A Brief Introduction to (Evolutionary) Multiobjective Optimization

Dimo Brockhoff
dimo.brockhoff@inria.fr
http://researchers.lille.inria.fr/~brockhof/

A Brief Introduction to (Evolutionary) Multiobjective Optimization

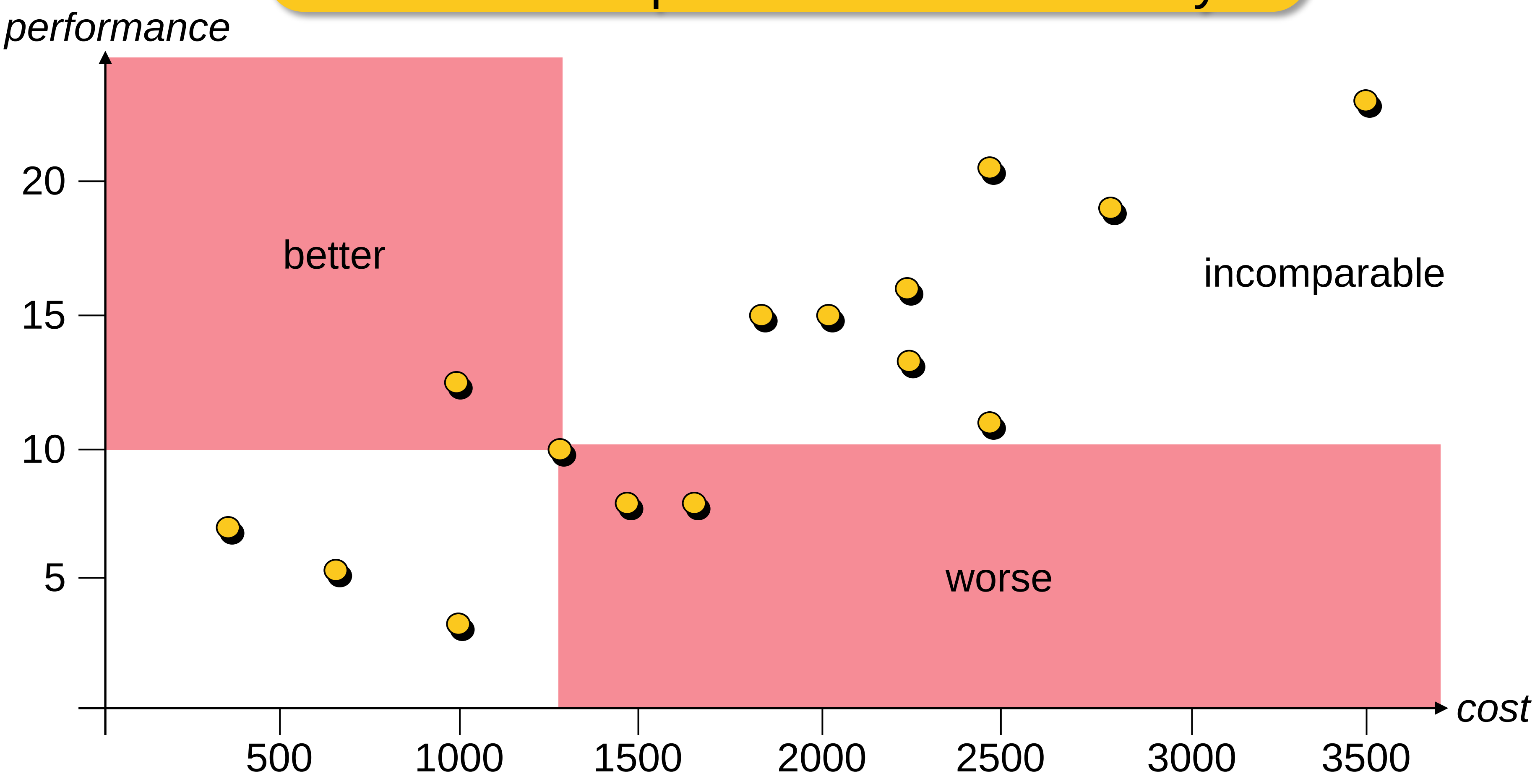
+ (even shorter) Intro to Coco Framework

Dimo Bro
dimo.brockhoff@inria.fr
http://researchers.lille.inria.fr/~brockhof/

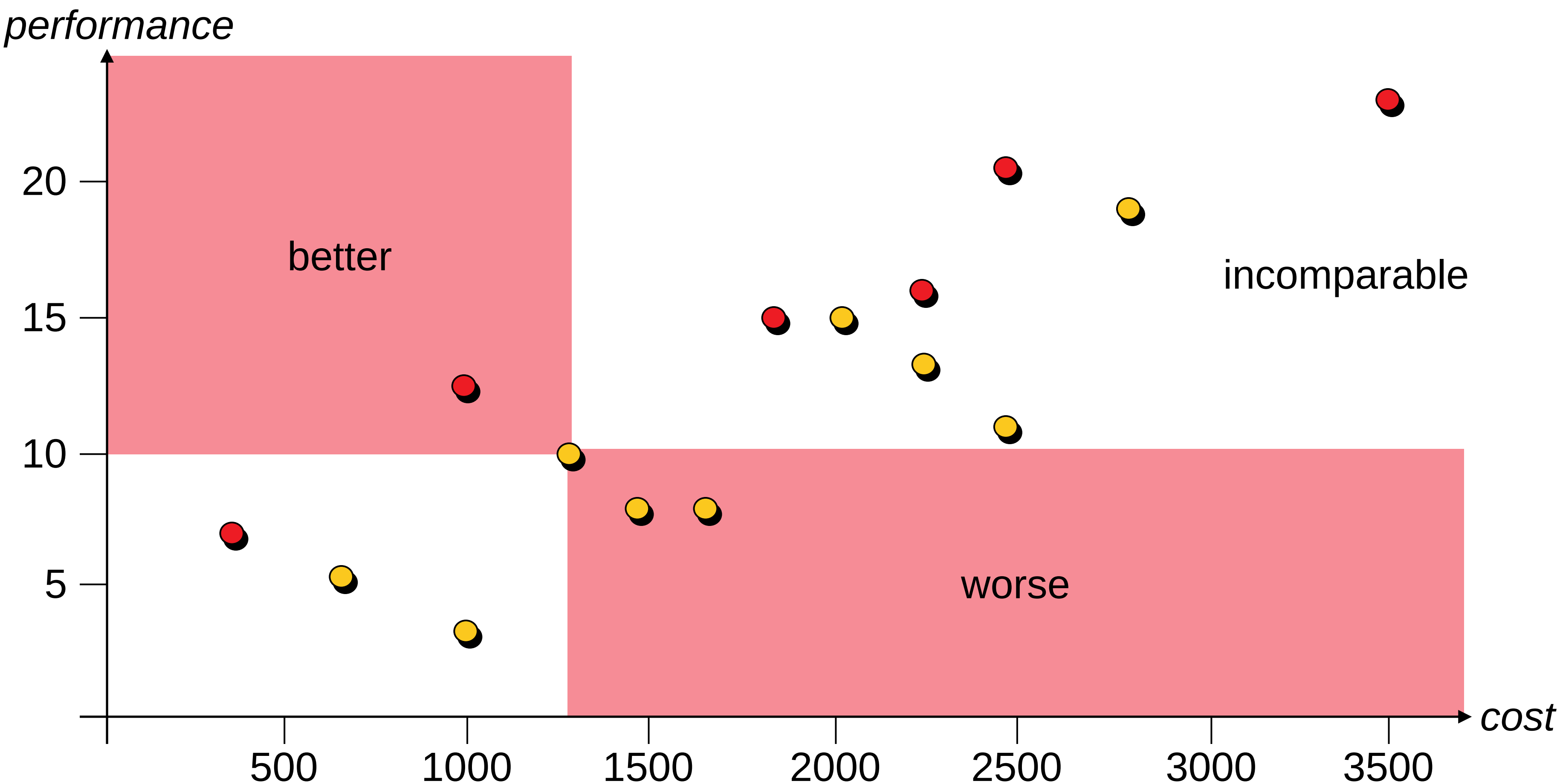
INVENTORS FOR THE DIGITAL WORLD

Multiobjective Optimization:

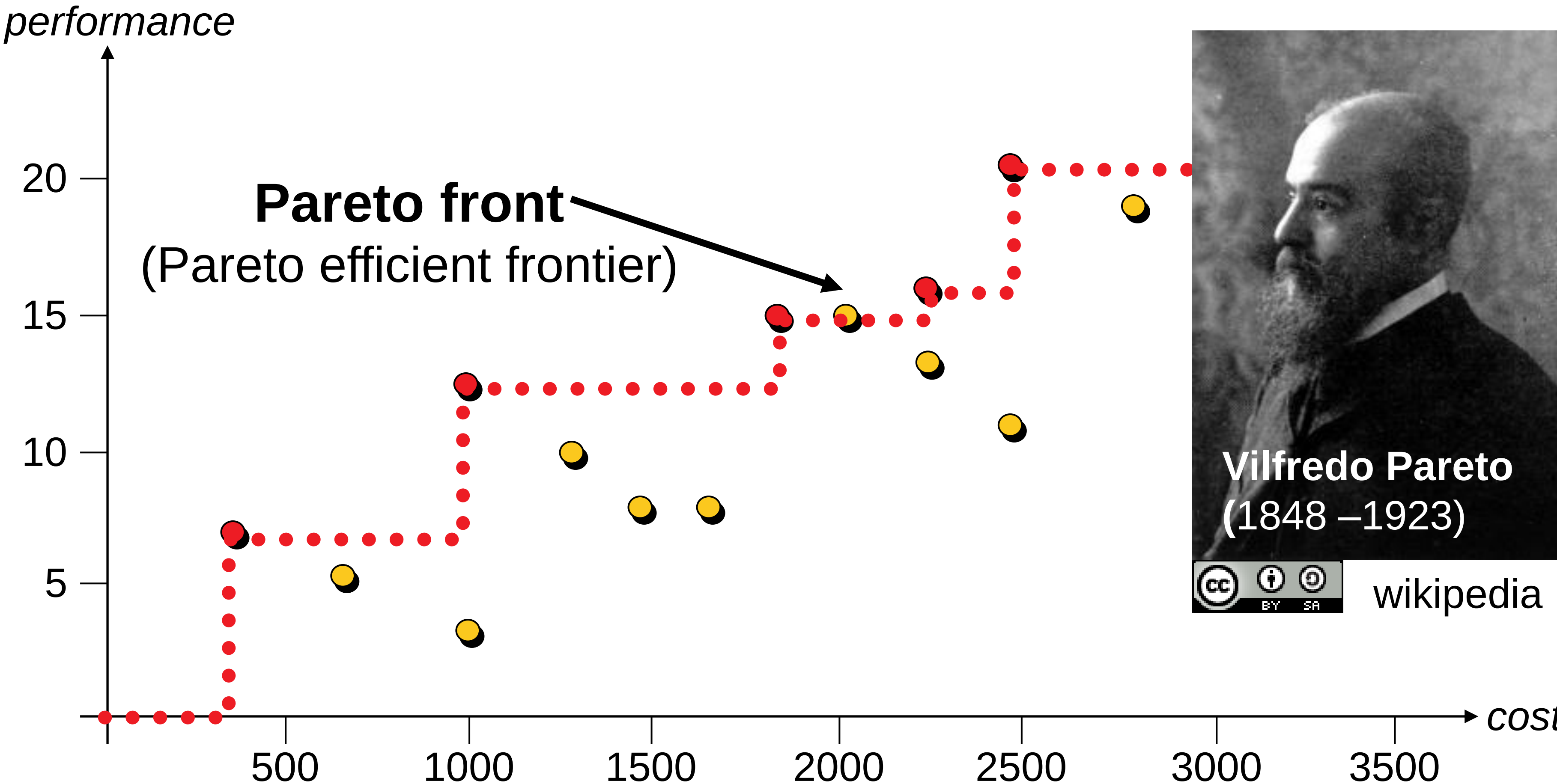
problems where multiple objectives have to be optimized simultaneously



