Benchmarking Powell's Legacy: Performance of Five Derivative-Free Solvers in pdfo on the bbob Test Suite with and without Outliers

Dimo Brockhoff and Tanguy Villain

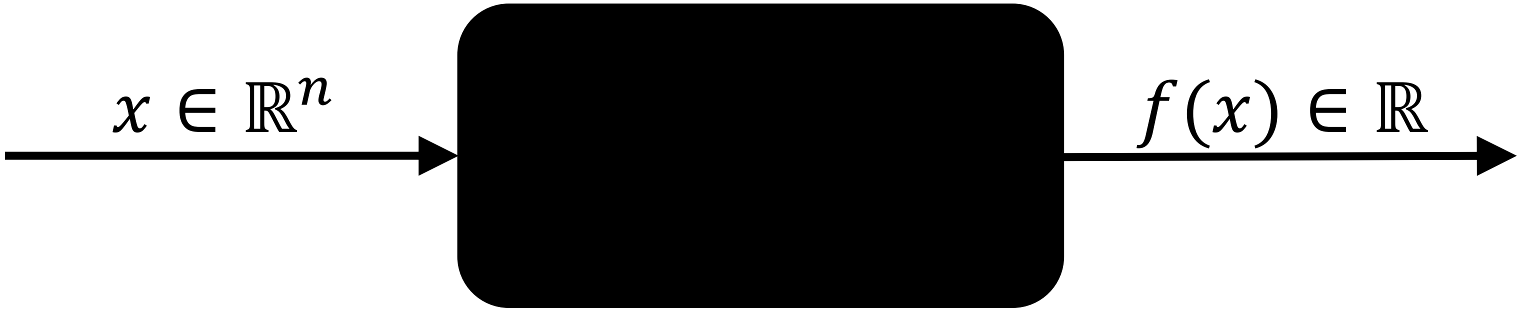
RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

Inria

INSTITUT POLYTECHNIQUE DE PARIS

Practical Blackbox Optimization

Given:



Not clear:

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

This is why we benchmark all kinds of solvers at the workshop

Our Goal

Investigate Derivative-Free Solvers

- and their performance on bbob suite
- with and without noise

Which algorithms?

- Michael J. D. Powell's: LINCOA, COBYLA, UOBYQA, BOBYQA, NEWUOA
- Why those?
  - Powell cared for practical algorithms
  - They are all available in python (partly in scipy.optimize) and easily usable
  - Here: implementation in PDFO by Tom M. Ragonneau and Zaikun Zhang based on original Fortran code of Powell (together with PRIMA, the "official" implementations of the algorithms)

The Algorithms of Michael Powell in a Nutshell

COBYLA (1994)

- trust-region algorithm, designed for "Constrained Optimization BY Linear Approximations"
- without constraints, the algo optimizes a linear model of the objective function within its trust region
- in COBYLA, the trust region radius is only decreased and never increased

UOBYQA (2002)

- "Unconstrained Optimization BY Quadratic Approximation"
- typical trust-region algorithm (trust region radius can decrease *and* increase)
- has lower bound on the radius (only decreased)

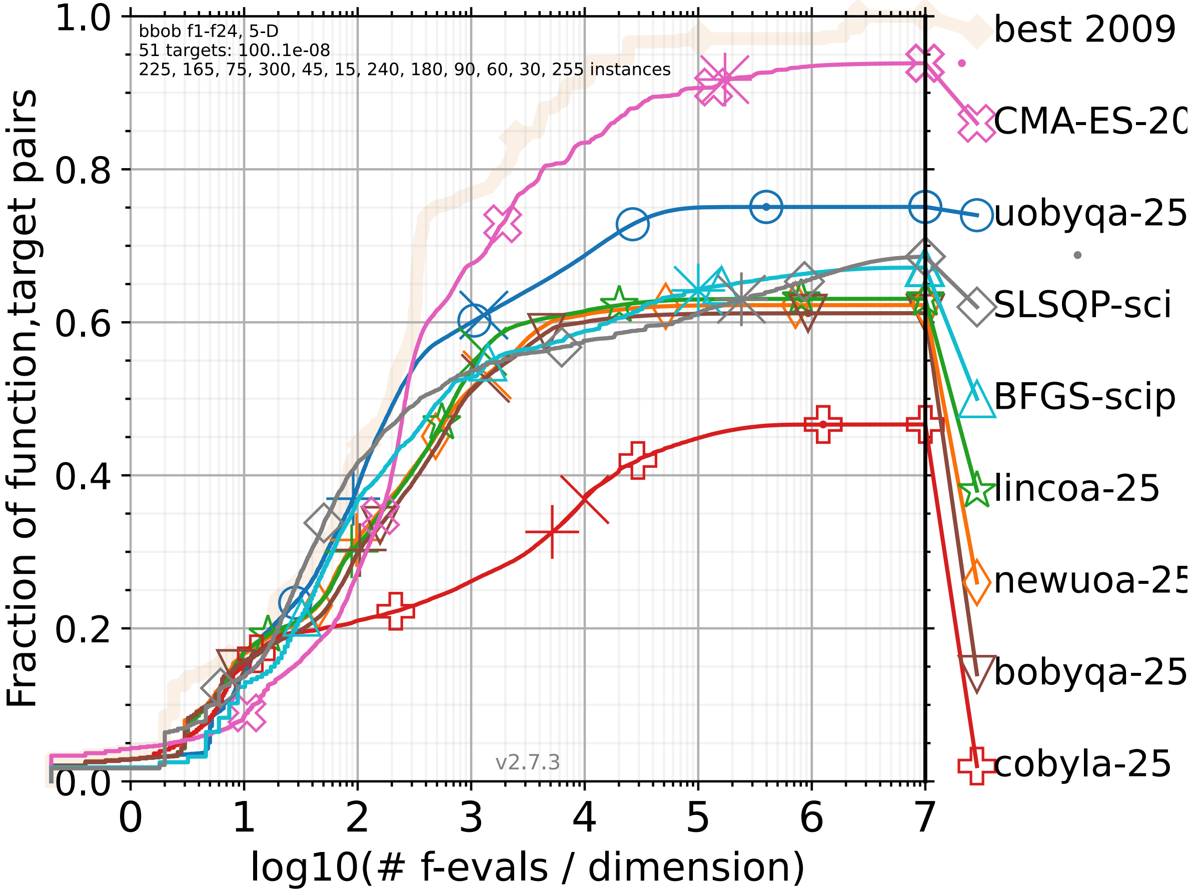
NEWUOA (2006), LINCOA (unpublished), and BOBYQA (2009)

