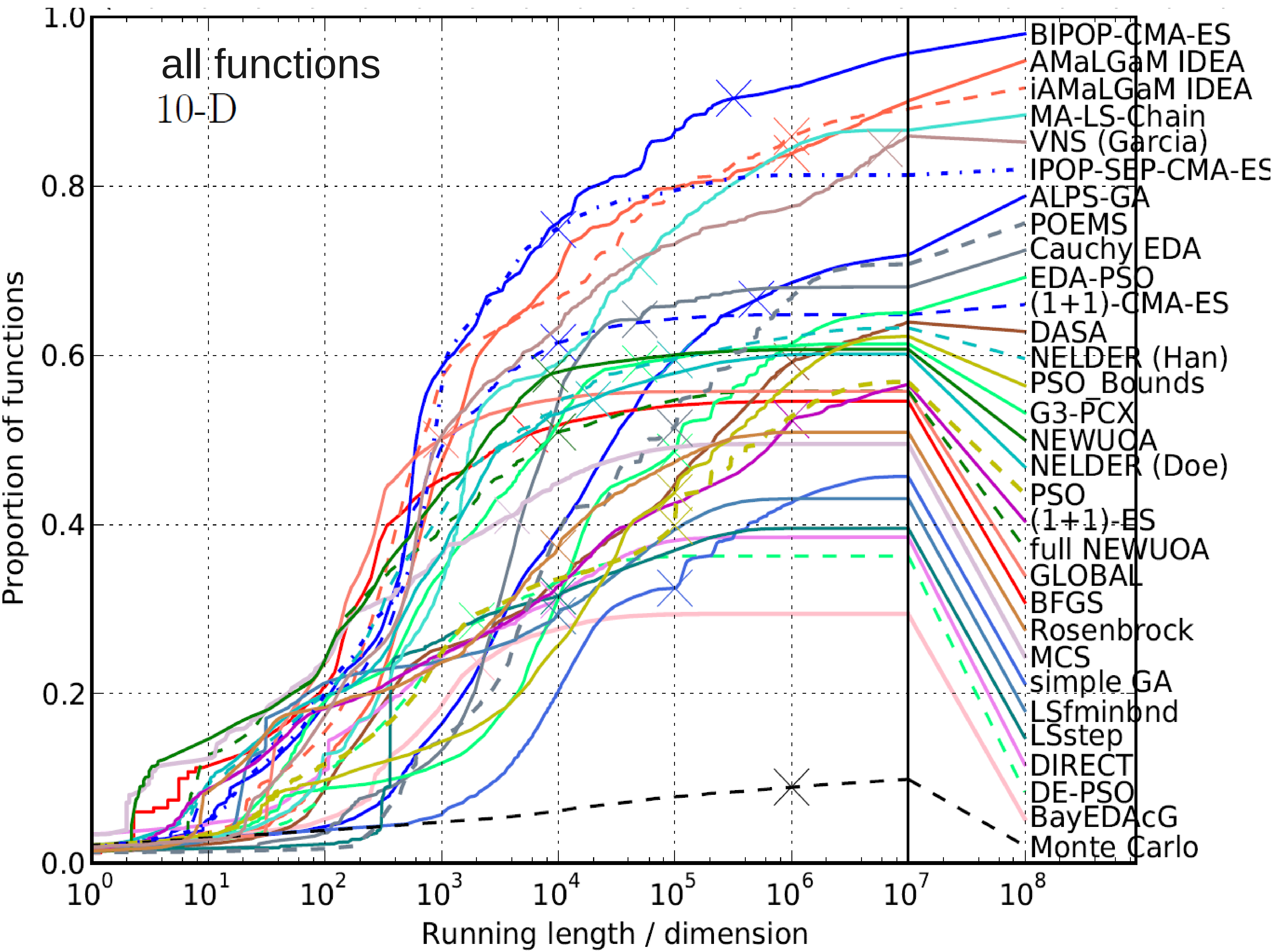


Results

- Functions are not that easy to solve: the best algorithms need 10000 D function evaluations to solve 75% of the problems (function-target pairs)
- Given at most 500 D evaluations: MCS, NEWUOA and GLOBAL do well
- Given more evaluations: variants of CMA-ES and AMaLGaM-IDEA do well
- In very low dimension Nelder-Mead is superior

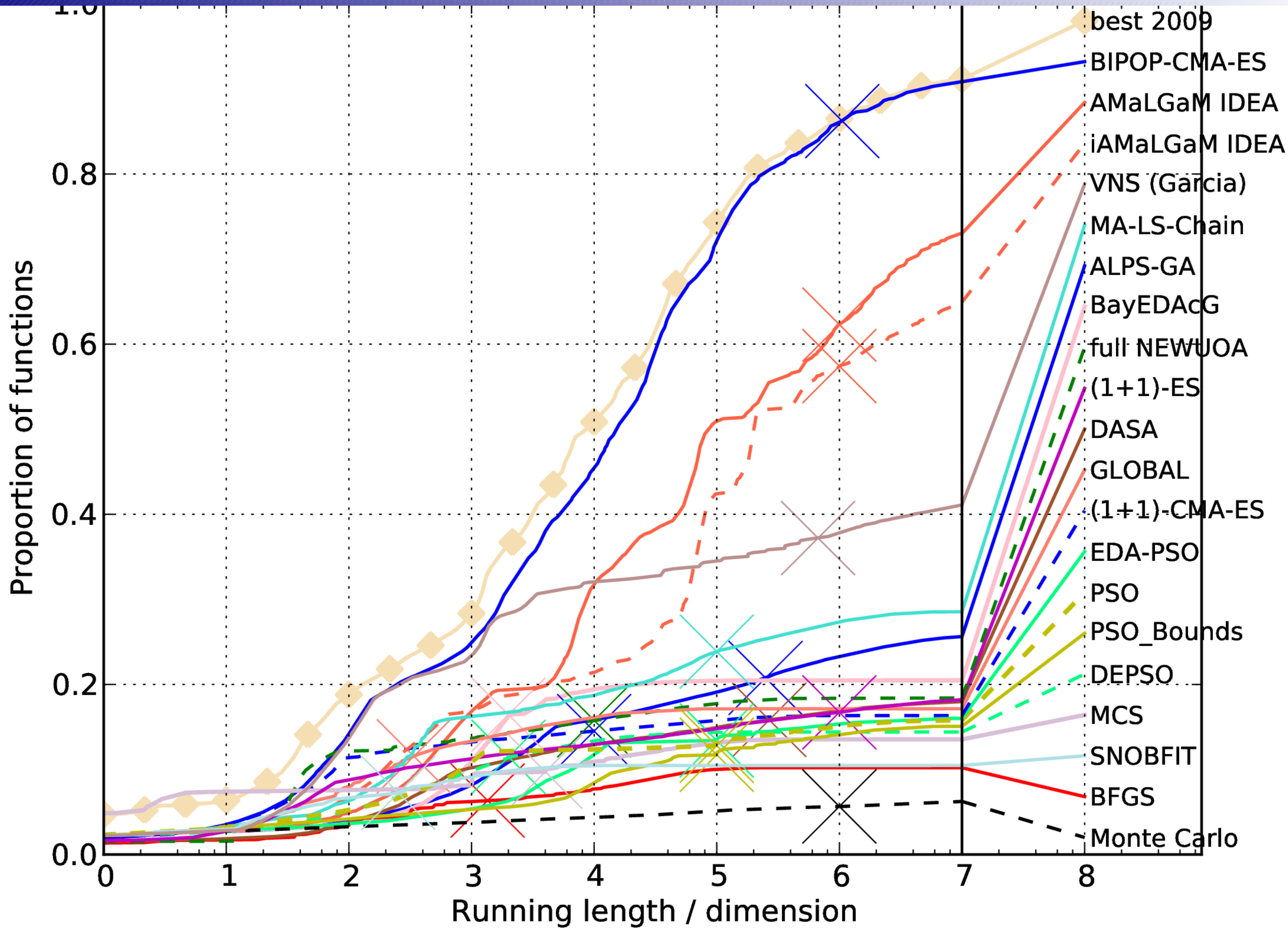


all functions
2-D
1.0
0.8
0.6
0.4
0.2
0.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
Running length / dimension
iAMaLGaM IDEA
AMaLGaM IDEA
NELDER (Doe)
IPOP-SEP-CMA-ES
DIRECT
NELDER (Han)
VNS (Garcia)
BIPOP-CMA-ES
PSO
ALPS-GA
PSO_Bounds
(1+1)-CMA-ES
DASA
MA-LS-Chain
G3-PCX
POEMS
full NEWUOA
MCS
NEWUOA
EDA-PSO
Cauchy EDA
Rosenbrock
(1+1)-ES
GLOBAL
BFGS
simple GA
DE-PSO
LSstep
LSfminbnd
Monte Carlo
BayEDAcG