Observations: ❶ there is no single optimal solution, but
❷ some solutions (●) are better than others (●)

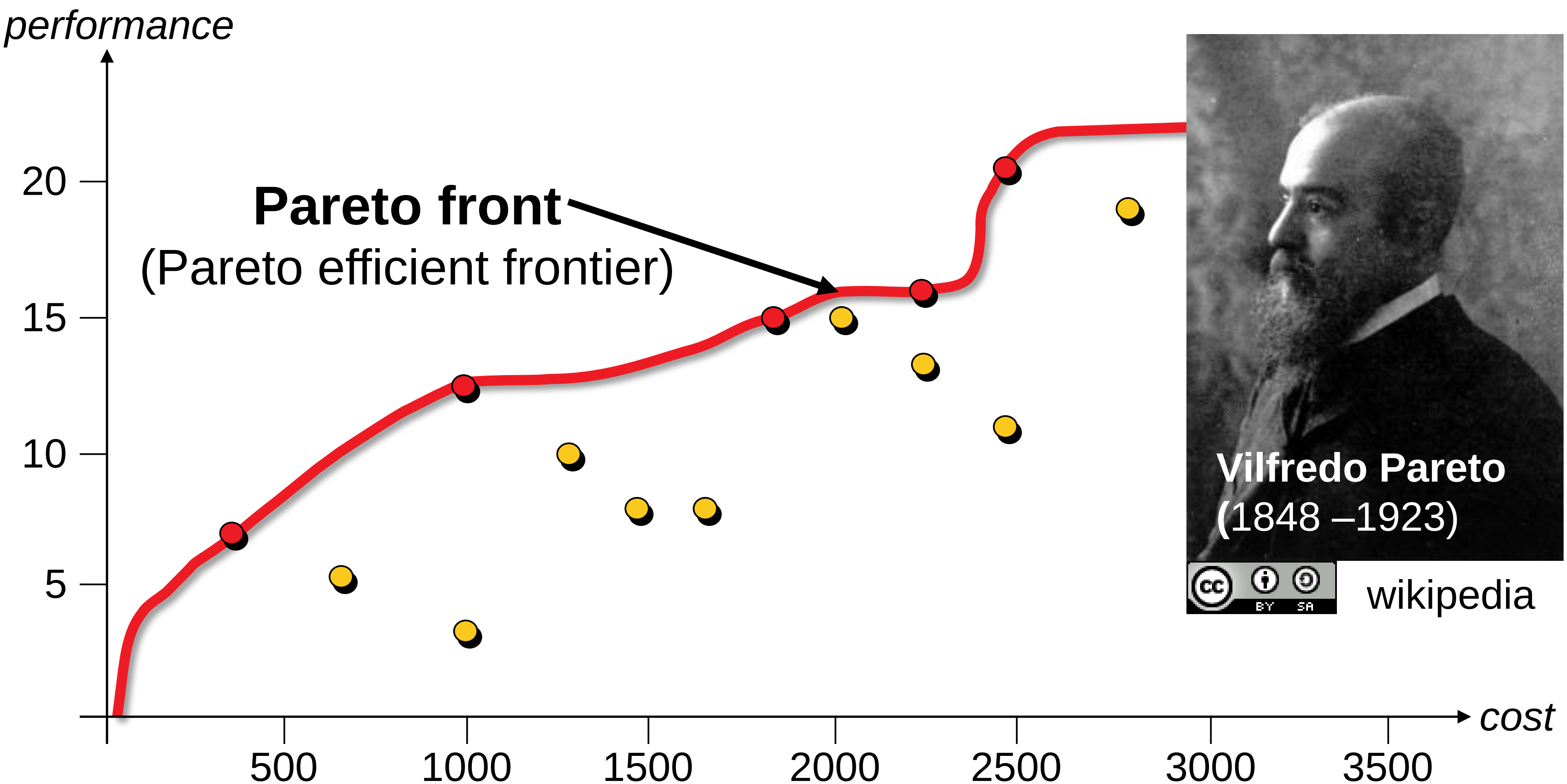


Observations: ❶ there is no single optimal solution, but
❷ some solutions (●) are better than others (●)

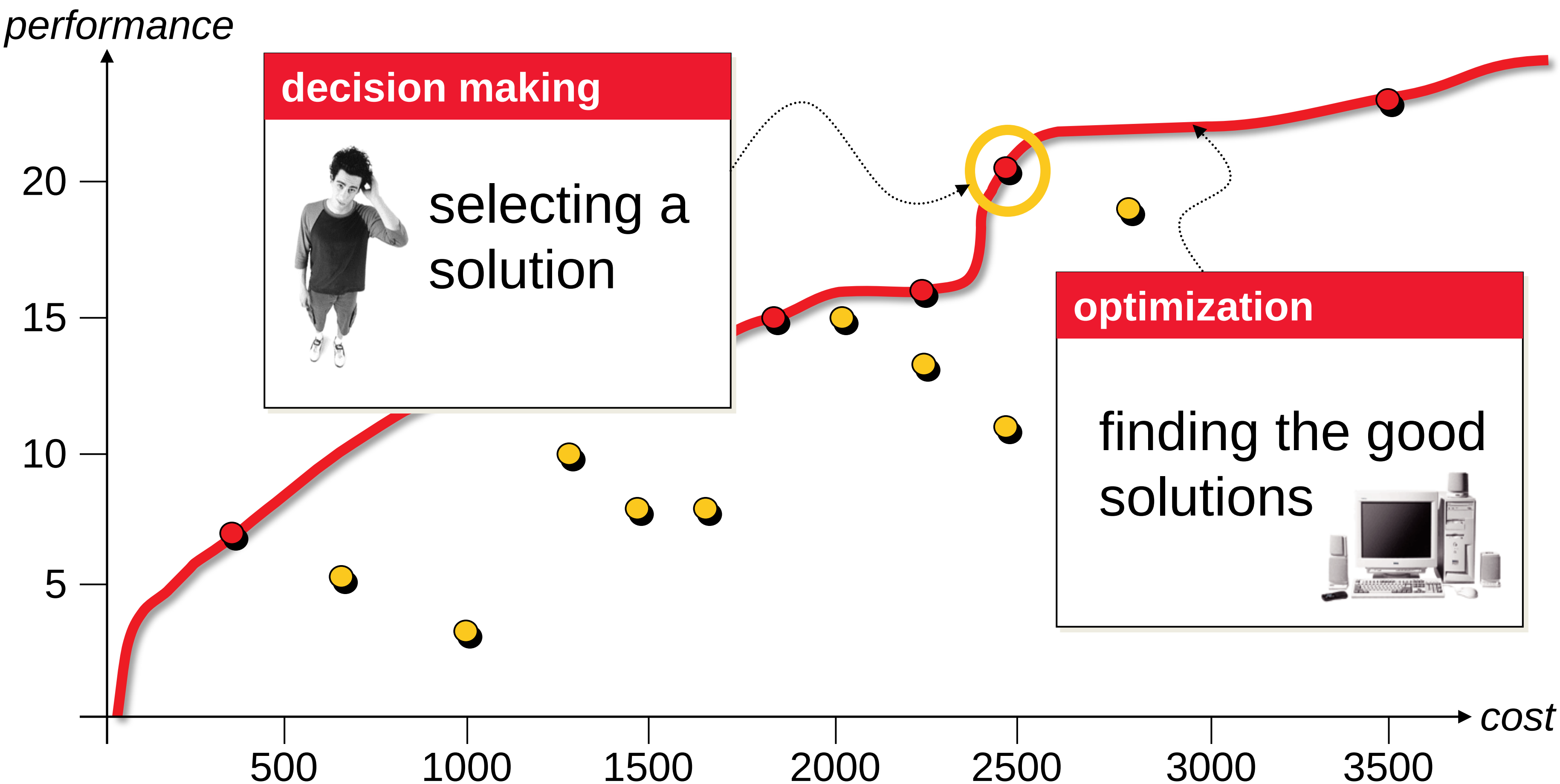


 wikipedia

Observations: ❶ there is no single optimal solution, but
❷ some solutions (●) are better than others (●)

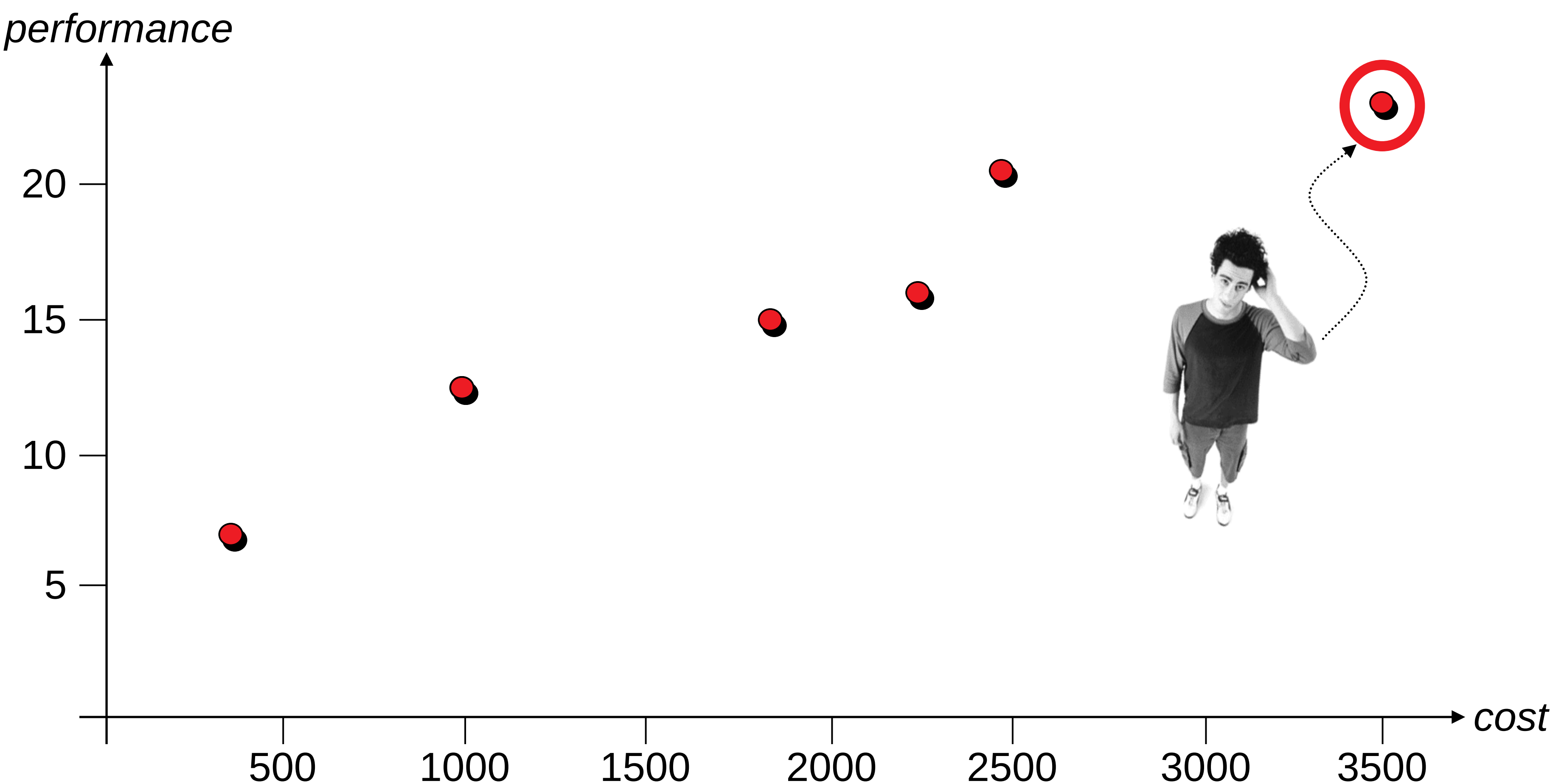


Observations: ❶ there is no single optimal solution, but
❷ some solutions (●) are better than others (●)