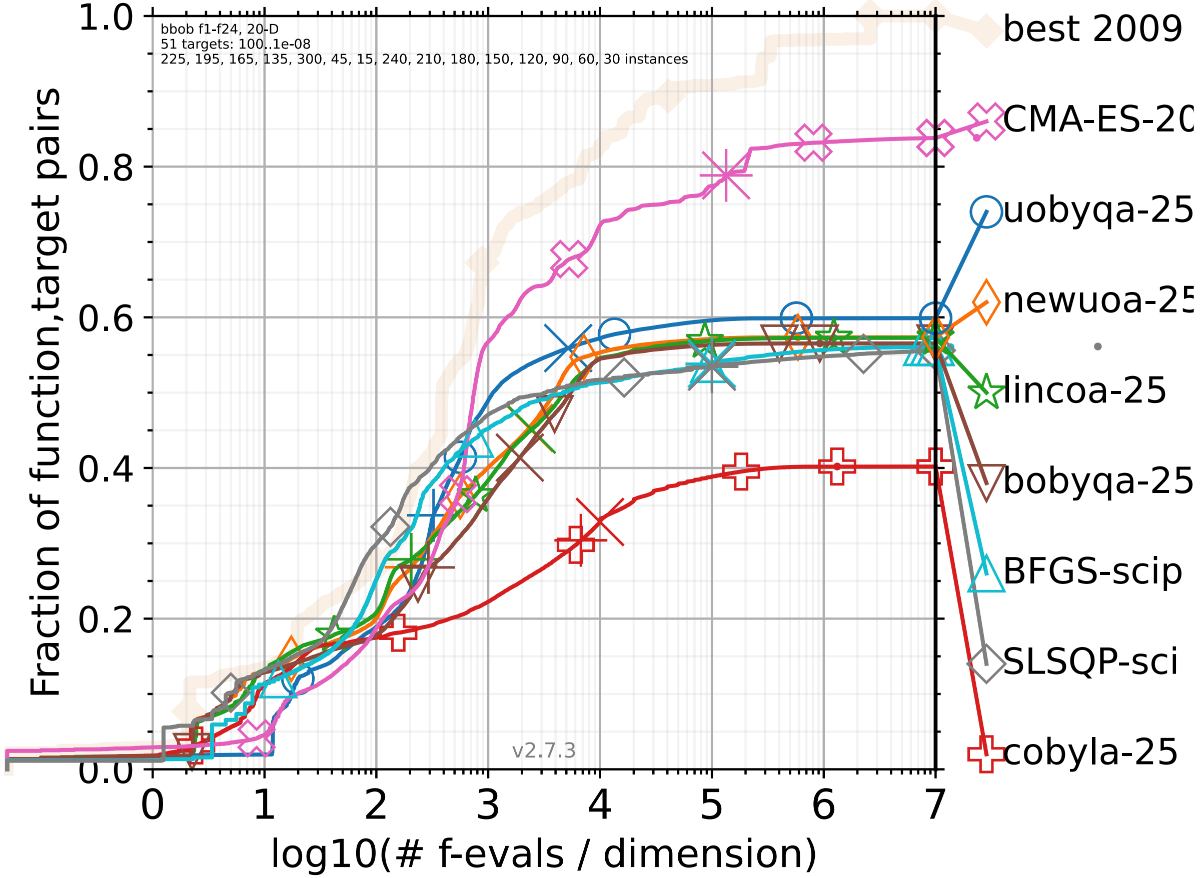
- the latest of Powell's algorithms—tailored towards unconstrained, linearly constrained and bound-constrained derivative-free problems
- trust-region-based model-building solvers that differ slightly how quadratic model's optimum is exploited
- the algos exploit (or don't in the case of NEWUOA) the availability of linear or bound constraints

Experiments on bbob (BBOB-Paper #1)

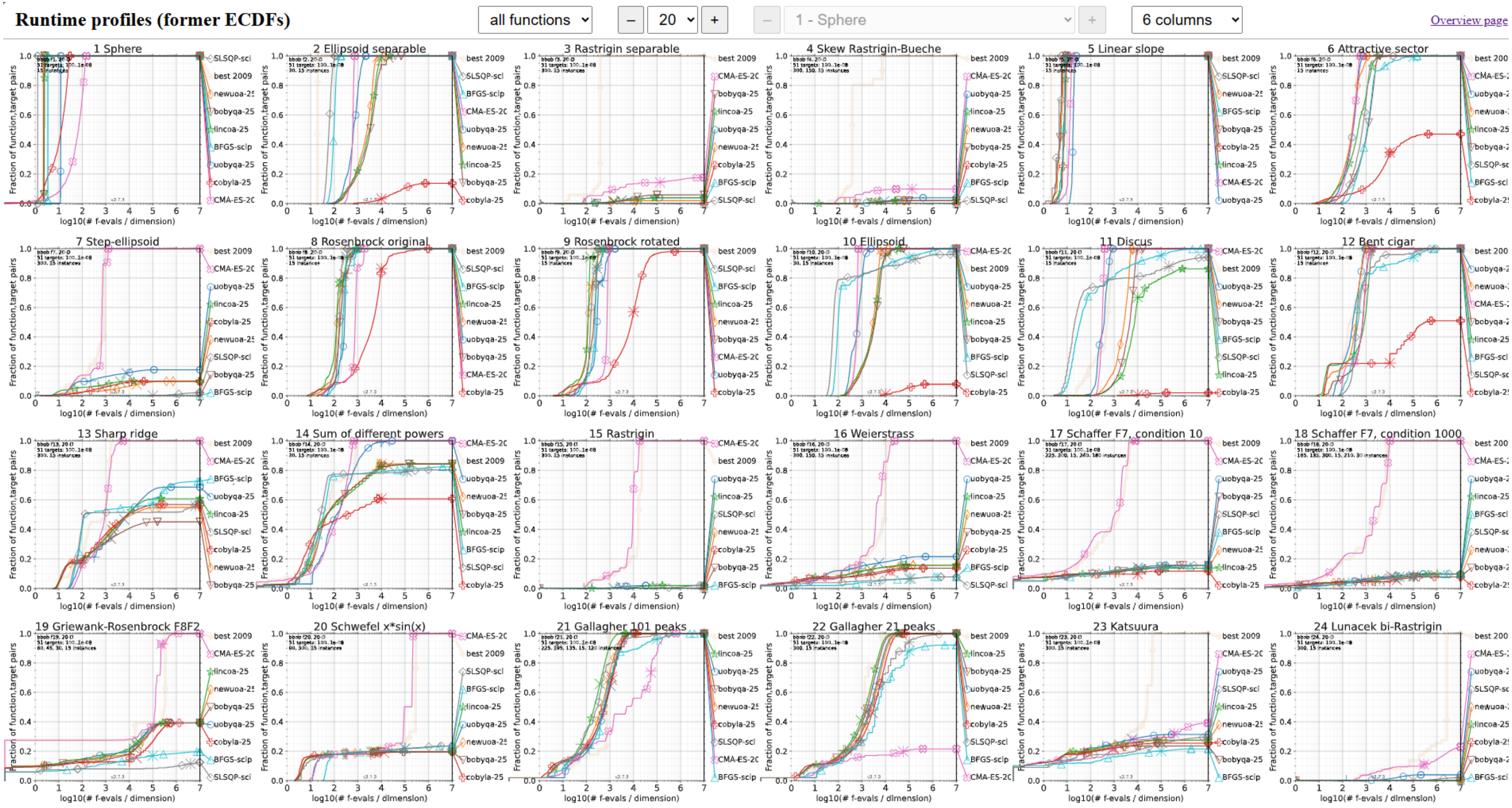
default settings in pdfo except

- trust region radius (initial: 1, minimum: 10^{-15})
- budget based stopping at $10^4 n$, with budget itself the same