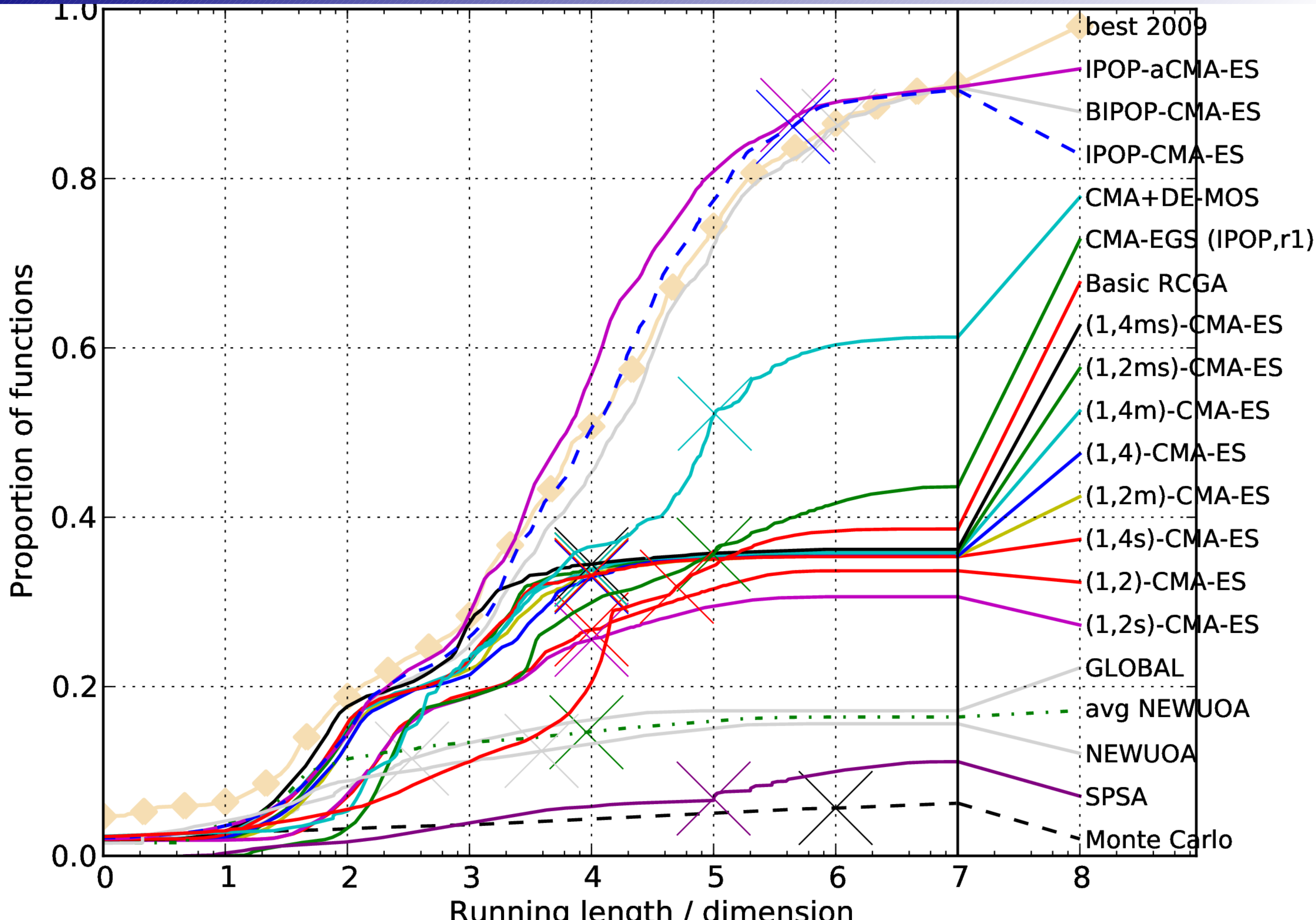


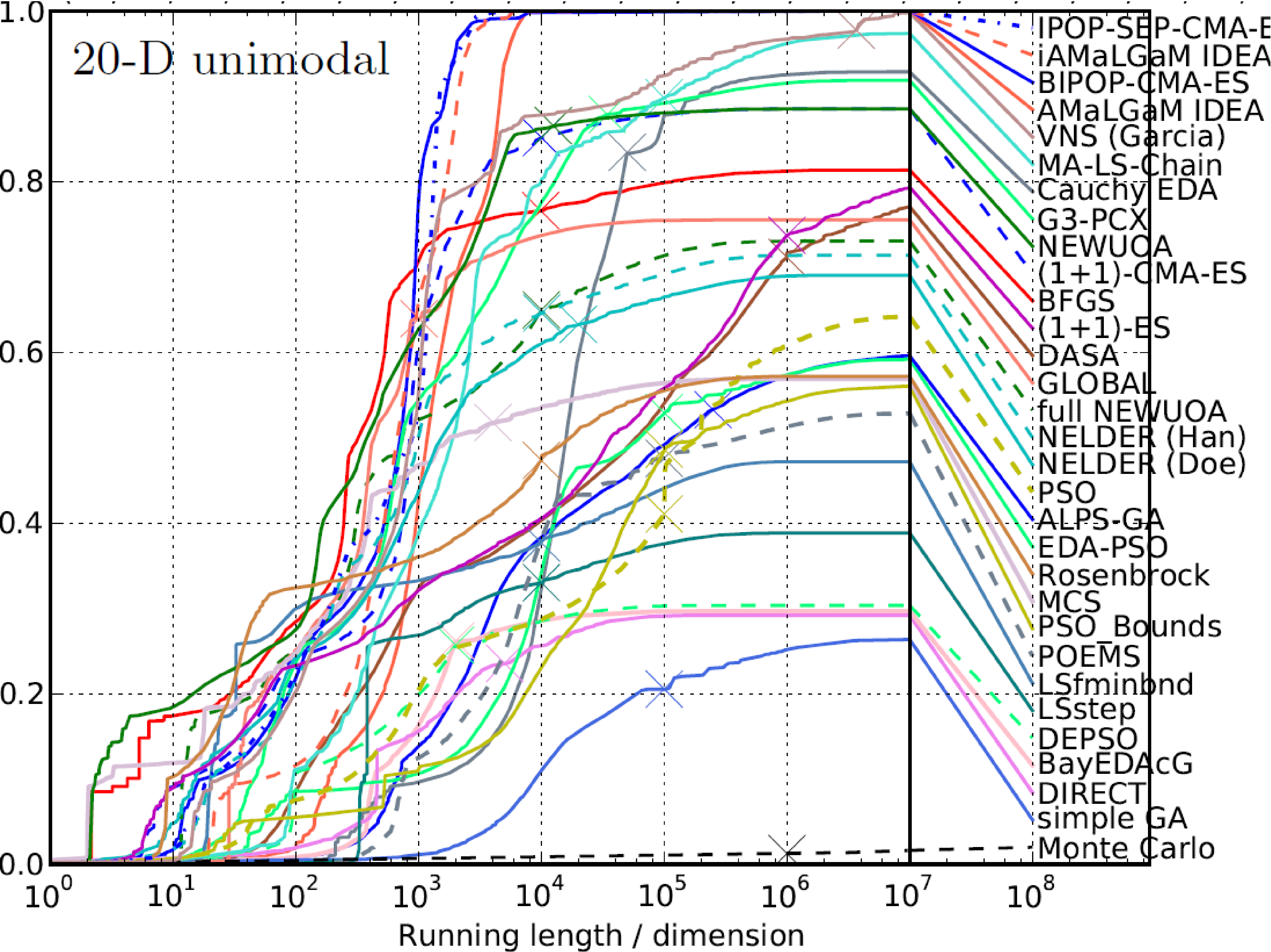
all functions
10-D
Proportion of functions
Running length / dimension
1.0
0.8
0.6
0.4
0.2
0.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
BIPOP-CMA-ES
AMaLGaM IDEA
iAMaLGaM IDEA
MA-LS-Chain
VNS (Garcia)
IPOP-SEP-CMA-ES
ALPS-GA
POEMS
Cauchy EDA
EDA-PSO
(1+1)-CMA-ES
DASA
NELDER (Han)
PSO_Bounds
G3-PCX
NEWUOA
NELDER (Doe)
PSO
(1+1)-ES
full NEWUOA
GLOBAL
BFGS
Rosenbrock
MCS
simple GA
LSfminbnd
LSstep
DIRECT
DE-PSO
BayEDAcG
Monte Carlo