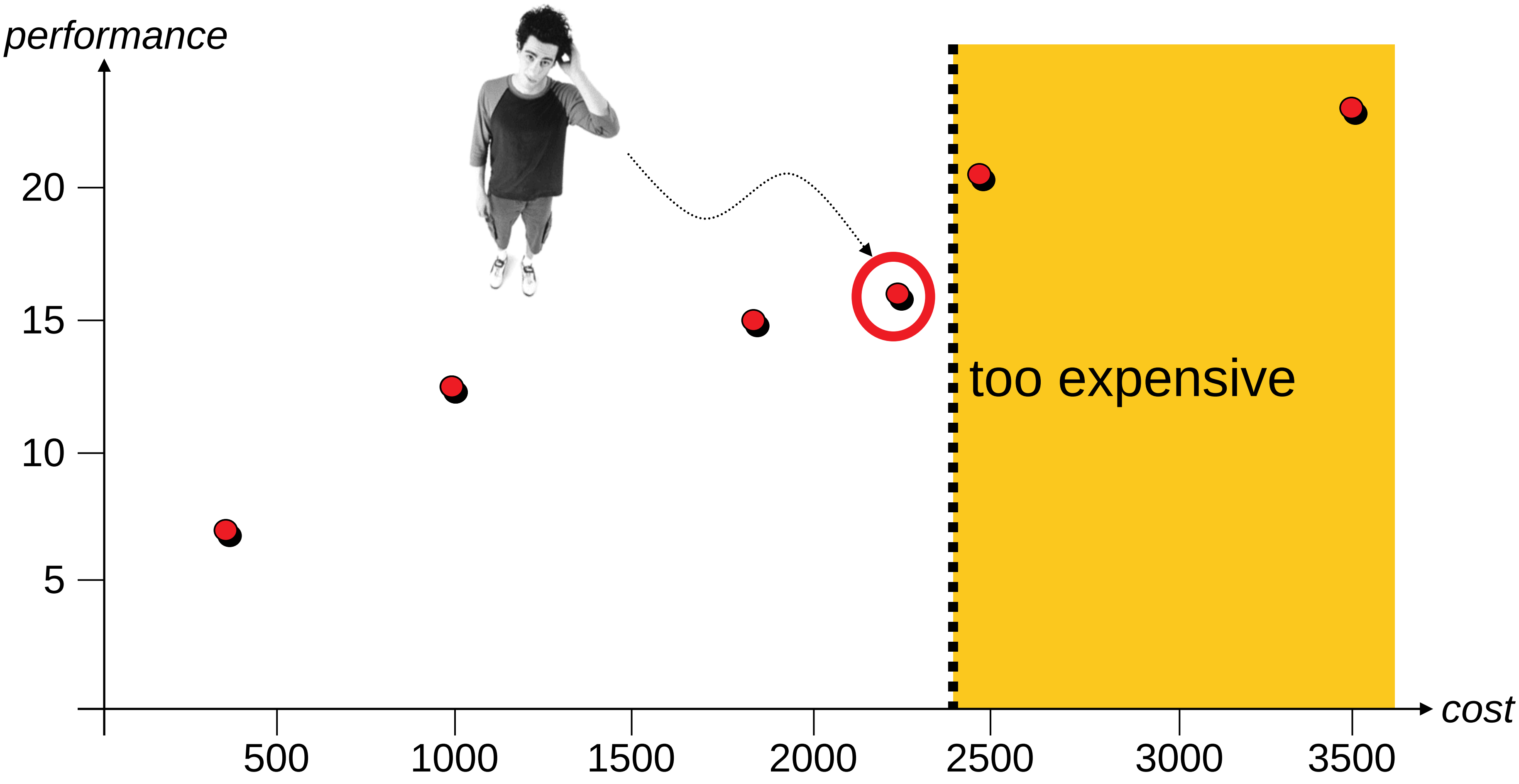
Possible Approaches:

❶ **ranking:** performance more important than cost



Selecting a Solution: Examples

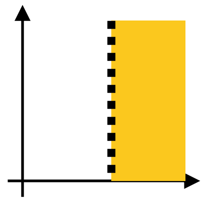
Possible Approaches:
❶ ranking: performance more important than cost
❷ **constraints:** cost must not exceed 2400



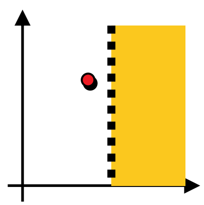
When to Make the Decision

Before Optimization:

rank objectives,
define constraints,…

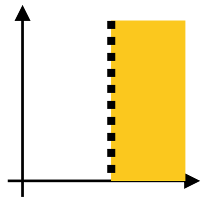


search for one
(good) solution



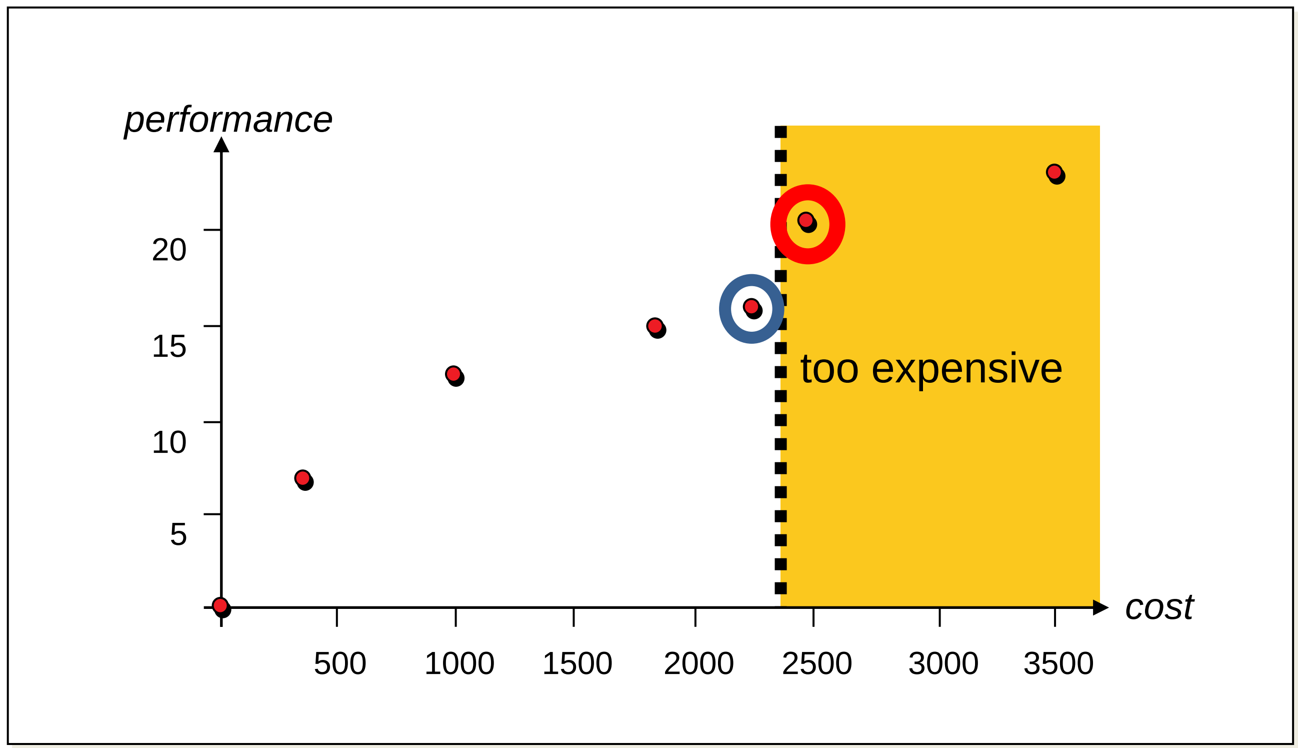
When to Make the Decision

Before Optimization:



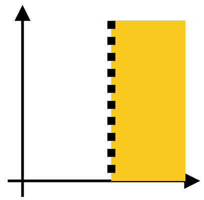
rank objectives,
define constraints,…

search for one
(good) solution

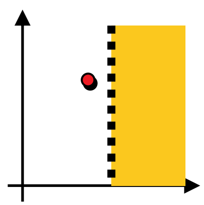


When to Make the Decision

Before Optimization:

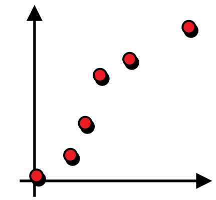


rank objectives,
define constraints,…

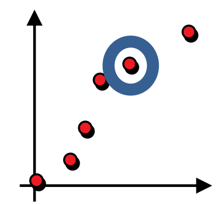


search for one
(good) solution

After Optimization:



search for a set of
(good) solutions

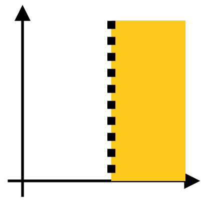


select one solution
considering
constraints, etc.

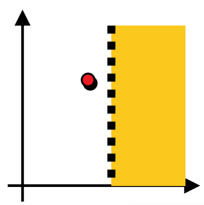
When to Make the Decision

Before Optimization:

rank objectives,
define constraints,…

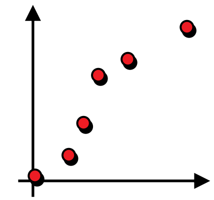


search for one
(good) solution

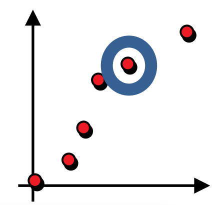


After Optimization:

search for a set of
(good) solutions



select one solution
considering
constraints, etc.



Focus: learning about a problem

- trade-off surface
- interactions among criteria
- structural information
- also: interactive optimization

Two Communities...