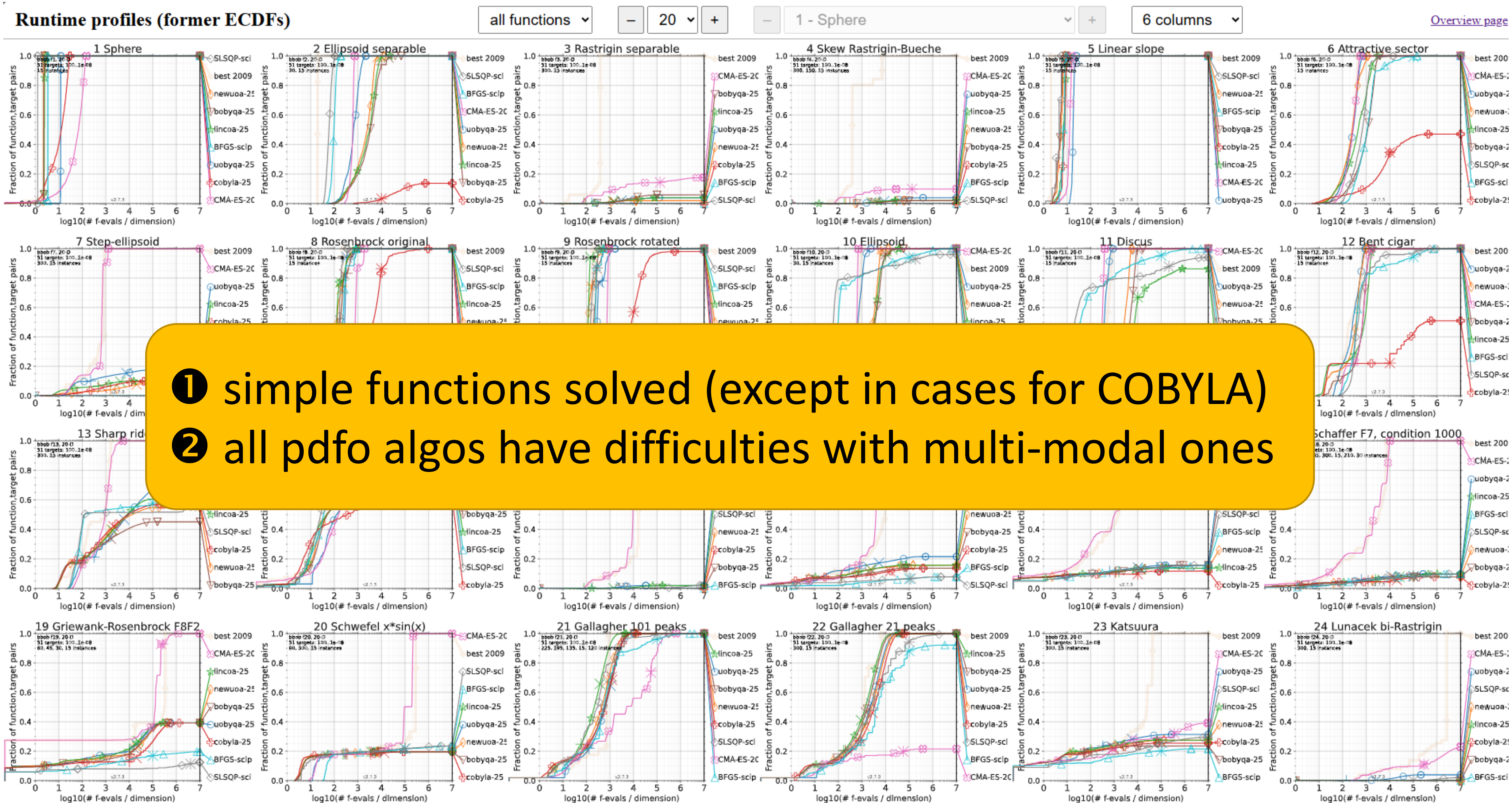


Experiments on bbob (BBOB-Paper #1)



Runtime profiles (former ECDFs)

Experiments on bbob (BBOB-Paper #1)



Runtime profiles (former ECDFs)

all functions ▾ – 20 ▾ + – 1 - Sphere ▾ + 6 columns ▾

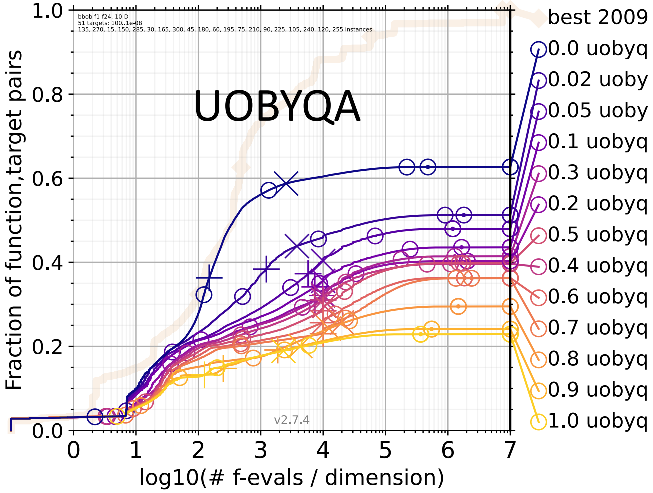
Overview page

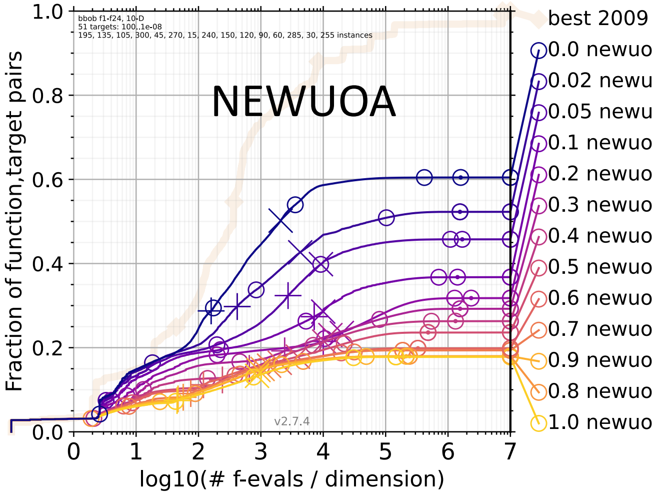
❶ simple functions solved (except in cases for COBYLA)