Results of 2009 (noisy, f_{101}-f_{130}, 20-D)

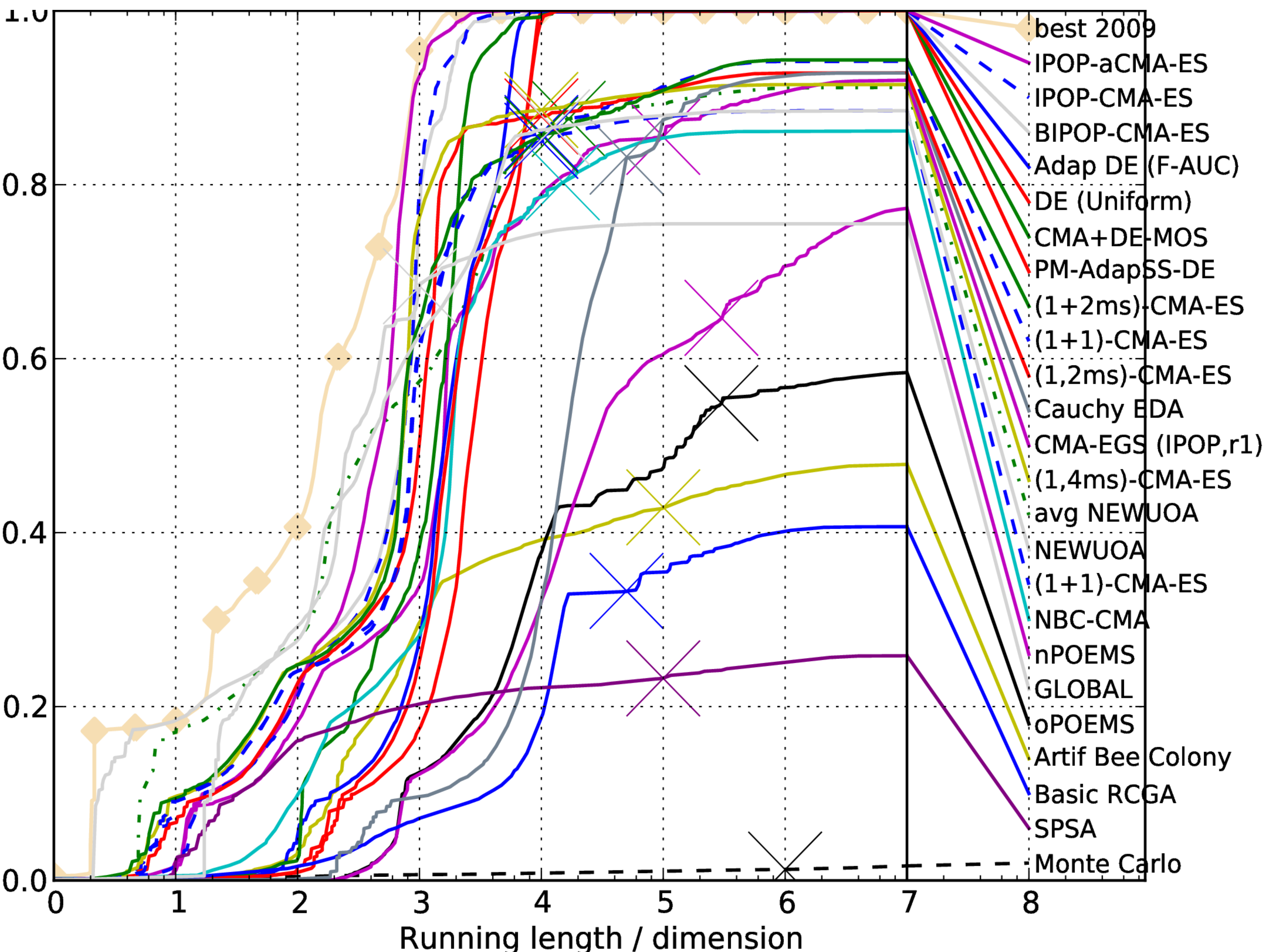


Results of 2010 (noisy, 20-D)