MCDM International Society on Multiple Criteria Decision Making

- beginning in 1950s/1960s
- bi-annual conferences since 1975
- background in economics, math, management science
- both optimization and decision making

emo

- quite young field (first papers in mid 1980s)
- bi-annual conference since 2001
- background evolutionary computation (applied math, computer science, engineering, ...)
- focus on optimization algorithms

...Slowly Merge Into One

MCDM International Society on Multiple Criteria Decision Making

emo

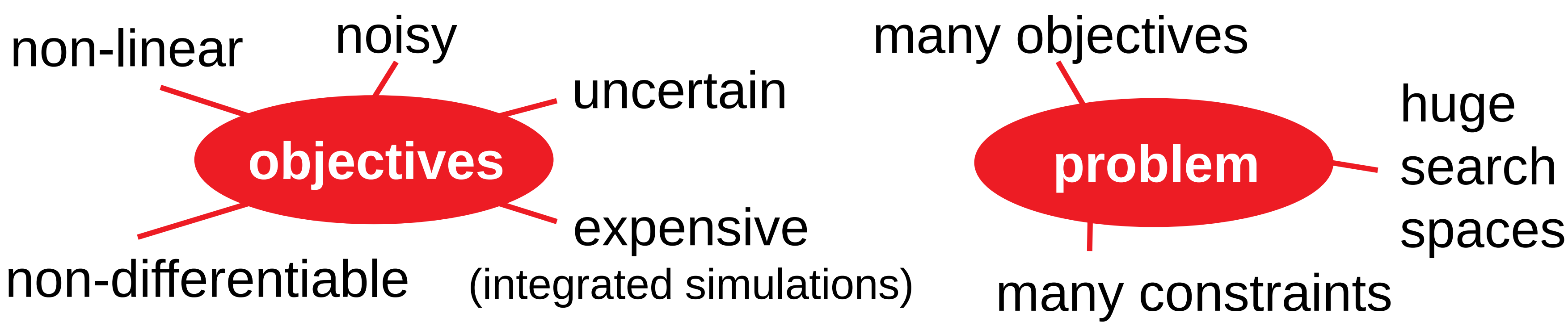
- MCDM track at EMO conference since 2009
- special sessions on EMO at the MCDM conference since 2008
- joint Dagstuhl seminars since 2004

Blackbox optimization

$$x \in S \longrightarrow \boxed{f} \longrightarrow (f_1(x), \ldots, f_k(x))$$

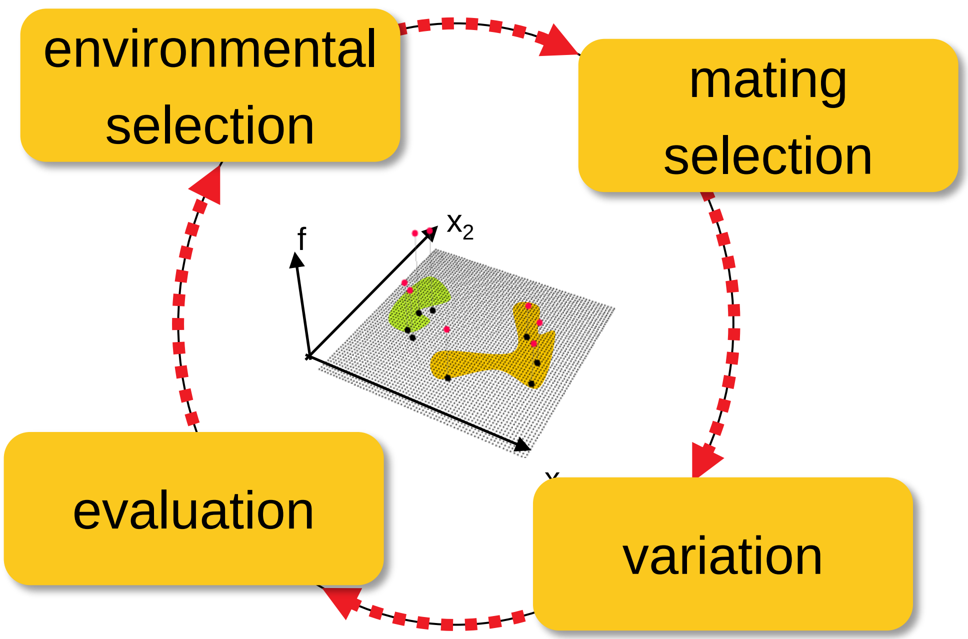
only mild assumptions

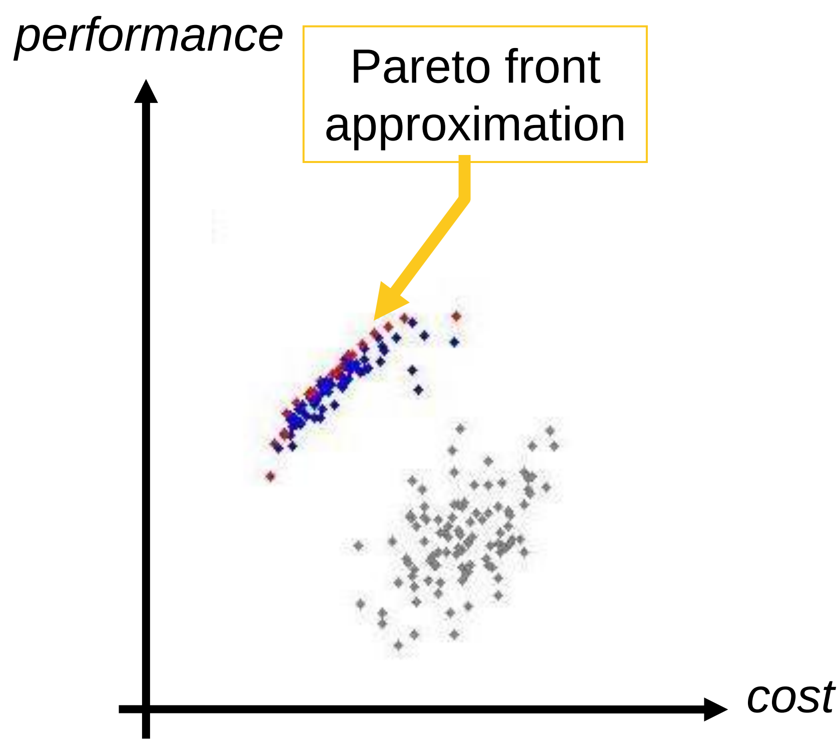
→ EMO therefore well-suited for real-world (engineering) problems



Evolutionary Multiobjective Optimization

- set-based algorithms
- therefore possible to approximate the Pareto front in one run





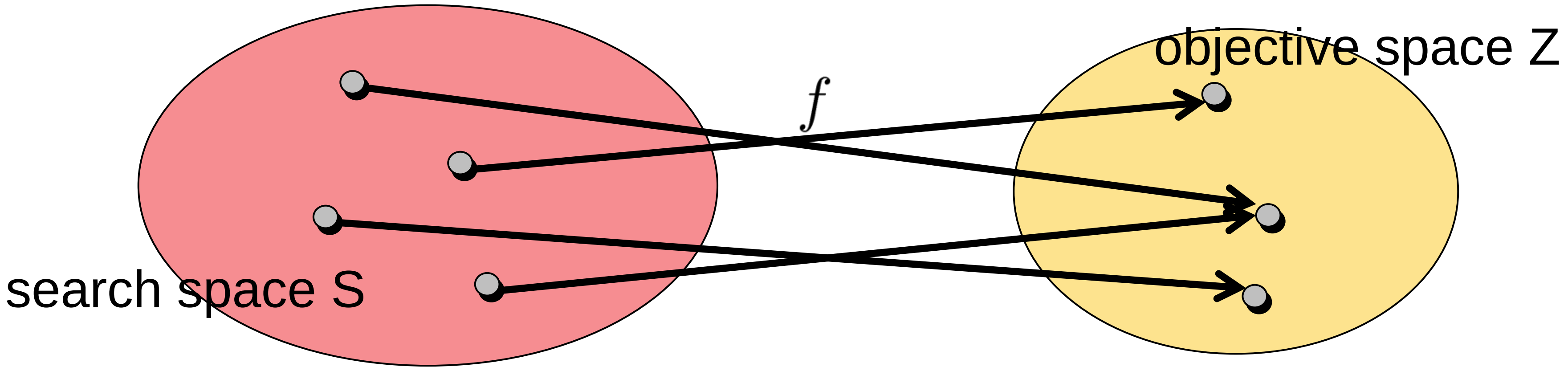
Overview

The Big Picture

Basic Principles of Multiobjective Optimization

- algorithm design principles and concepts
- performance assessment & benchmarking

Differences To Single-Objective Problems



single objective

solutions always comparable

clearly defined optimum

with a better than b
if $f(a) \leq f(b)$

multiple objectives

solutions not always comparable

set of Pareto-optimal solutions

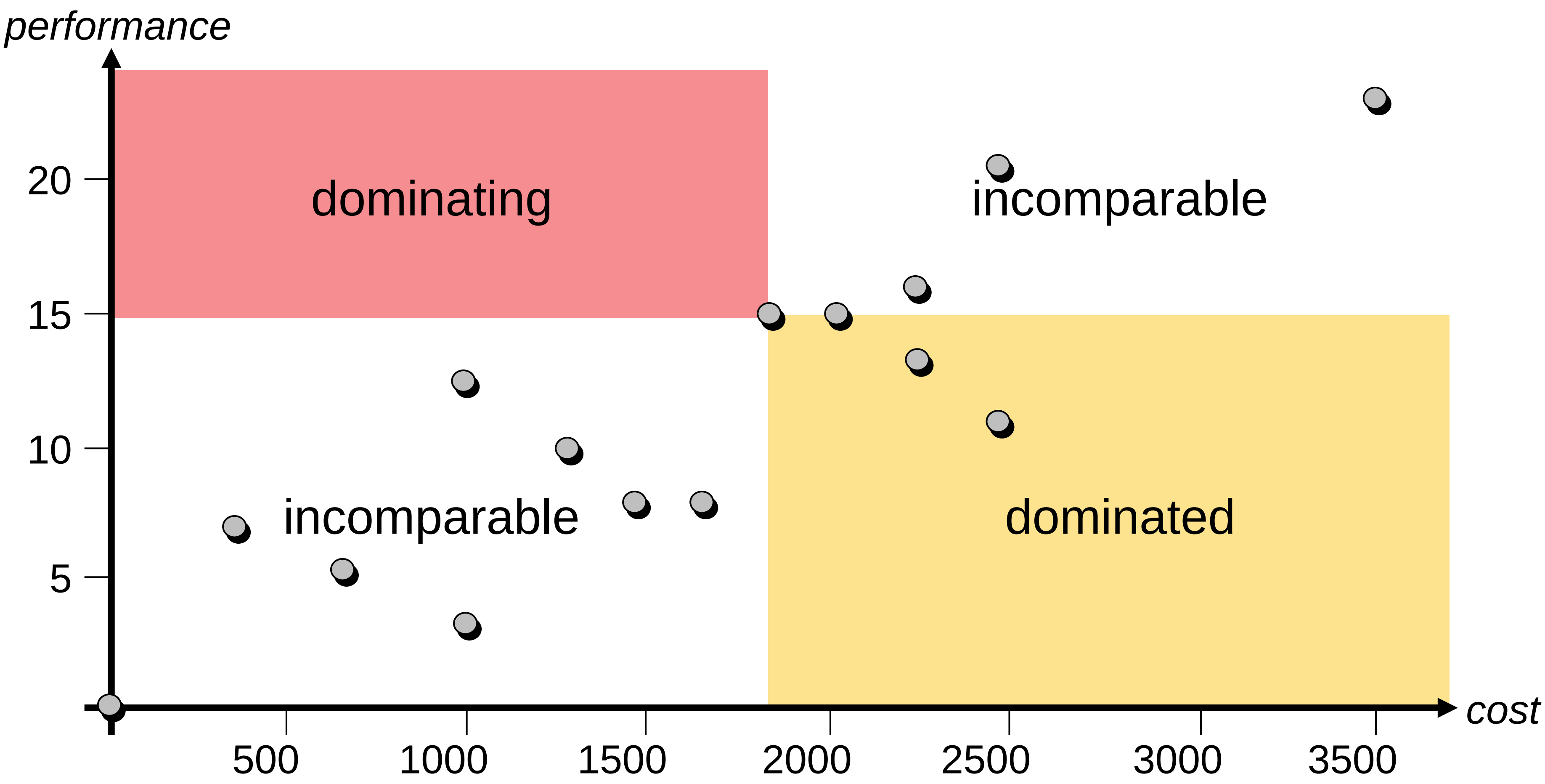
with a better than b
if $f(a)$ $\mathrm{prefrel}$ $f(b)$

even more complicated:
sought are **sets**!