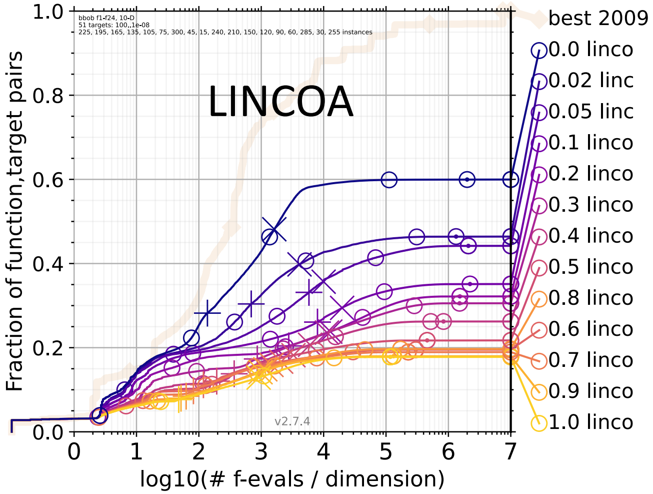
❷ all pdfo algos have difficulties with multi-modal ones

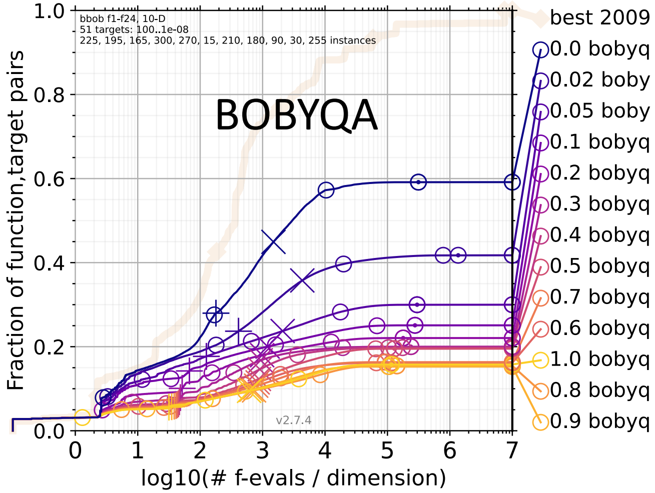
Experiments with Outliers (BBOB-Paper #2)

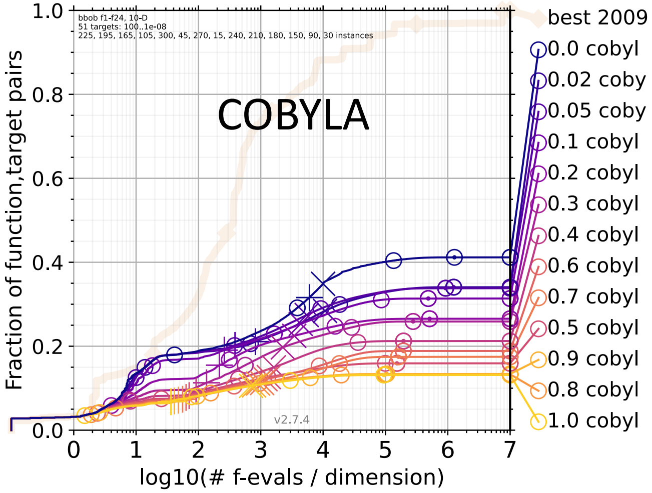
- Only UOBYQA in BBOB paper, but results for all solvers available
- Same outlier scenario(s) as Alexandre in previous talk