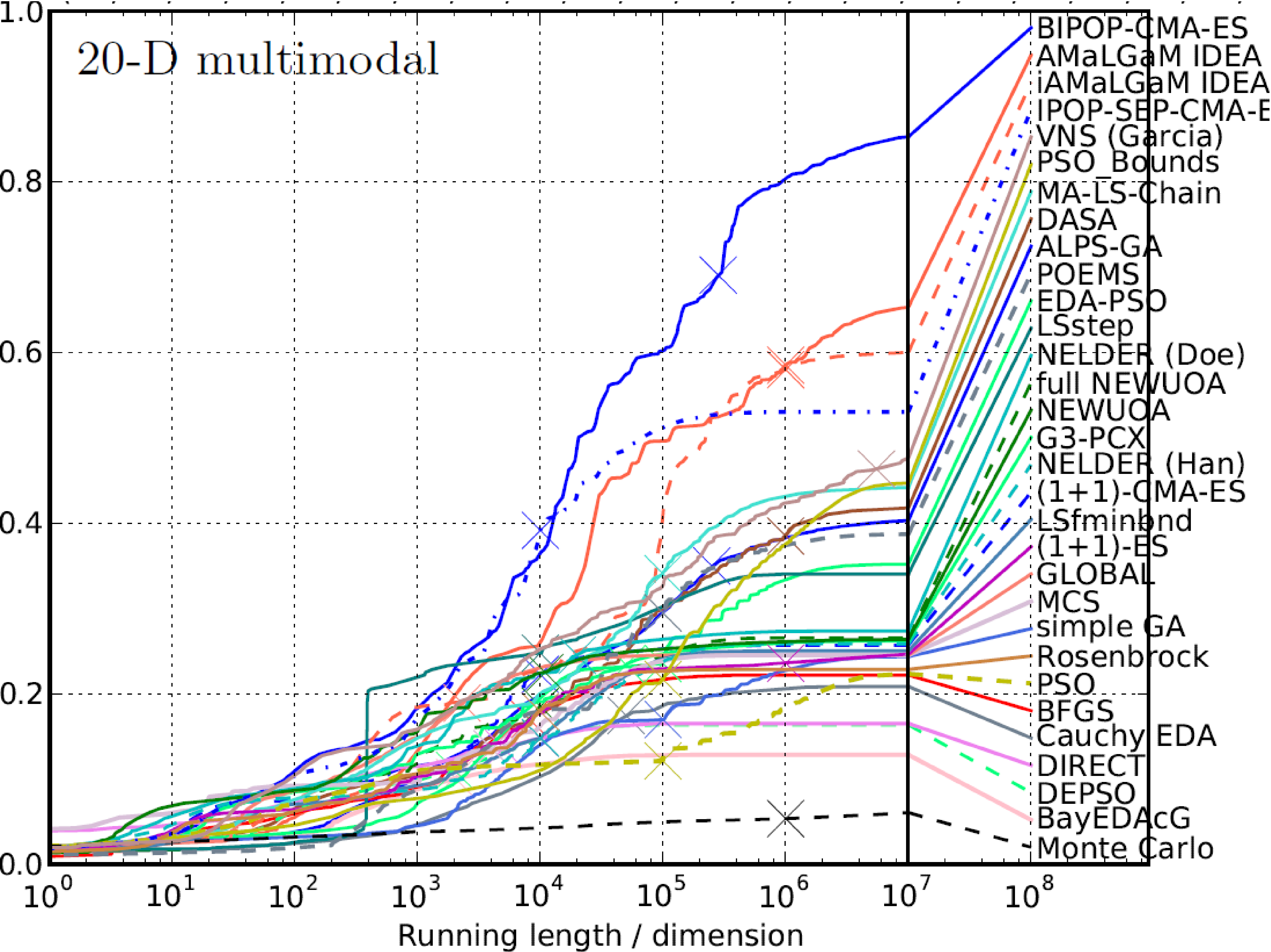


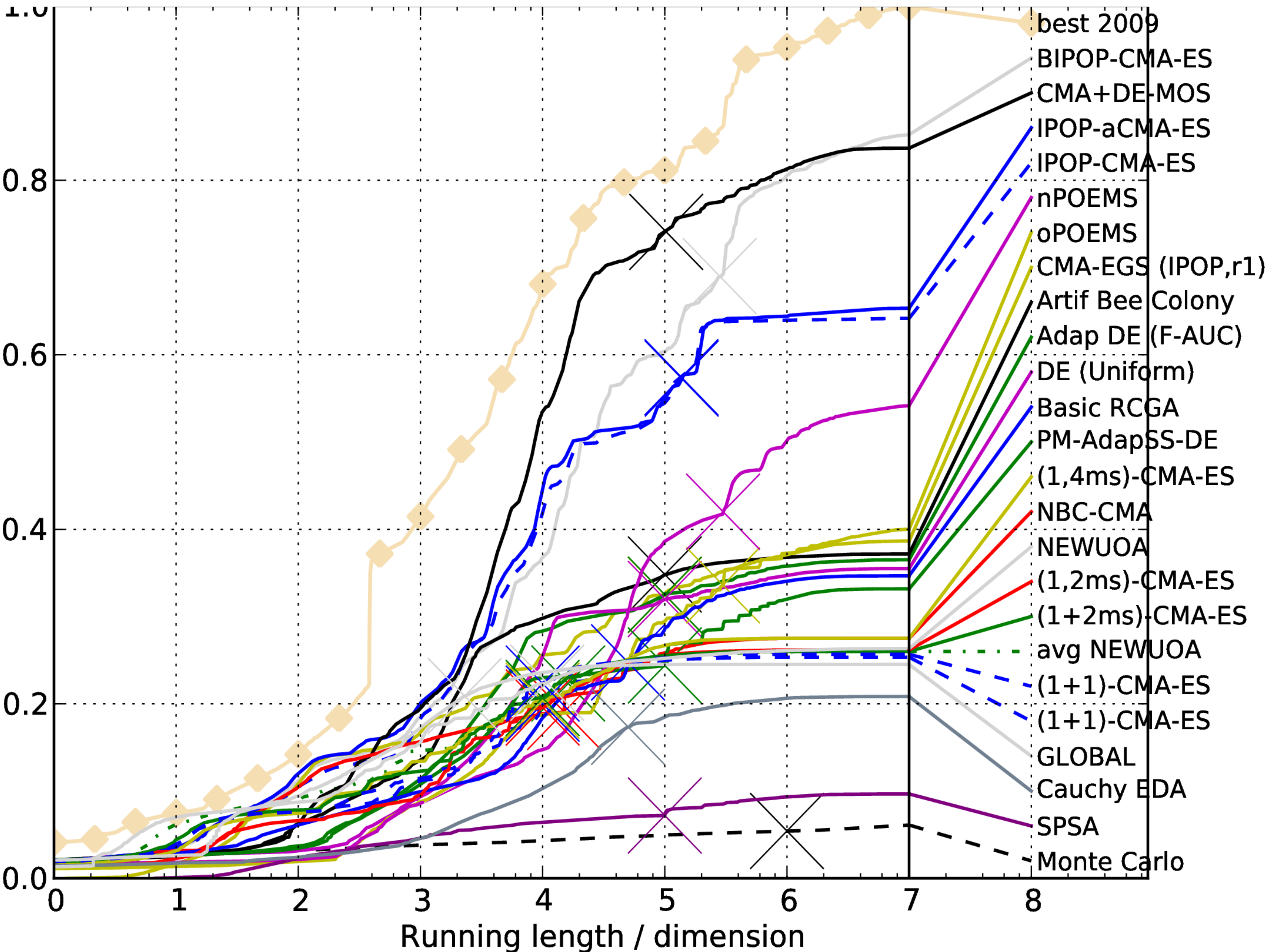


20-D unimodal
IPOP-SEP-CMA-E
iAMaLGaM IDEA
BIPOP-CMA-ES
AMaLGaM IDEA
VNS (Garcia)
MA-LS-Chain
Cauchy EDA
G3-PCX
NEWUOA
(1+1)-CMA-ES
BFGS
(1+1)-ES
DASA
GLOBAL
full NEWUOA
NELDER (Han)
NELDER (Doe)
PSO
ALPS-GA
EDA-PSO
Rosenbrock
MCS
PSO_Bounds
POEMS
LSfminbnd
LSstep
DEPSO
BayEDAcG
DIRECT
simple GA
Monte Carlo
Running length / dimension



1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
Running length / dimension
best 2009
IPOP-aCMA-ES
IPOP-CMA-ES
BIPOP-CMA-ES
Adap DE (F-AUC)
DE (Uniform)
CMA+DE-MOS
PM-AdapSS-DE
(1+2ms)-CMA-ES
(1+1)-CMA-ES
(1,2ms)-CMA-ES
Cauchy EDA
CMA-EGS (IPOP,r1)
(1,4ms)-CMA-ES
avg NEWUOA
NEWUOA
(1+1)-CMA-ES
NBC-CMA
nPOEMS
GLOBAL
oPOEMS
Artif Bee Colony
Basic RCGA
SPSA
Monte Carlo


20-D multimodal
1.0
0.8
0.6
0.4
0.2
0.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
Running length / dimension
BIPOP-CMA-ES
AMaLGaM IDEA
iAMaLGaM IDEA
IPOP-SEP-CMA-E
VNS (Garcia)
PSO_Bounds
MA-LS-Chain
DASA
ALPS-GA
POEMS
EDA-PSO
LSstep
NELDER (Doe)
full NEWUOA
NEWUOA
G3-PCX
NELDER (Han)
(1+1)-CMA-ES
LSfminbnd
(1+1)-ES
GLOBAL
MCS
simple GA
Rosenbrock
PSO
BFGS
Cauchy EDA
DIRECT
DEPSO
BayEDAcG
Monte Carlo


1.0
0.8
0.6
0.4
0.2
0.0
0
1
2
3
4
5
6
7
8
Running length / dimension
best 2009
BIPOP-CMA-ES
CMA+DE-MOS
IPOP-aCMA-ES
IPOP-CMA-ES
nPOEMS
oPOEMS
CMA-EGS (IPOP,r1)
Artif Bee Colony
Adap DE (F-AUC)
DE (Uniform)
Basic RCGA
PM-AdapSS-DE
(1,4ms)-CMA-ES
NBC-CMA
NEWUOA
(1,2ms)-CMA-ES
(1+2ms)-CMA-ES
avg NEWUOA
(1+1)-CMA-ES
(1+1)-CMA-ES
GLOBAL
Cauchy EDA
SPSA
Monte Carlo