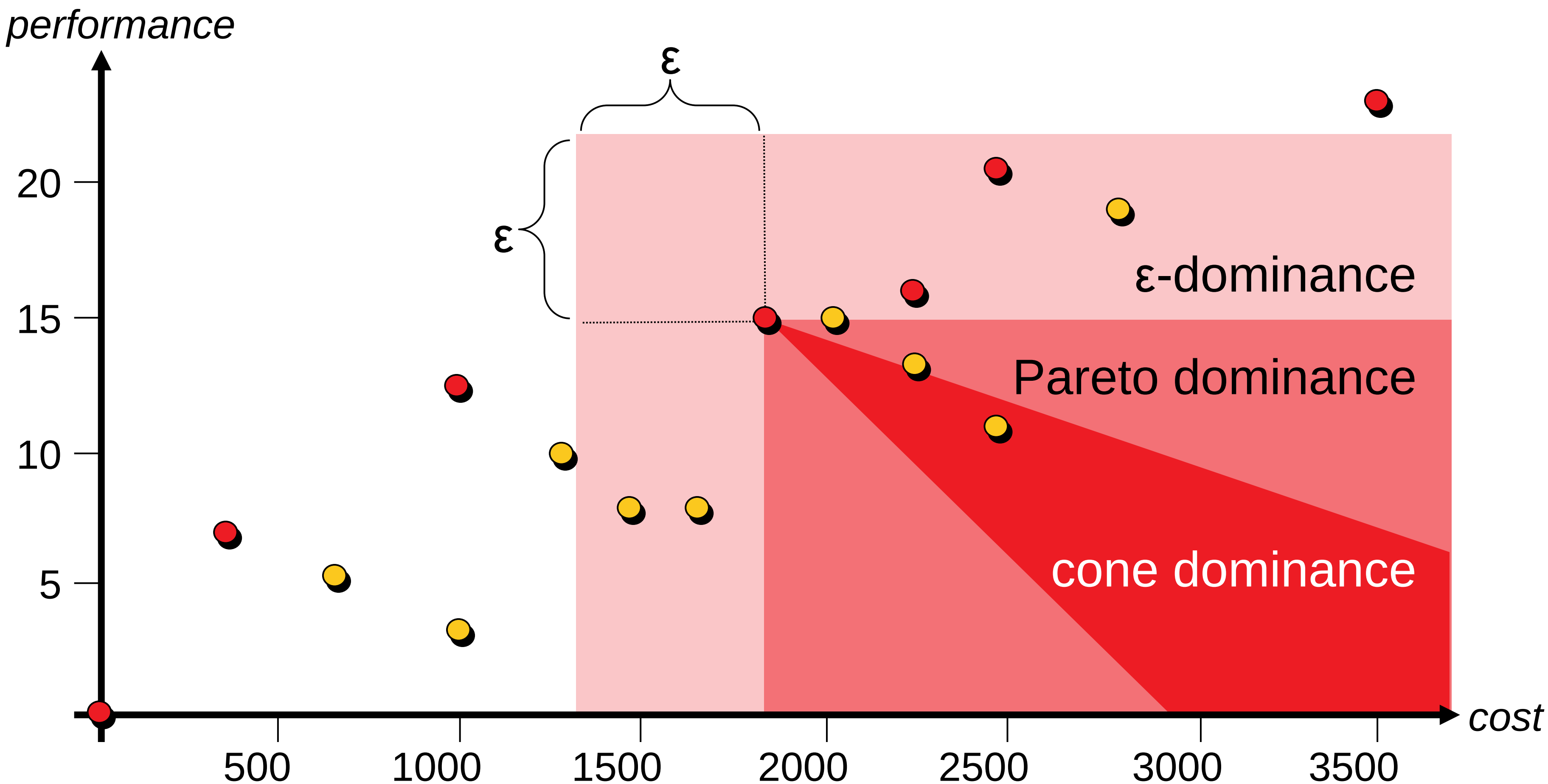
Most Common Example: Pareto Dominance

$$u \text{ } \textit{weakly Pareto dominates } v \ (u \leqslant_{par} v): \quad \forall 1 \leq i \leq k : f_i(u) \leq f_i(v)$$

$$u \text{ } \textit{Pareto dominates } v \ (u <_{par} v): \quad u \leqslant_{par} v \ \wedge \ v \not\leqslant_{par} u$$



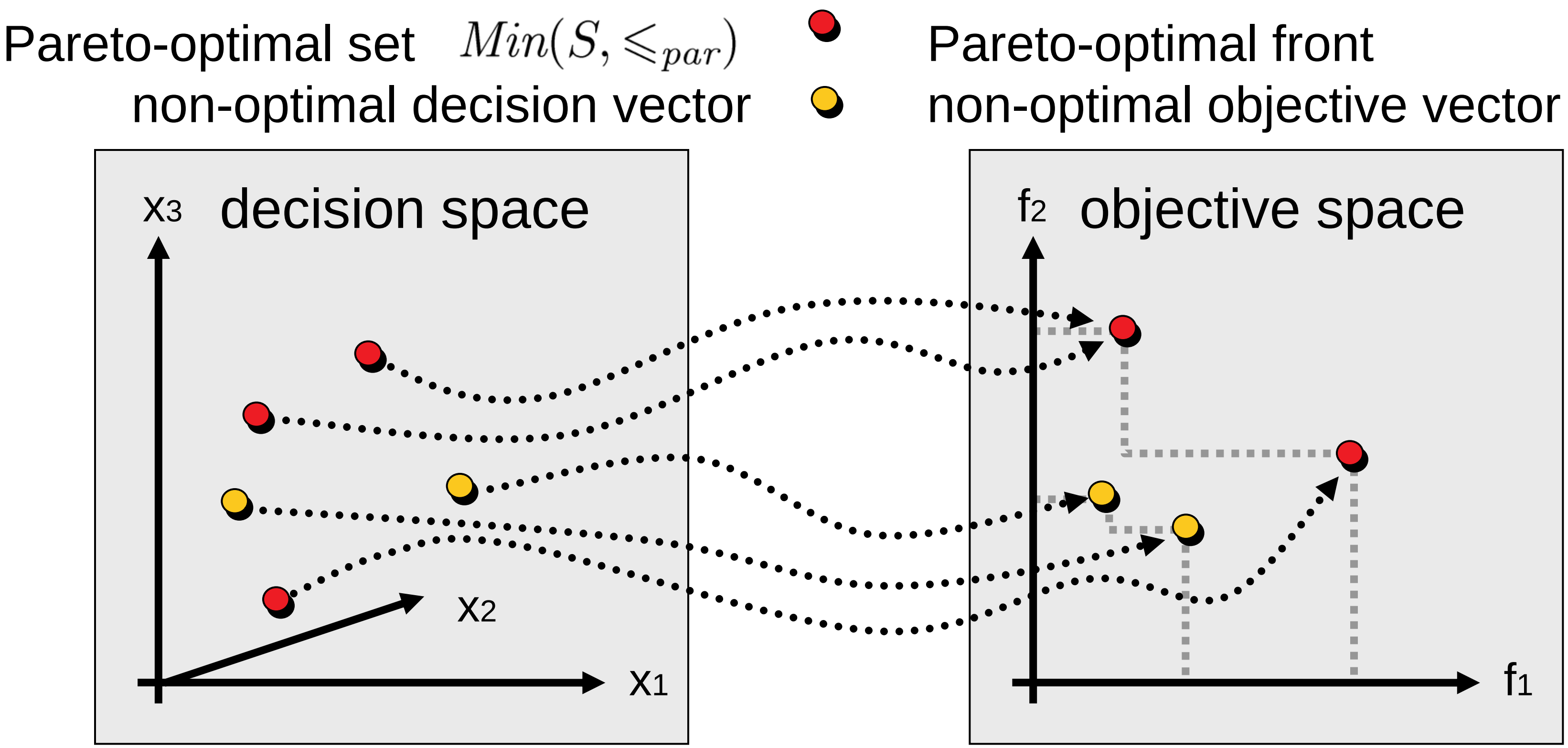
Different Notions of Dominance



Pareto-optimal Set and Pareto(-optimal) Front

The *minimal set* of a preordered set $(Y, \leqq)$ is defined as

$$Min(Y, \leqq) := \{a \in Y \mid \forall b \in Y : b \leqq a \Rightarrow a \leqq b\}$$



Approaches To Multiobjective Optimization

A multiobjective problem is as such underspecified

…because not any Pareto-optimum is equally suited!

Additional preferences are needed to tackle the problem:

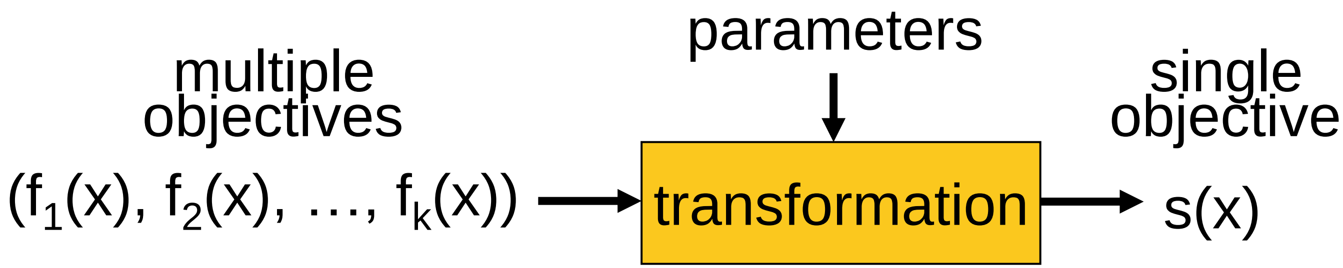
Solution-Oriented Problem Transformation:
Classical approach: Induce a total order on the decision space, e.g., by aggregation

Set-Oriented Problem Transformation:
Recent view on EMO: First transform problem into a set problem and then define an objective function on sets [Zitzler et al. 2010]

Preferences are needed in both cases, but the latter are weaker!

Solution-based Approaches (classical approaches)

Solution-Oriented Problem Transformations

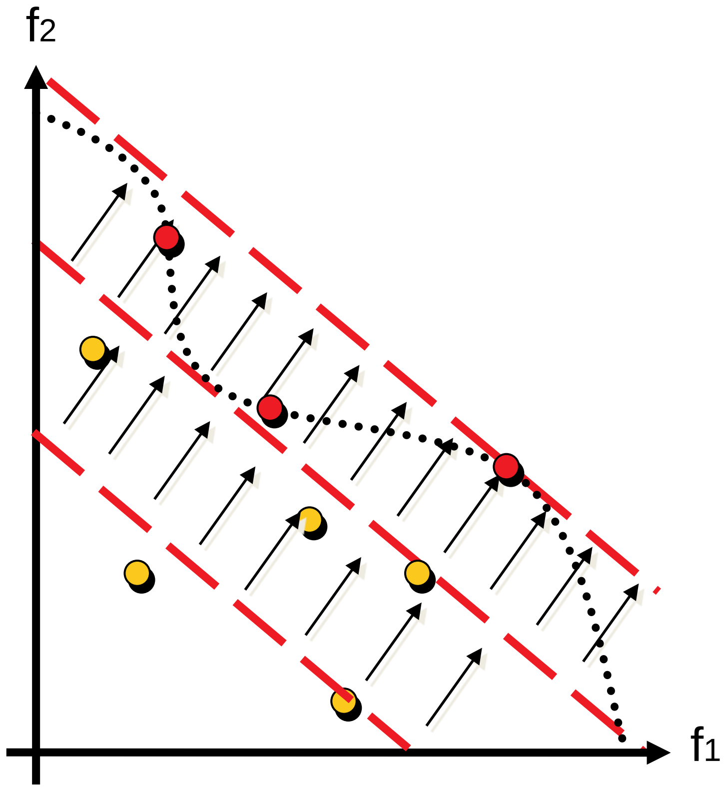


A scalarizing function s is a function $s: Z \to \mathbb{R}$ that maps each objective vector $u = (u_1, \ldots, u_n) \in Z$ to a real value $s(u) \in \mathbb{R}$

Solution-Oriented Problem Transformations

multiple objectives

$(f_1(x), f_2(x), \ldots, f_k(x))$ → parameters → **transformation** → single objective $s(x)$