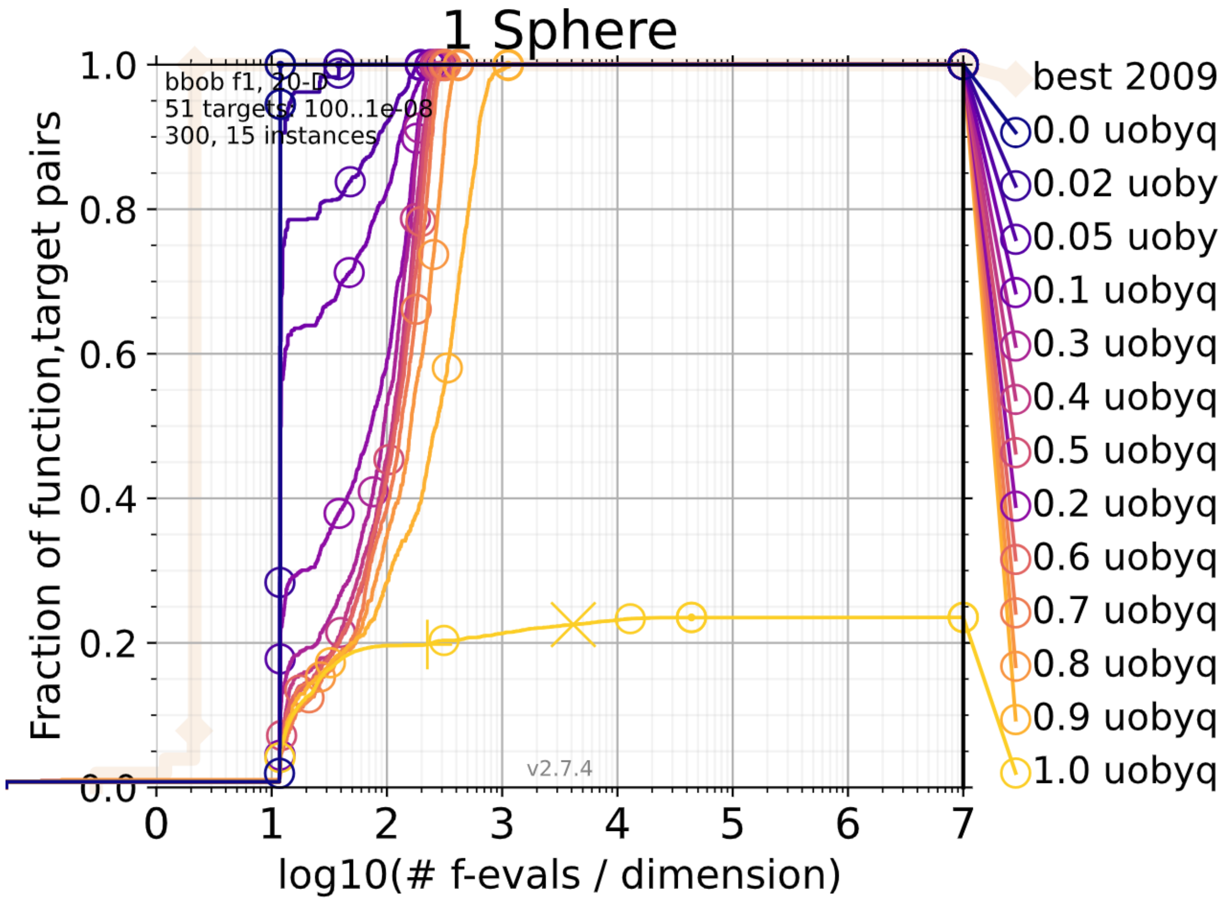


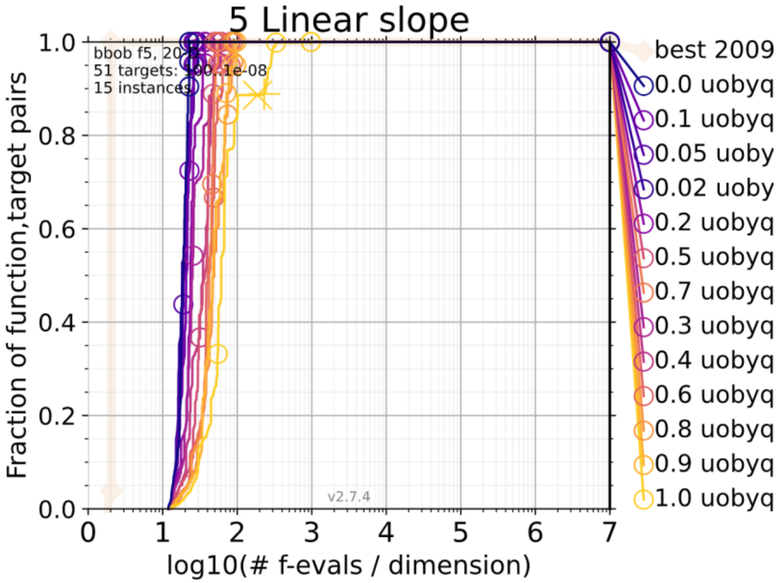


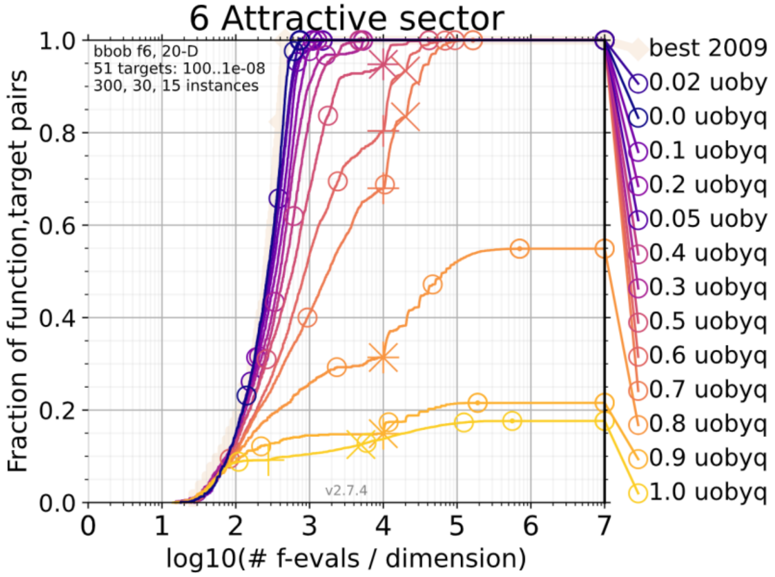




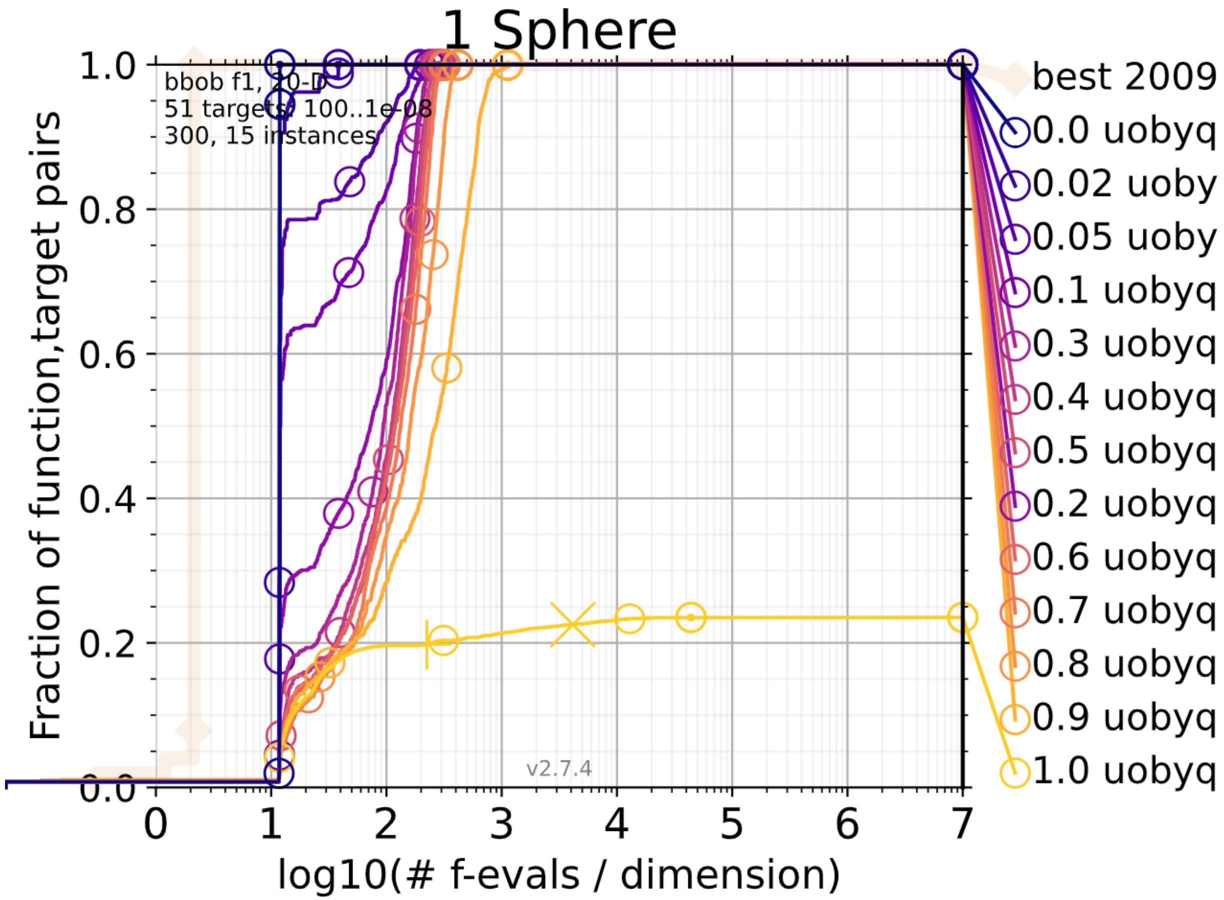
Experiments with Outliers (BBOB-Paper #2)