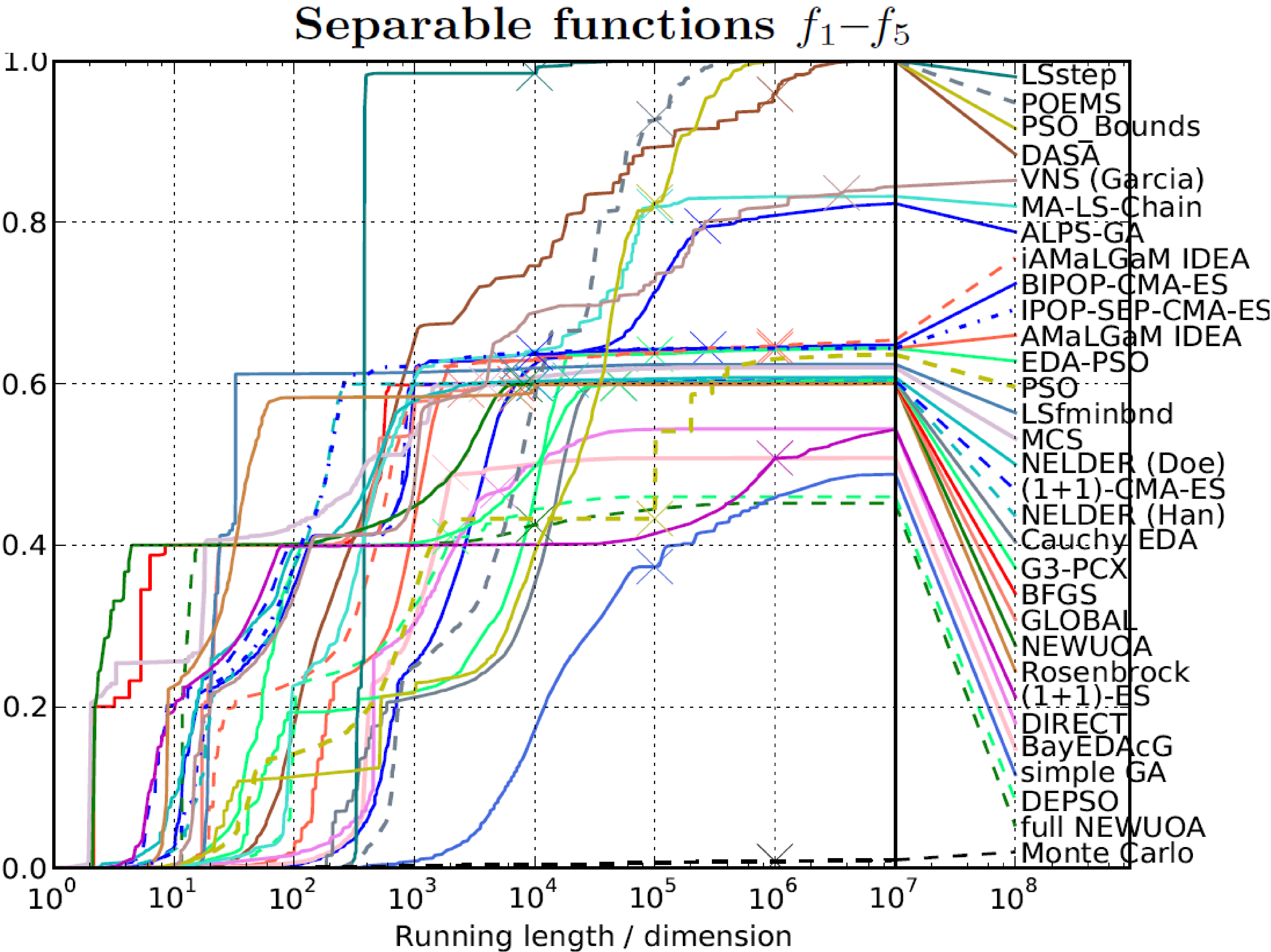
% SEPARABLE
1 Sphere
2 Ellipsoid separable with monotone x-transformation, condition 1e6
3 Rastrigin separable with both x-transformations "condition" 10
4 Skew Rastrigin-Bueche separable, "condition" 10, skew-"condition" 100
5 Linear slope, neutral extension outside the domain (not flat)

% LOW OR MODERATE CONDITION
6 Attractive sector function
7 Step-ellipsoid, condition 100
8 Rosenbrock, original
9 Rosenbrock, rotated

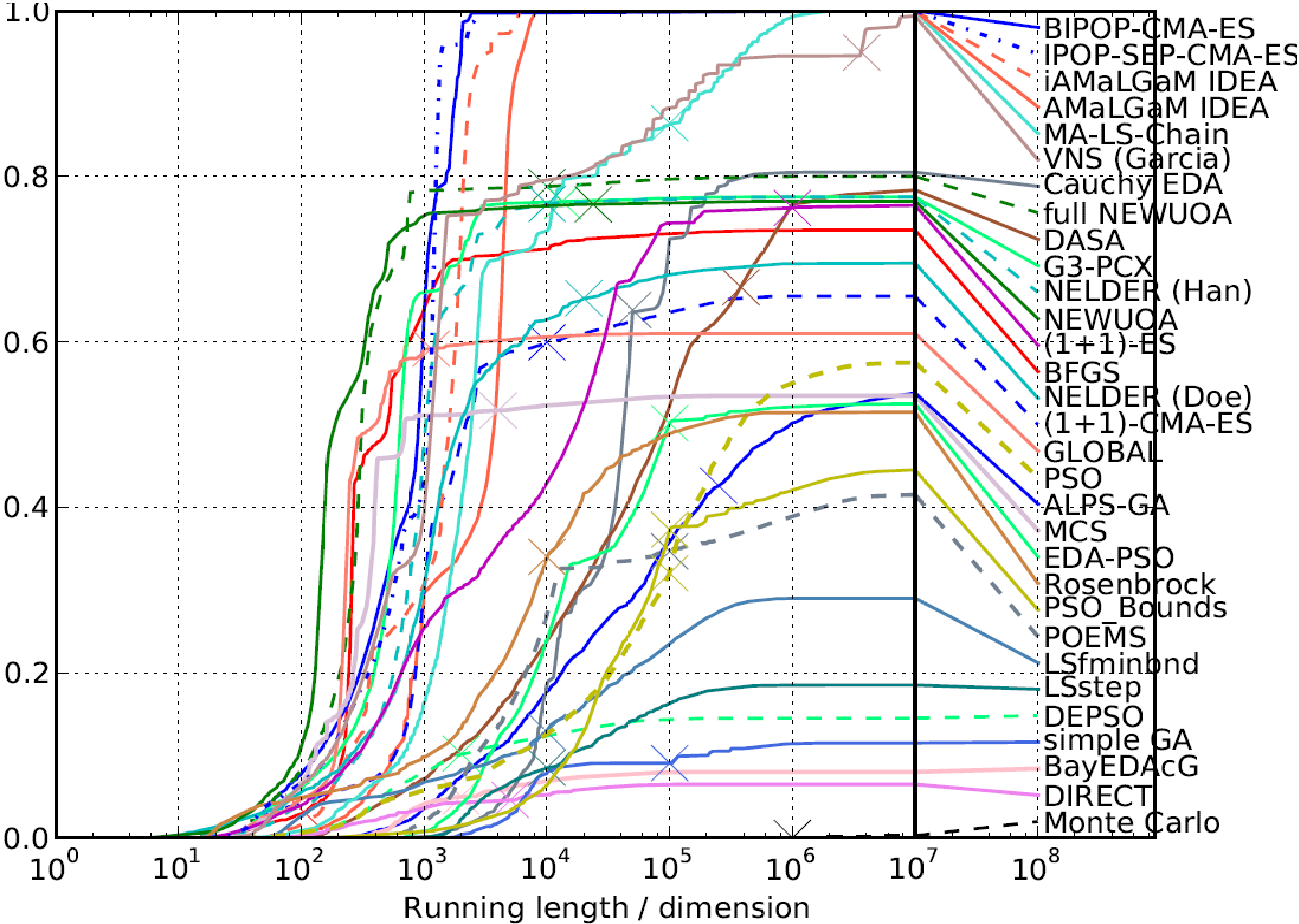
% HIGH CONDITION
10 Ellipsoid with monotone x-transformation, condition 1e6
11 Discus with monotone x-transformation, condition 1e6
12 Bent cigar with asymmetric x-transformation, condition 1e6
13 Sharp ridge, slope 1:100, condition 10
14 Sum of different powers