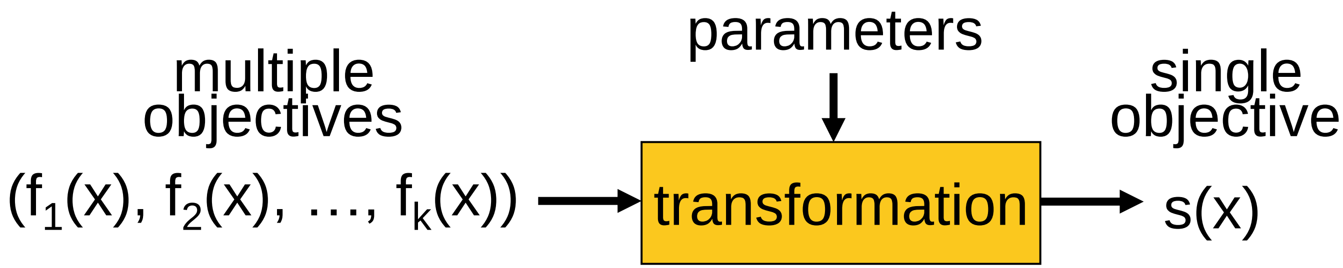
Example 1: weighted sum approach

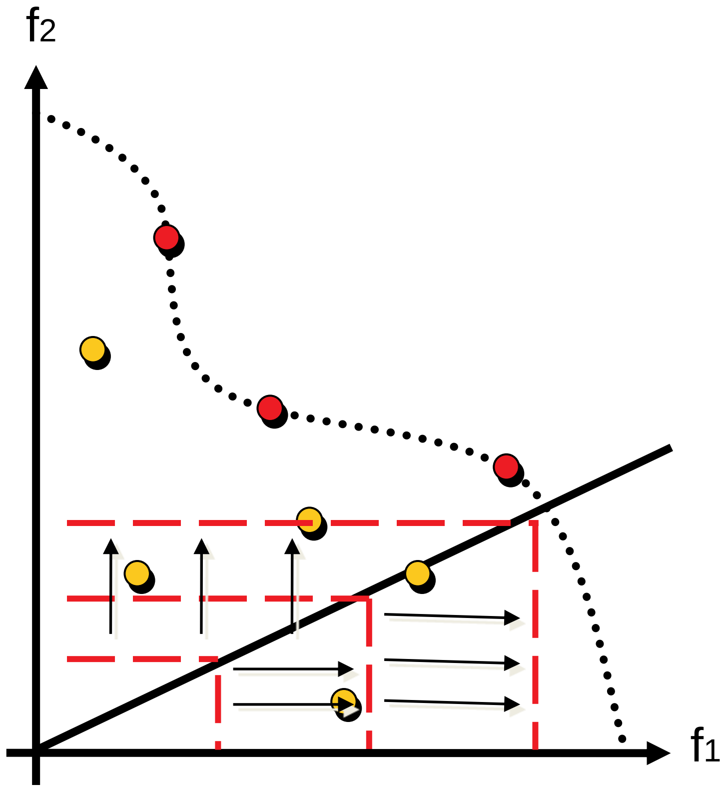
$(w_1, w_2, \ldots, w_k)$

$y = w_1 y_1 + \ldots + w_k y_k$

Disadvantage: not all Pareto-optimal solutions can be found if the front is not convex

Solution-Oriented Problem Transformations





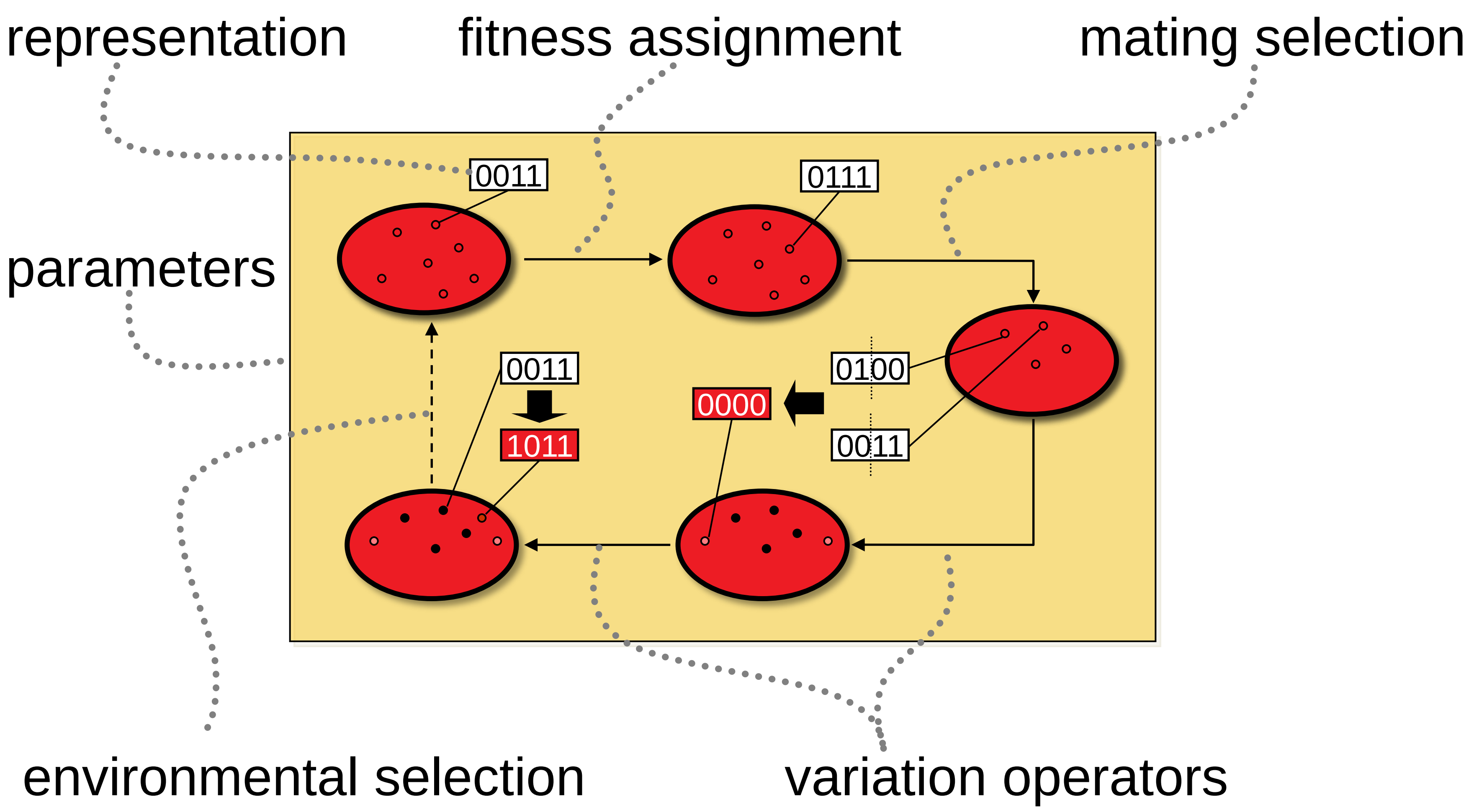
Example 2: weighted Tchebycheff

$(\lambda_1, \lambda_2, \ldots, \lambda_k)$

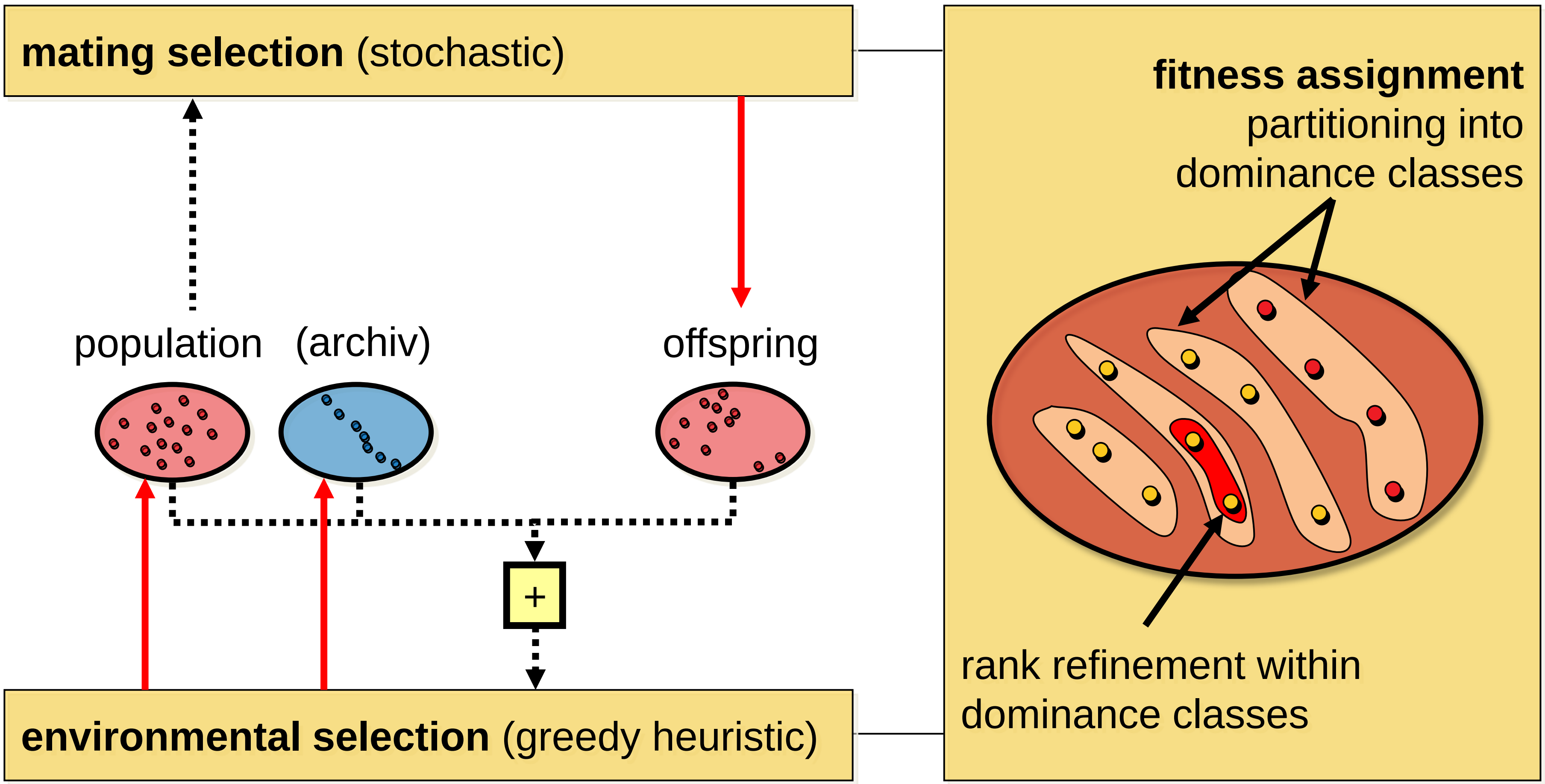
$$y = \max_i |\lambda_i(u_i - z_i)|$$

Several other scalarizing functions are known, see e.g. [Miettinen 1999]

Set-based Approaches
(Evolutionary Multiobjective Optimizers)


representation
fitness assignment
mating selection
parameters
0011
0111
0100
0011
0000
0011
1011
environmental selection
variation operators