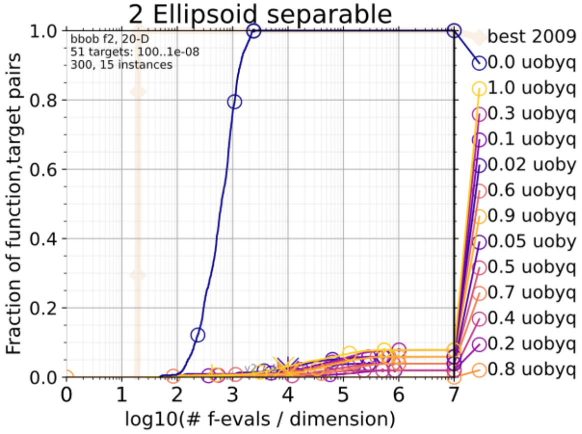


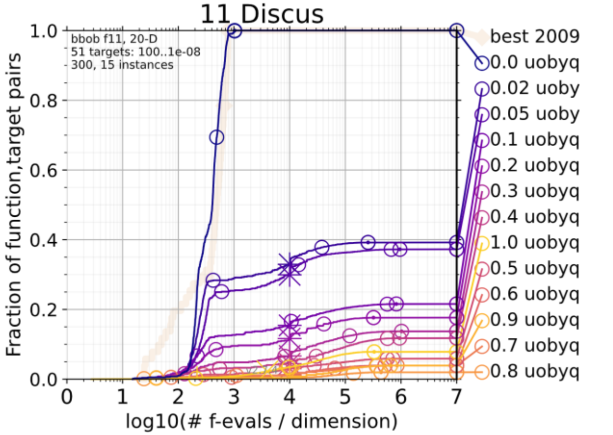


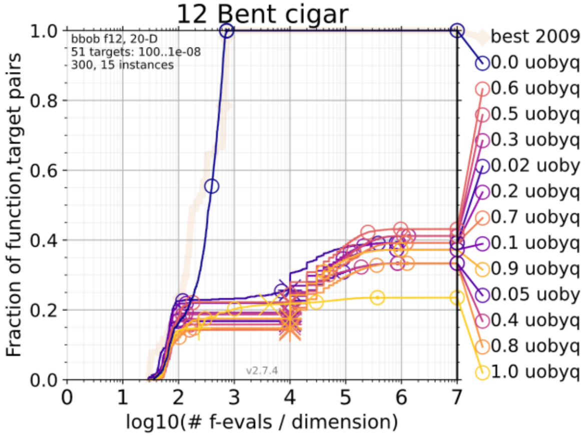


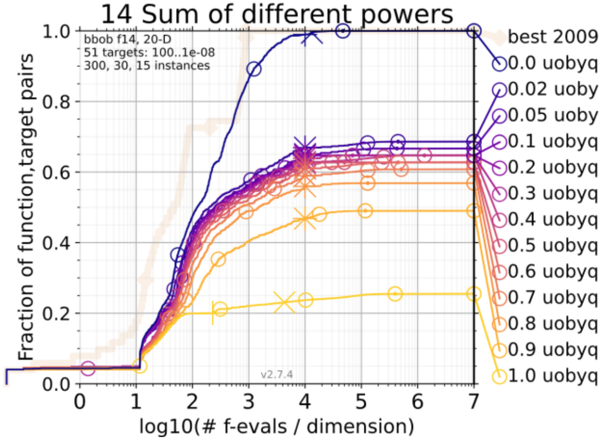
Experiments with Outliers (BBOB-Paper #2)