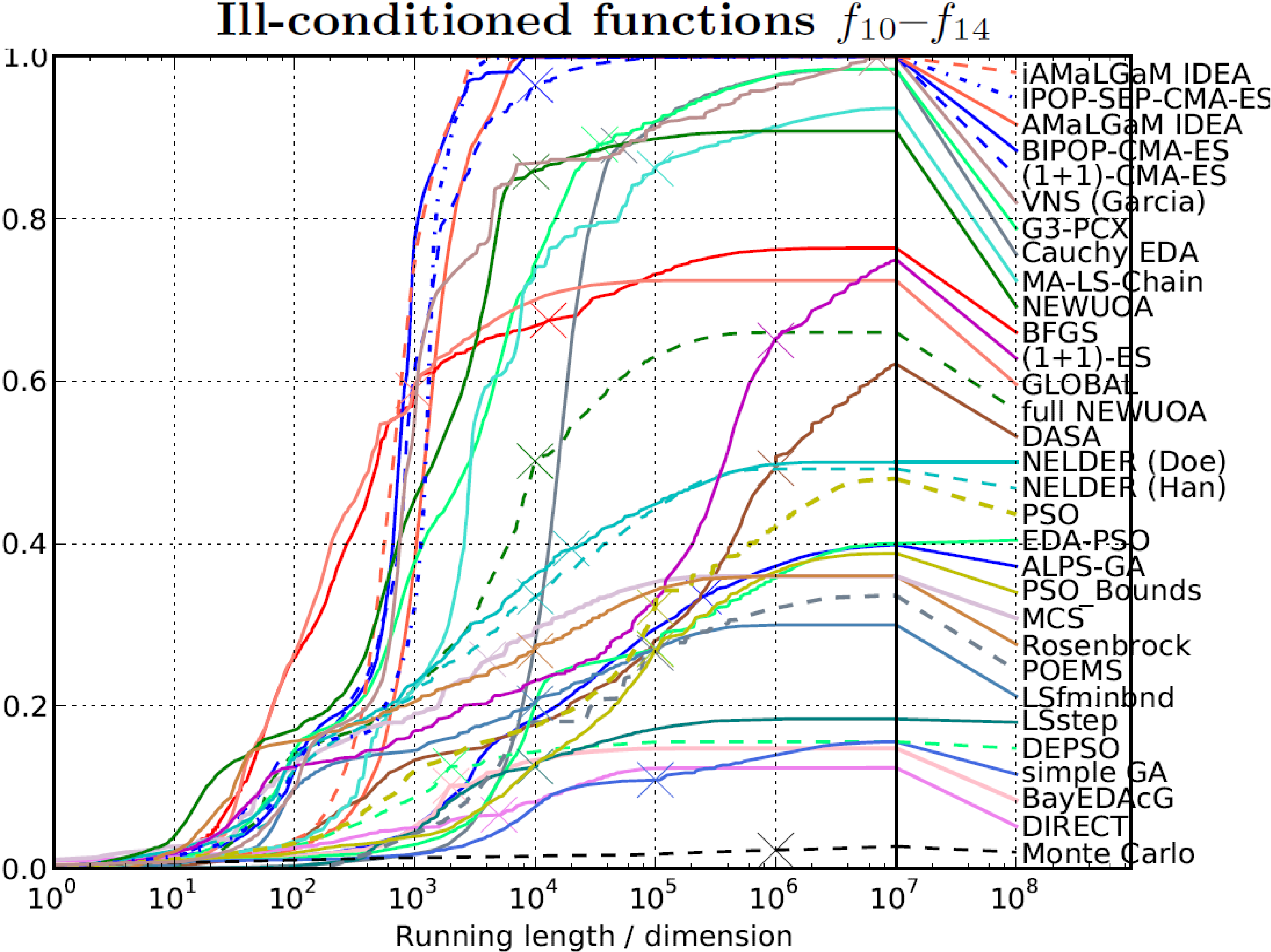
% MULTI-MODAL
15 Rastrigin with both x-transformations, condition 10
16 Weierstrass with monotone x-transformation, condition 100
17 Schaffer F7 with asymmetric x-transformation, condition 10
18 Schaffer F7 with asymmetric x-transformation, condition 1000
19 F8F2 composition of 2-D Griewank-Rosenbrock

% MULTI-MODAL WITH WEAK GLOBAL STRUCTURE
20 Schwefel x*sin(x) with tridiagonal transformation, condition 10
21 Gallagher 101 Gaussian peaks, condition up to 1000
22 Gallagher 21 Gaussian peaks, condition up to 1000, 1000 for global opt
23 Katsuuras repetitive rugged function
24 Lunacek bi-Rastrigin, condition 100

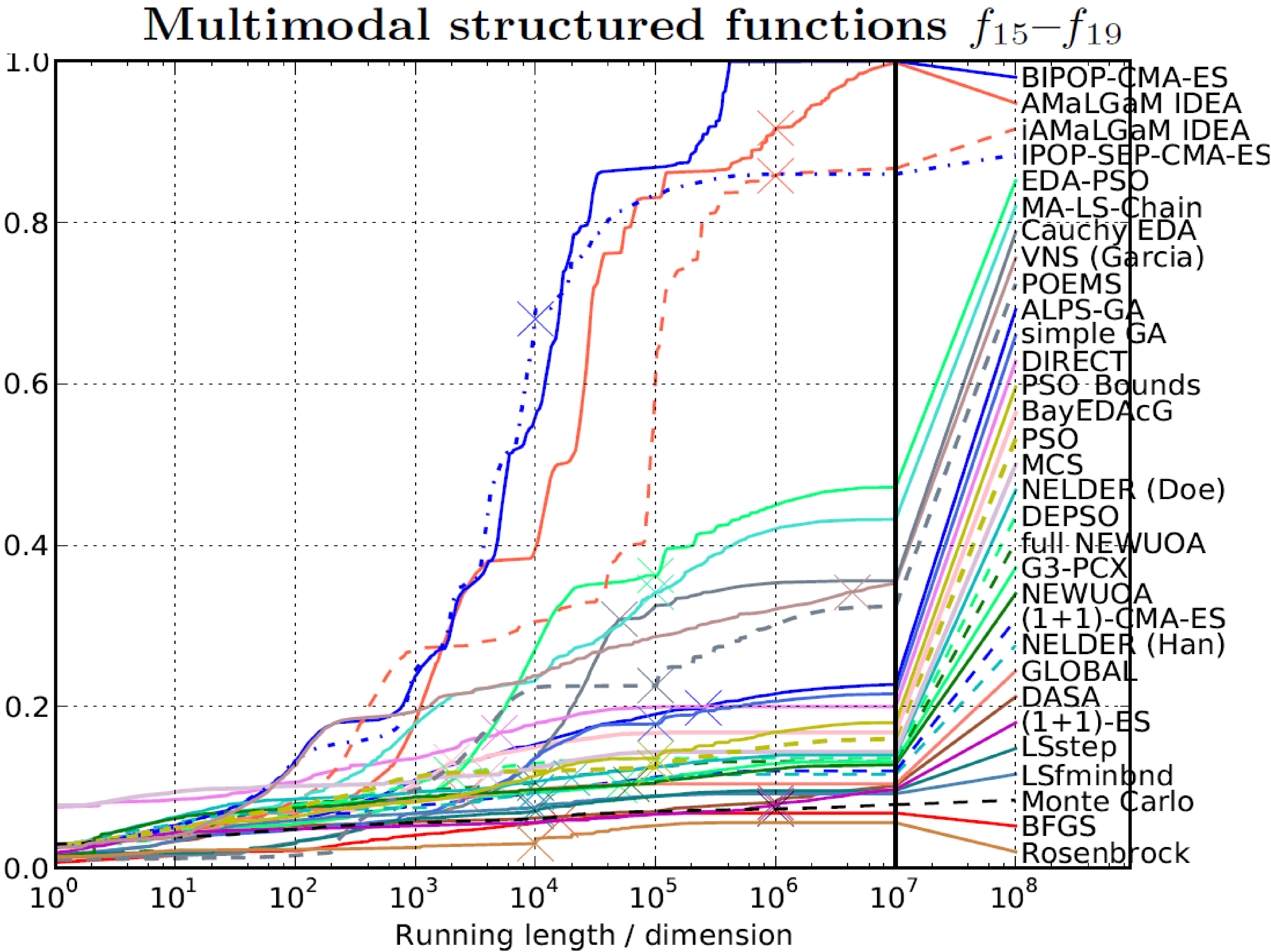

Separable functions f_1–f_5
1.0
0.8
0.6
0.4
0.2
0.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
Running length / dimension
LSstep
POEMS
PSO_Bounds
DASA
VNS (Garcia)
MA-LS-Chain
ALPS-GA
iAMaLGaM IDEA
BIPOP-CMA-ES
IPOP-SEP-CMA-ES
AMaLGaM IDEA
EDA-PSO
PSO
LSfminbnd
MCS
NELDER (Doe)
(1+1)-CMA-ES
NELDER (Han)
Cauchy EDA
G3-PCX
BFGS
GLOBAL
NEWUOA
Rosenbrock
(1+1)-ES
DIRECT
BayEDAcG
simple GA
DEPSO
full NEWUOA
Monte Carlo