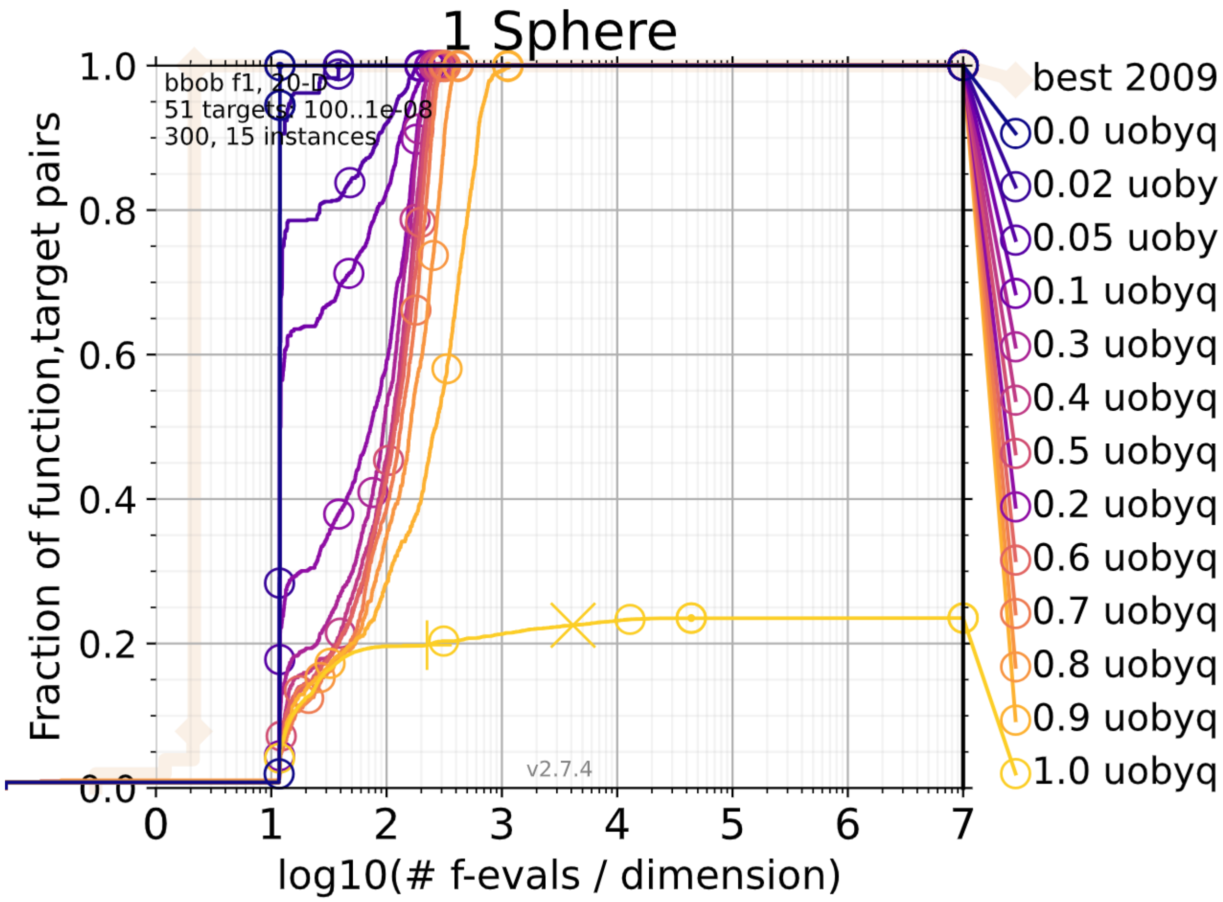


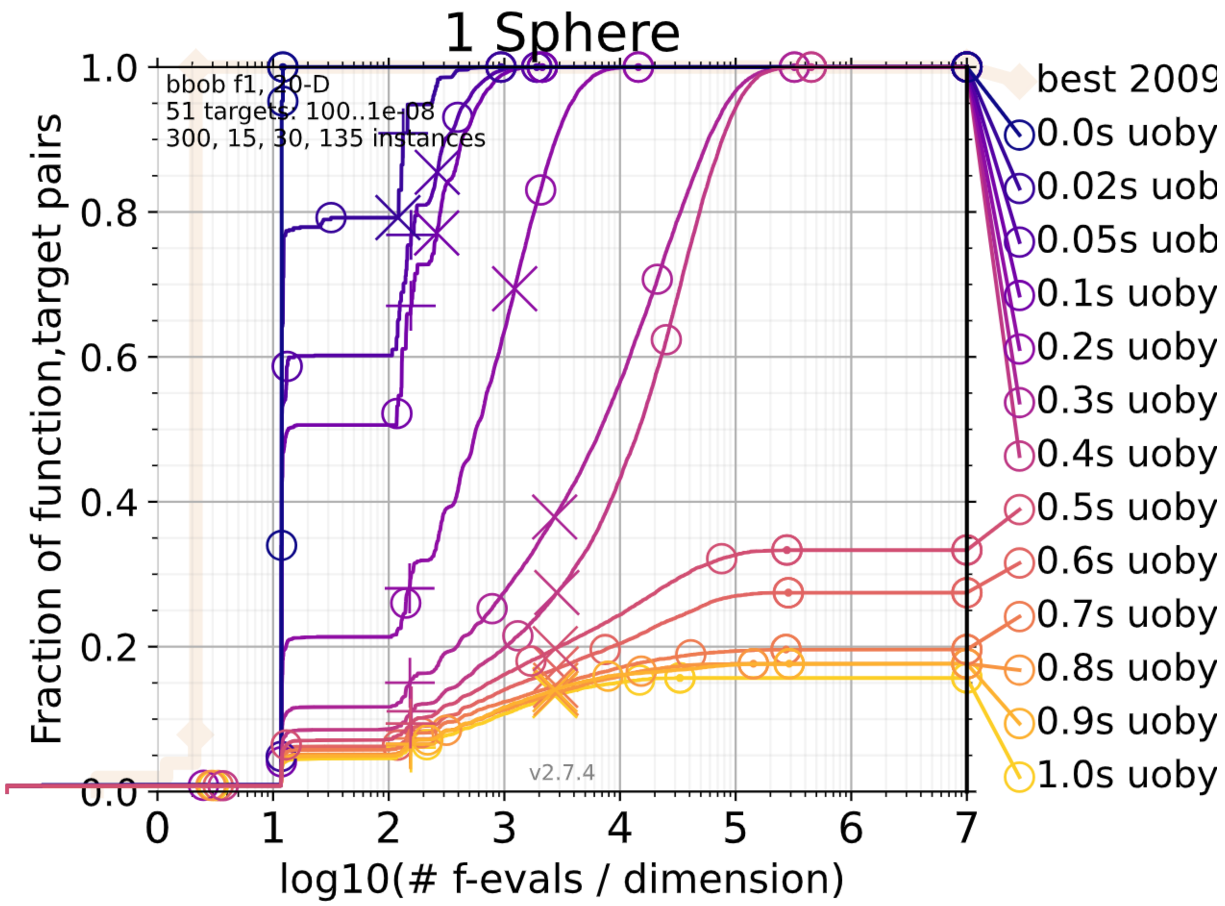






Experiments with Outliers (BBOB-Paper #2)