Moderate functions f_6–f_9

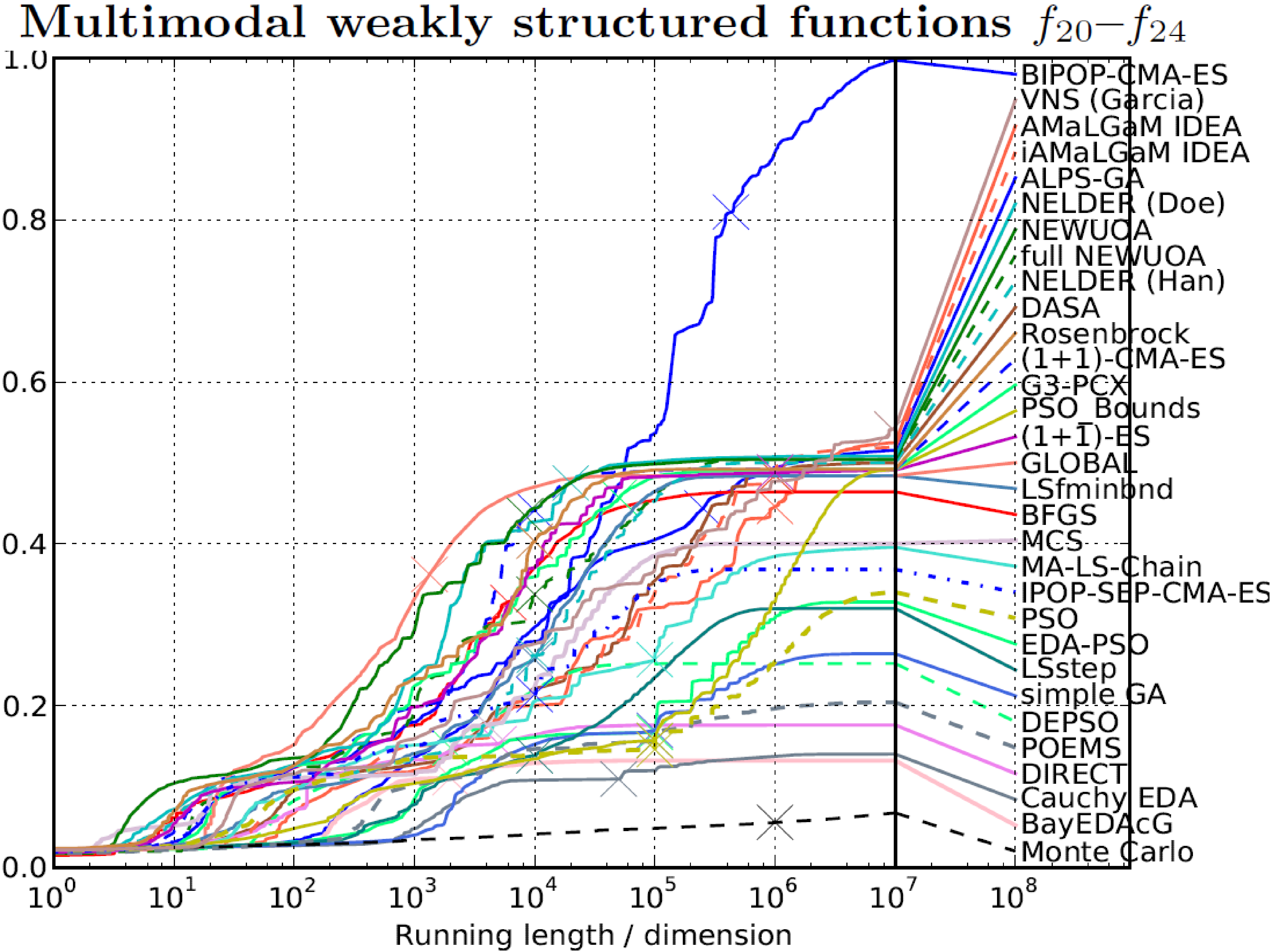




Ill-conditioned functions f_{10}–f_{14}
1.0
0.8
0.6
0.4
0.2
0.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
Running length / dimension
iAMaLGaM IDEA
IPOP-SEP-CMA-ES
AMaLGaM IDEA
BIPOP-CMA-ES
(1+1)-CMA-ES
VNS (Garcia)
G3-PCX
Cauchy EDA
MA-LS-Chain
NEWUOA
BFGS
(1+1)-ES
GLOBAL
full NEWUOA
DASA
NELDER (Doe)
NELDER (Han)
PSO
EDA-PSO
ALPS-GA
PSO_Bounds
MCS
Rosenbrock
POEMS
LSfminbnd
LSstep
DEPSO
simple GA
BayEDAcG
DIRECT
Monte Carlo

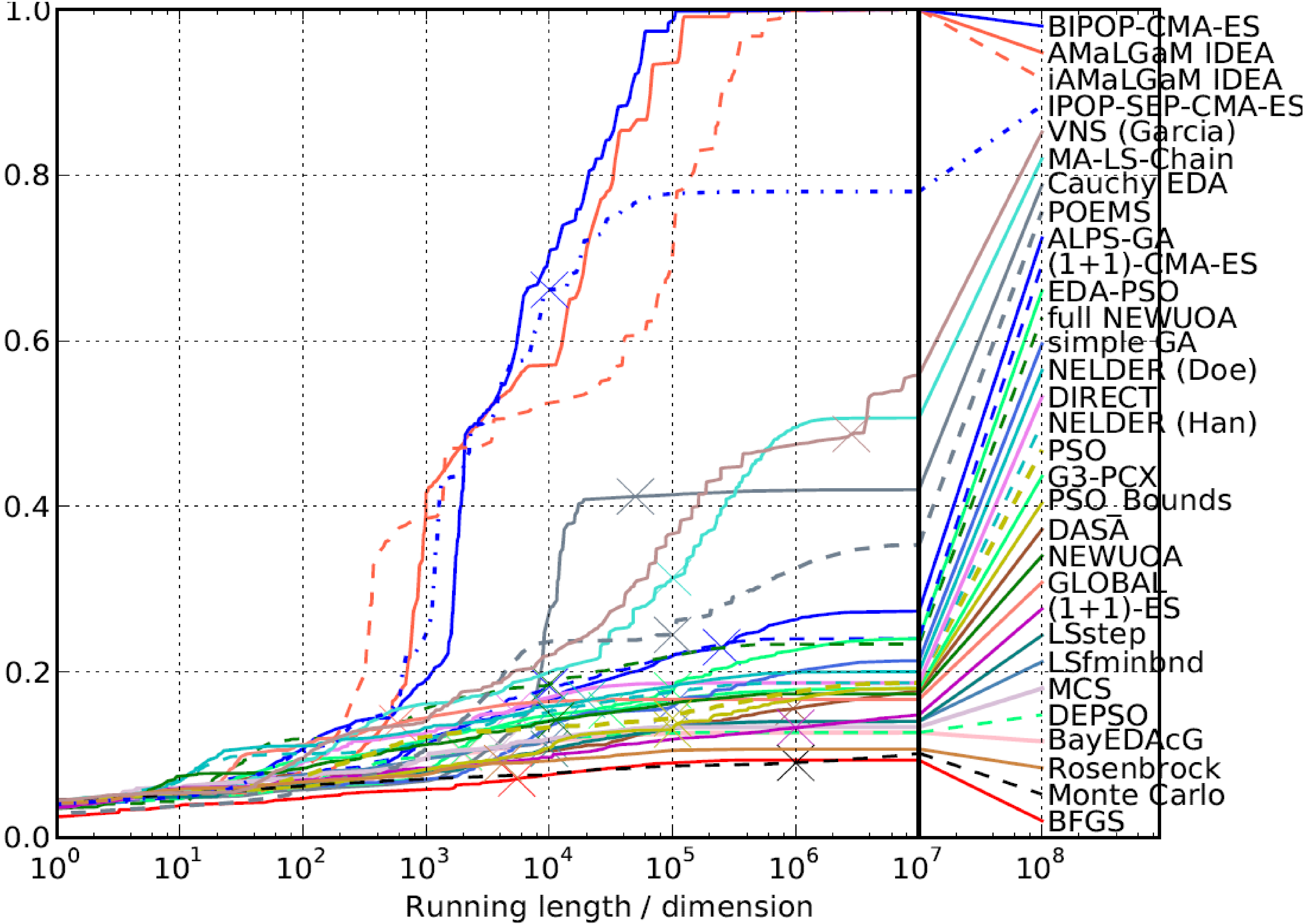


Multimodal structured functions f_{15}–f_{19}
1.0
0.8
0.6
0.4
0.2
0.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
Running length / dimension
BIPOP-CMA-ES
AMaLGaM IDEA
iAMaLGaM IDEA
IPOP-SEP-CMA-ES
EDA-PSO
MA-LS-Chain
Cauchy EDA
VNS (Garcia)
POEMS
ALPS-GA
simple GA
DIRECT
PSO_Bounds
BayEDAcG
PSO
MCS
NELDER (Doe)
DEPSO
full NEWUOA
G3-PCX
NEWUOA
(1+1)-CMA-ES
NELDER (Han)
GLOBAL
DASA
(1+1)-ES
LSstep
LSfminbnd
Monte Carlo
BFGS
Rosenbrock



Multimodal weakly structured functions f_{20}–f_{24}
1.0
0.8
0.6
0.4
0.2
0.0
10^0
10^1
10^2
10^3
10^4
10^5
10^6
10^7
10^8
Running length / dimension
BIPOP-CMA-ES
VNS (Garcia)
AMaLGaM IDEA
iAMaLGaM IDEA
ALPS-GA
NELDER (Doe)
NEWUOA
full NEWUOA
NELDER (Han)
DASA
Rosenbrock
(1+1)-CMA-ES
G3-PCX
PSO_Bounds
(1+1)-ES
GLOBAL
LSfminbnd
BFGS
MCS
MA-LS-Chain
IPOP-SEP-CMA-ES
PSO
EDA-PSO
LSstep
simple GA
DEPSO
POEMS
DIRECT
Cauchy EDA
BayEDAcG
Monte Carlo

Non-smooth functions f_7, f_{16}, f_{23}



Single Function Table

Table 6: 20-D, running time excess ERT/$\mathrm{ERT}_{\mathrm{best}}$ on f_6, in italics is given the median final function value and the median number of function evaluations to reach this value divided by dimension

6 Attractive sector

Δftarget	1e+03	1e+02	1e+01	1e+00	1e-01	1e-02	1e-03	1e-04	1e-05	1e-07	Δftarget
$\mathrm{ERT}_{\mathrm{best}}$/D	4.03	26	64.7	87.2	123	152	184	219	248	309	$\mathrm{ERT}_{\mathrm{best}}$/D
ALPS	59	25	34	54	64	78	100	150	370	*14e-7/2e5*	ALPS [17]
AMaLGaM IDEA	26	22	19	22	21	22	22	21	22	22	AMaLGaM IDEA [4]
avg NEWUOA	**2.3**	**1.1**	1	**1**	1	**1**	**1**	**1**	**1**	1	avg NEWUOA [31]
BayEDAcG	46	41	*60e+0/2e3*	.	.	.	.	.	.	.	BayEDAcG [10]
BFGS	**2.2**	**2.7**	3.6	4.7	4.7	4.9	5	4.8	4.9	61	BFGS [30]
Cauchy EDA	6200	1500	1e3	1700	*17e-1/5e4*	.	.	.	.	.	Cauchy EDA [24]
BIPOP-CMA-ES	**2.9**	**2.2**	**1.5**	**1.7**	**1.6**	**1.6**	**1.6**	**1.5**	**1.6**	**1.6**	BIPOP-CMA-ES [15]
(1+1)-CMA-ES	**1.9**	4.5	13	180	1200	*13e-1/1e4*	.	.	.	.	(1+1)-CMA-ES [2]
DASA	12	6.8	9.9	19	25	33	49	58	63	74	DASA [19]
DEPSO	11	7.5	12	64	*13e-1/2e3*	.	.	.	.	.	DEPSO [12]
DIRECT	18	31	*40e+0/5e3*	.	.	.	.	.	.	.	DIRECT [25]
EDA-PSO	27	46	40	45	44	44	44	44	44	44	EDA-PSO [6]
full NEWUOA	5	**1.9**	**1.5**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	**1.4**	full NEWUOA [31]
G3-PCX	4.1	**1.4**	**1.4**	**2**	**2.1**	**2.1**	**2.2**	**2.2**	**2.3**	**2.4**	G3-PCX [26]
simple GA	320	130	2e3	*11e+0/1e5*	.	.	.	.	.	.	simple GA [22]
GLOBAL	5	**2.9**	3.6	4.9	8.5	*42e-3/2e3*	.	.	.	.	GLOBAL [23]
iAMaLGaM IDEA	5.1	5.6	5.4	6.8	7.1	7.7	7.8	7.7	8	8.3	iAMaLGaM IDEA [4]
LSfminbnd	9	31	160	760	1100	960	*72e-1/1e4*	.	.	.	LSfminbnd [28]
LSstep	140	260	2300	*59e+0/1e4*	.	.	.	.	.	.	LSstep [28]
MA-LS-Chain	11	4.9	7.5	8.9	8	7.7	7.2	6.7	6.5	6	MA-LS-Chain [21]
MCS (Neum)	**1.8**	33	*42e+0/4e3*	.	.	.	.	.	.	.	MCS (Neum) [18]
NELDER (Han)	**2.2**	**2.4**	**2.7**	3.3	3.2	3.5	3.5	3.5	4	7.4	NELDER (Han) [16]
NELDER (Doe)	**1.5**	**2.3**	9.1	20	28	65	110	430	*46e-5/2e4*	.	NELDER (Doe) [5]
NEWUOA	**1**	**1**	1	**1.3**	**1.4**	**1.5**	**1.6**	**1.6**	**1.7**	**1.7**	NEWUOA [31]
(1+1)-ES	**2**	**2.2**	**2.1**	**2.8**	3.9	5.2	6.1	6.5	6.4	6.7	(1+1)-ES [1]
POEMS	89	26	31	37	36	36	36	35	36	37	POEMS [20]
PSO	6.4	280	1100	1400	980	820	710	620	570	790	PSO [7]
PSO_Bounds	9.5	45	120	150	140	140	140	130	160	220	PSO_Bounds [8]
Monte Carlo	2.4e5	*48e+1/1e6*	.	.	.	.	.	.	.	.	Monte Carlo [3]
Rosenbrock	**2.1**	3.9	31	76	210	230	810	*21e-2/1e4*	.	.	Rosenbrock [27]
IPOP-SEP-CMA-ES	3.2	**2.1**	**1.7**	**1.9**	**1.9**	**1.9**	**1.9**	**1.9**	**2**	**2**	IPOP-SEP-CMA-ES [29]
VNS (Garcia)	5	**2.8**	**1.9**	**1.9**	**1.7**	**1.7**	**1.7**	**1.6**	**1.6**	**1.6**	VNS (Garcia) [11]

Overview of best algorithms (20-D)

Functions	short runtime	long runtime
separable	NEWUOA (BFGS), LS-fminbnd	LS-step
moderate	NEWUOA (BFGS, GLOBAL)	IPOP-aCMA-ES
ill-conditioned	(NEWUOA) BFGS, GLOBAL	IPOP-aCMA-ES
non-smooth (2009)	IDEA (CMA-ES)	CMA-ES, IDEA
multimodal	(MCS, DIRECT, CMA-ES, IDEA)	IPOP-CMA-ES (ID
weak structure	(NEWUOA) GLOBAL	(BIPOP-CMA-ES)
noisy	(MCS, CMA-ES)	IPOP-aCMA-ES

(more) questions?

Any intelligent fool can make things bigger, more complex, and more violent. It takes a touch of genius, and a lot of courage, to move in the opposite direction.

Albert Einstein