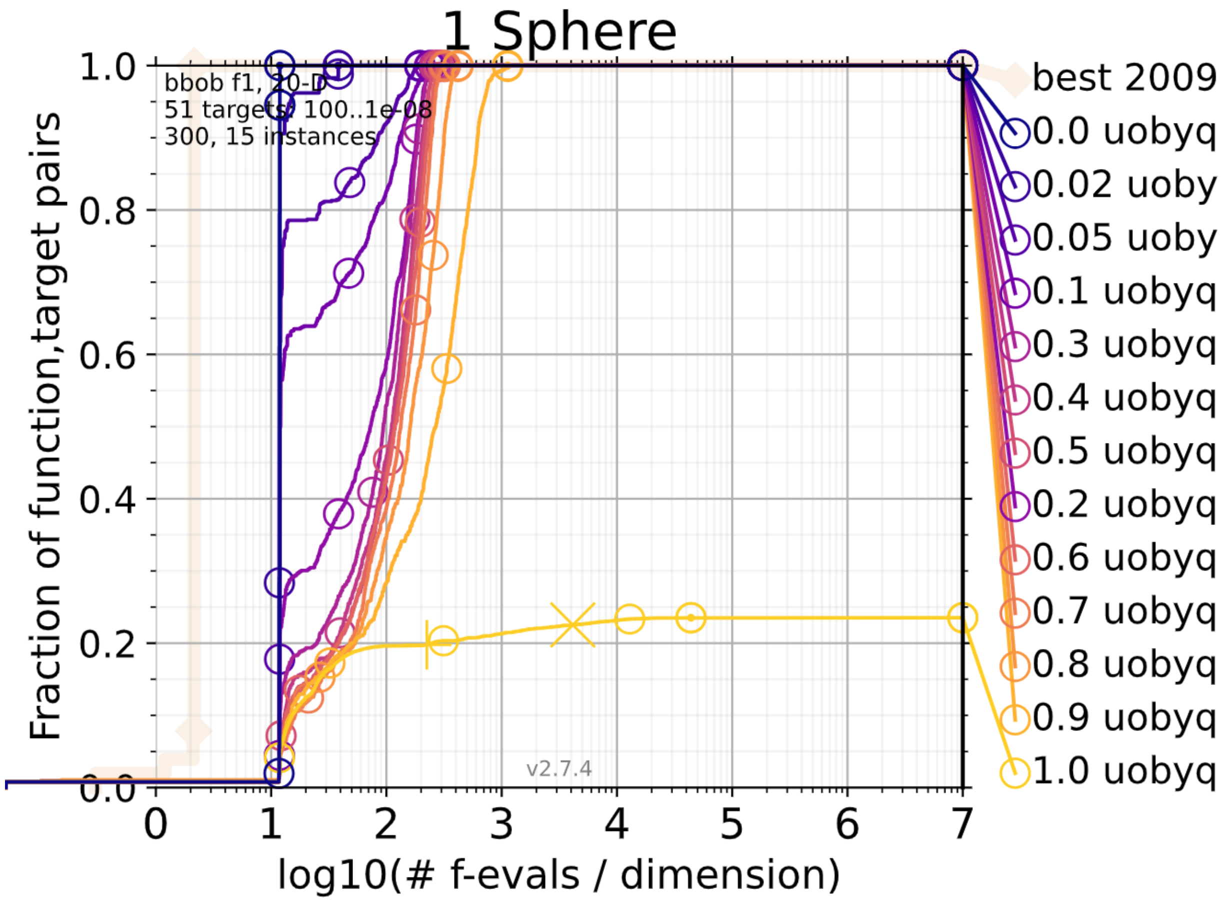




additive half-Cauchy
changing prob. p of outliers

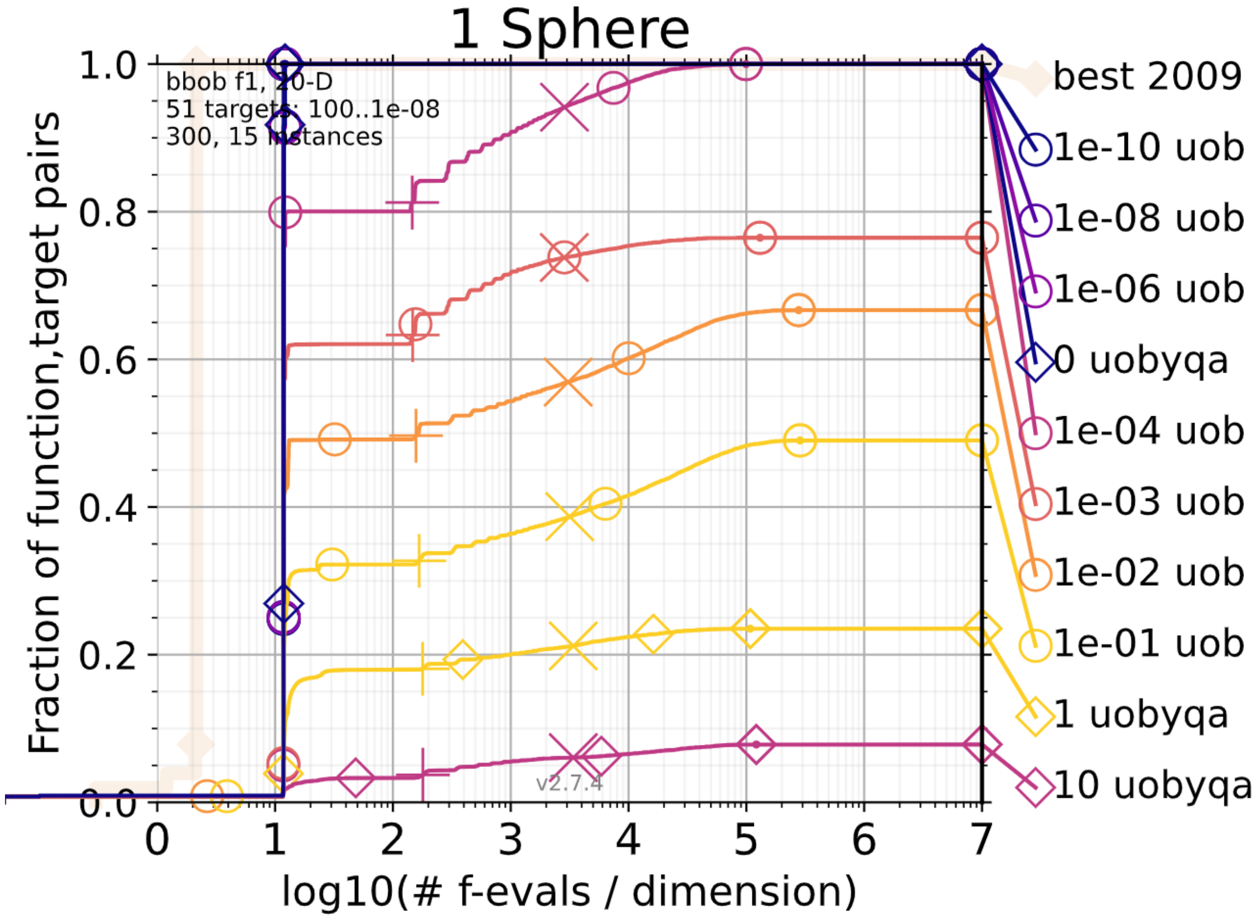
substractive half-Cauchy
changing prob. p of outliers

Experiments with Outliers (BBOB-Paper #2)



additive half-Cauchy

changing prob. p of outliers



Gaussian

changing variance ($p = 1$)

Conclusions

We compared BOBYQA, COBYLA, LINCOA, NEWUOA, and UOBYQA

Noiseless bbob

- simpler **bbob** functions can be solved (except for COBYLA)
- multi-modal functions difficult (except Gallagher)
- UOBYQA with best performance

no real difference between original Powell implementations in Fortran 77 and latest pdfo one

bbob with Noise/Outliers

- more sensitive than BFGS (talk by Alexandre) and CMA-ES (talk by Oskar)
- on some functions *extremely robust to additive outliers*!

sphere, linear, attractive sector

- but not anymore robust wrt. substractive outliers and Gaussian noise