General Scheme of Most Dominance-Based EMO

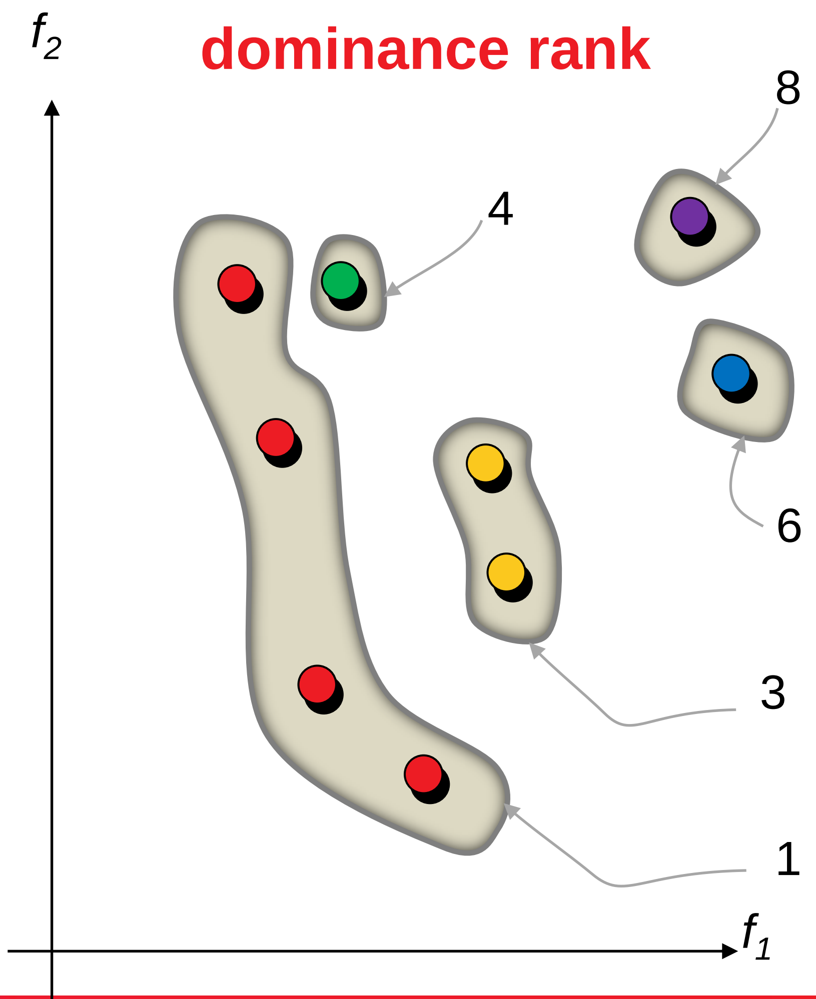


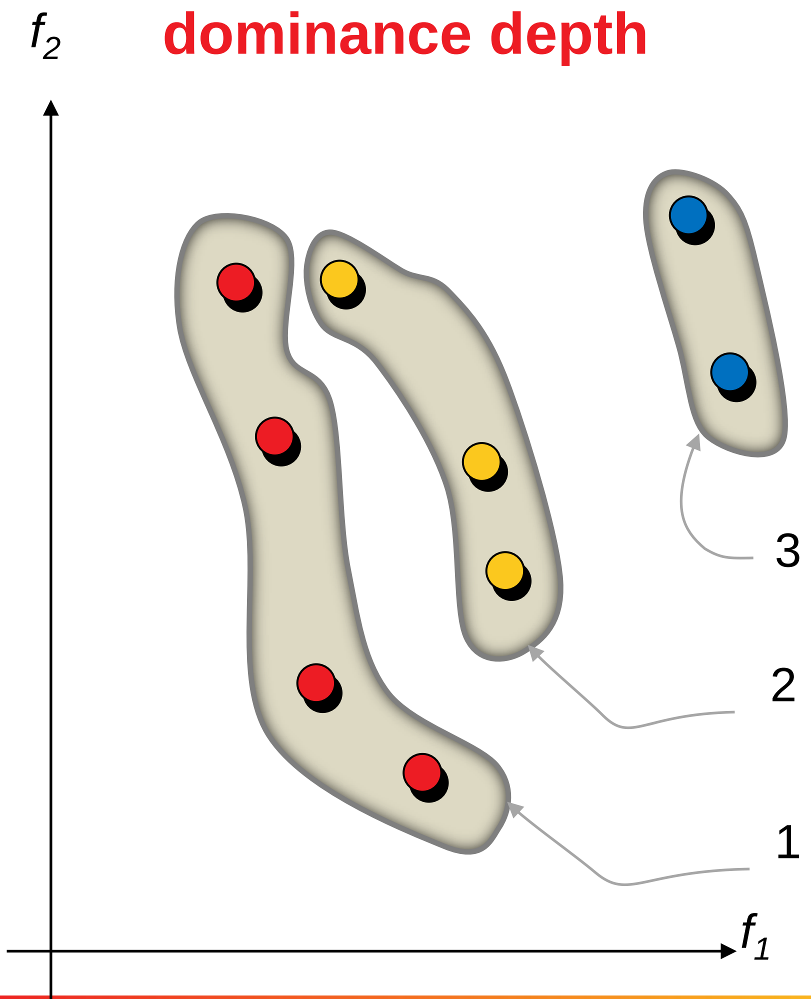
Ranking of the Population Using Dominance

... is based on pairwise comparisons of the individuals only

[Goldberg 1989]

Examples:





Refinement of Dominance Rankings

Goal: rank incomparable solutions within a dominance class

➊ Density information (good for search, but usually incompatible with Pareto-dominance)

➋ Quality indicator (good for set quality)

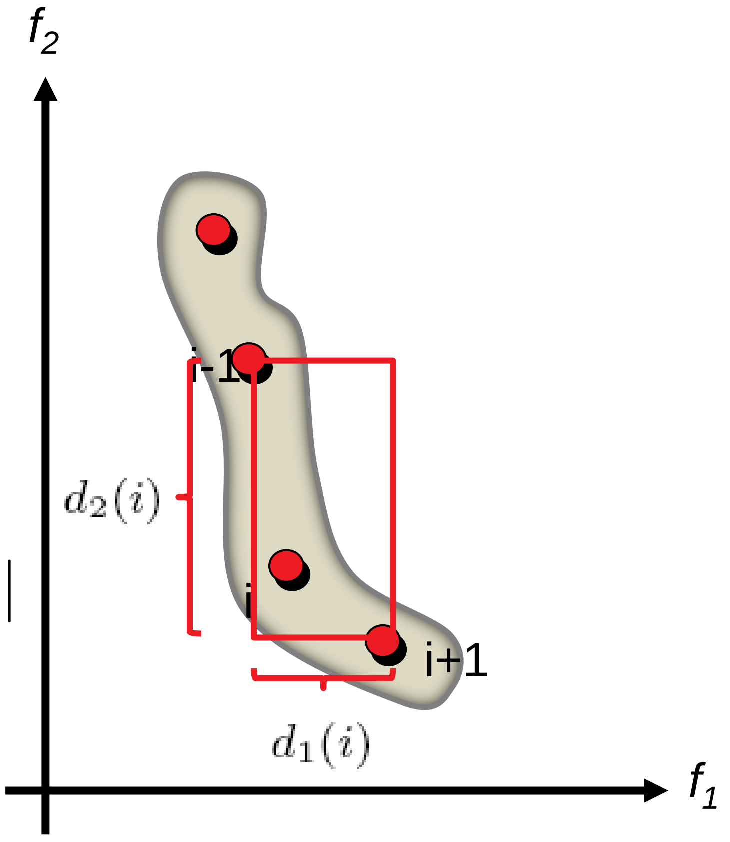
Density Estimation

crowding distance:

- sort solutions wrt. each objective

- crowding distance to neighbors:

$$d(i) = \sum_{\text{obj. } m} |f_m(i-1) - f_m(i+1)|$$

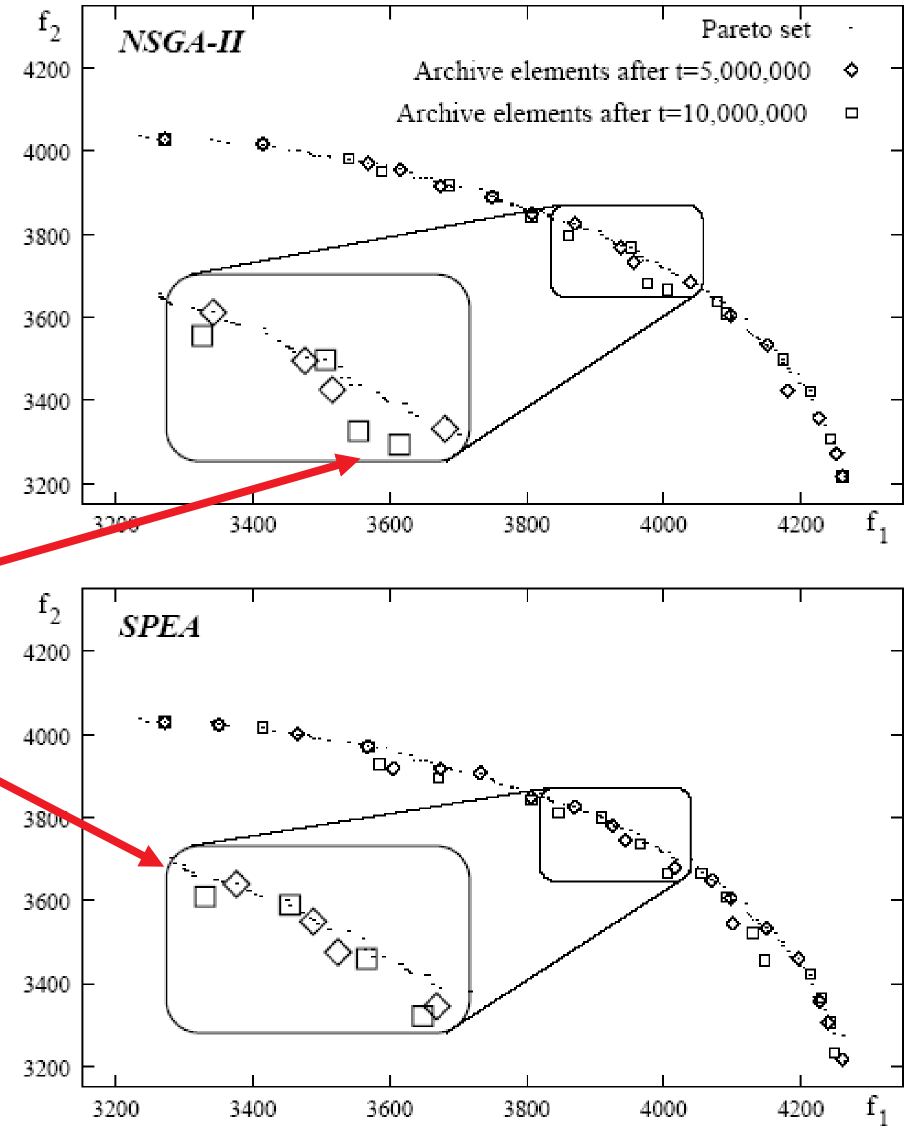


SPEA2 and NSGA-II: Cycles in Optimization

Selection in SPEA2 and NSGA-II can result in

deteriorative cycles

non-dominated solutions already found can be lost



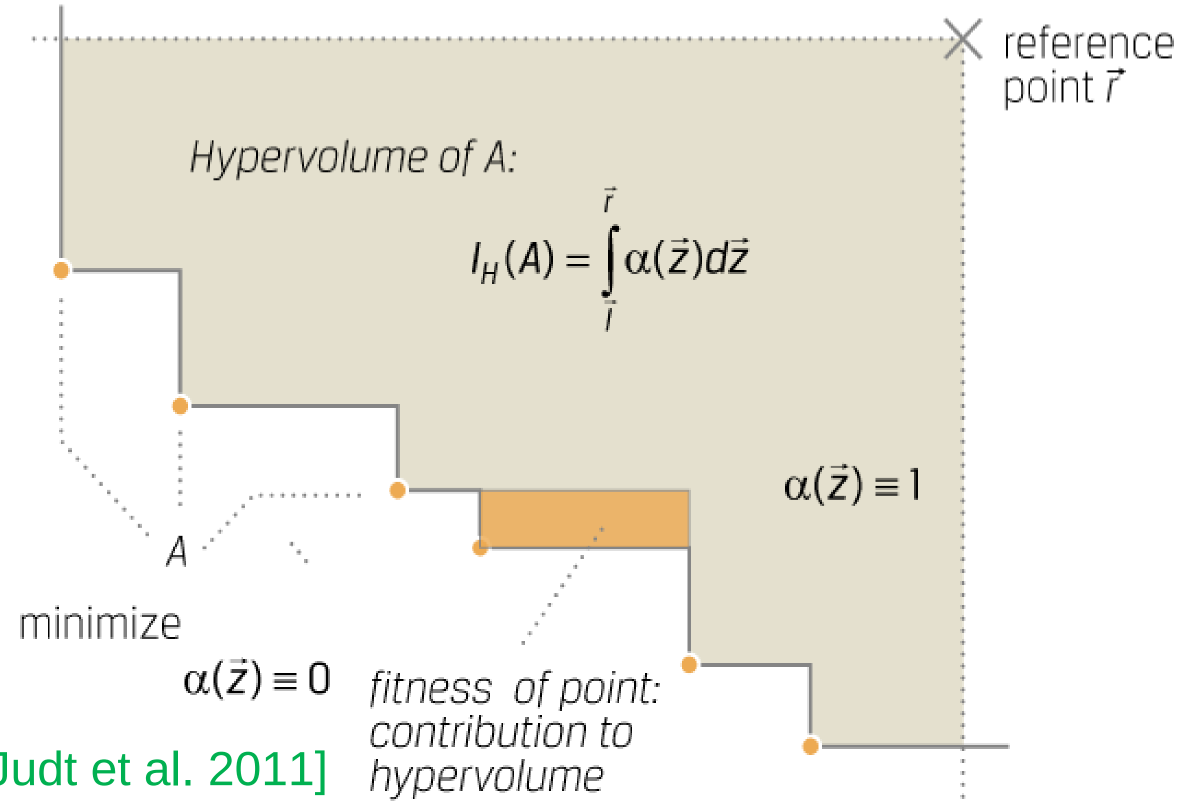
Hypervolume-Based Selection

Latest Approach (SMS-EMOA, MO-CMA-ES, HypE, …)

use hypervolume indicator to guide the search

Main idea

Delete solutions with the smallest hypervolume loss $d(s) = I_H(P) - I_H(P / \{s\})$ iteratively



But: can also result in cycles if reference point is not constant [Judt et al. 2011] and is expensive to compute exactly [Bringmann and Friedrich 2009]

Moreover: HypE [Bader and Zitzler 2011]

Sampling + Contribution if more than 1 solution deleted

Example Algorithm: the MO-CMA-ES

CMA-ES [remember talk of Rodolphe]

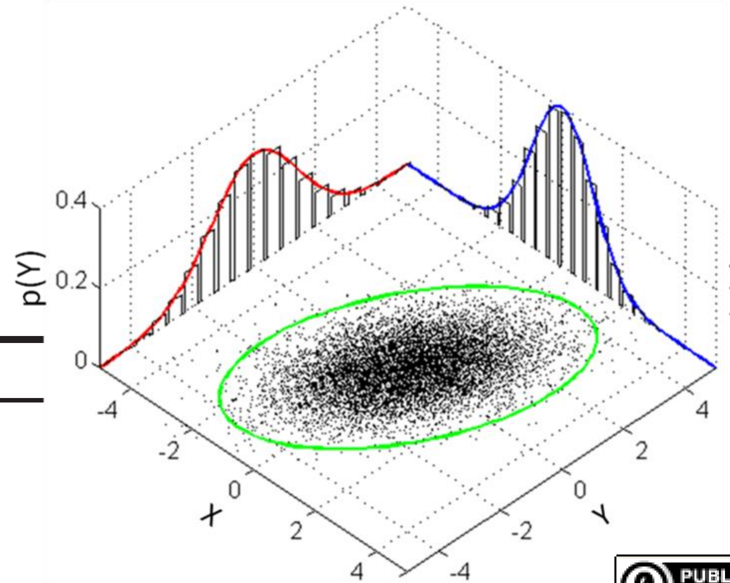
- Covariance Matrix Adaptation Evolution Strategy
  [Hansen and Ostermeier 1996, 2001]
- the state-of-the-art numerical black box optimizer for large budgets and difficult functions [Hansen et al. 2010]

CMA-ES for multiobjective optimization

- “THE” MO-CMA-ES does not exist
- original one of [Igel et al. 2007] in ECJ
- improved success definition [Voß et al. 2010] at GECCO 2010
- recombination between solutions [Voß et al. 2009] at EMO 2009
- all based on combination of μ single (1+1)-CMA-ES

(1+λ)-CMA-ES

$$a^{(g)} = \left(\mathbf{x}^{(g)}, \overline{p}_{succ}^{(g)}, \sigma^{(g)}, \mathbf{p}_c^{(g)}, \mathbf{C}^{(g)}\right)$$



Algorithm 1: (1+λ)-CMA-ES

$g = 0$, initialize $a_{\text{parent}}^{(g)}$
repeat
  $a_{\text{parent}}^{(g+1)} \leftarrow a_{\text{parent}}^{(g)}$
  for $k = 1, \ldots, \lambda$ **do**
    $x_k^{(g+1)} \sim \mathcal{N}\left(x_{\text{parent}}^{(g)}, {\sigma^{(g)}}^2 C^{(g)}\right)$
  `updateStepSize` $\left(a_{\text{parent}}^{(g+1)}, \frac{\lambda_{\text{succ}}^{(g+1)}}{\lambda}\right)$
  if $f\left(x_{1:\lambda}^{(g+1)}\right) \leq f\left(x_{\text{parent}}^{(g)}\right)$ **then**
    $x_{\text{parent}}^{(g+1)} \leftarrow x_{1:\lambda}^{(g+1)}$
    `updateCovariance` $\left(a_{\text{parent}}^{(g+1)}, \frac{x_{\text{parent}}^{(g+1)} - x_{\text{parent}}^{(g)}}{\sigma_{\text{parent}}^{(g)}}\right)$
  $g \leftarrow g + 1$
until stopping criterion is met

(1+λ)-CMA-ES: Updates

Procedure `updateStepSize(`$a = [\boldsymbol{x}, \overline{p}_{\text{succ}}, \sigma, \boldsymbol{p}_c, \boldsymbol{C}], p_{\text{succ}}$`)`

$$\overline{p}_{\text{succ}} \leftarrow (1 - c_p)\,\overline{p}_{\text{succ}} + c_p p_{\text{succ}}$$

$$\sigma \leftarrow \sigma \cdot \exp\left(\frac{1}{d}\,\frac{\overline{p}_{\text{succ}} - p_{\text{succ}}^{\text{target}}}{1 - p_{\text{succ}}^{\text{target}}}\right)$$

Procedure `updateCovariance(`$a = [\boldsymbol{x}, \overline{p}_{\text{succ}}, \sigma, \boldsymbol{p}_c, \boldsymbol{C}], \boldsymbol{x}_{\text{step}} \in \mathbb{R}^n$`)`

if $\overline{p}_{\text{succ}} < p_{\text{thresh}}$ **then**

$$\boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c + \sqrt{c_c(2 - c_c)}\;\boldsymbol{x}_{\text{step}}$$

$$\boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \boldsymbol{p}_c {\boldsymbol{p}_c}^{\mathrm{T}}$$

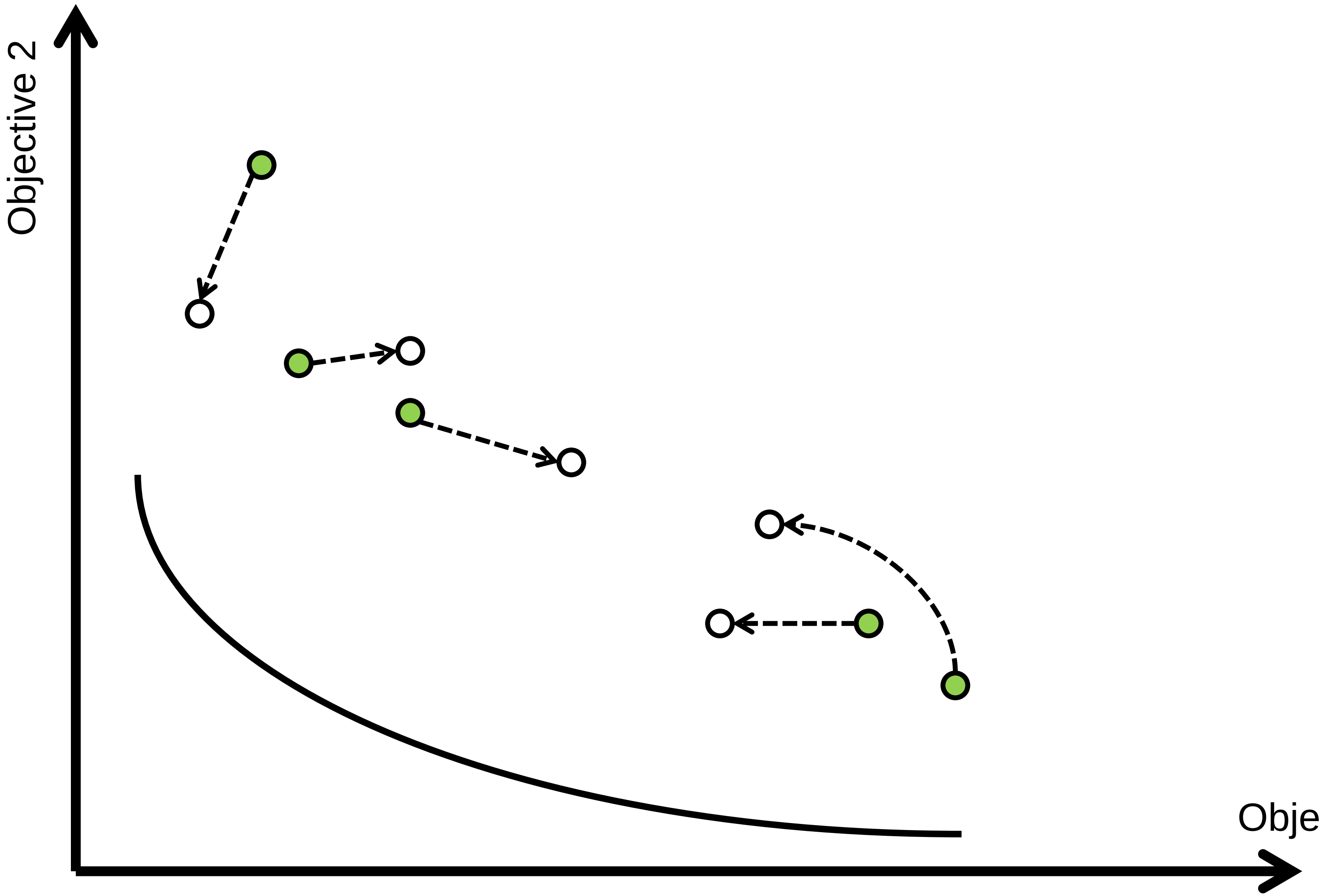
else

$$\boldsymbol{p}_c \leftarrow (1 - c_c)\boldsymbol{p}_c$$

$$\boldsymbol{C} \leftarrow (1 - c_{\text{cov}})\boldsymbol{C} + c_{\text{cov}} \cdot \left(\boldsymbol{p}_c {\boldsymbol{p}_c}^{\mathrm{T}} + c_c(2 - c_c)\boldsymbol{C}\right)$$

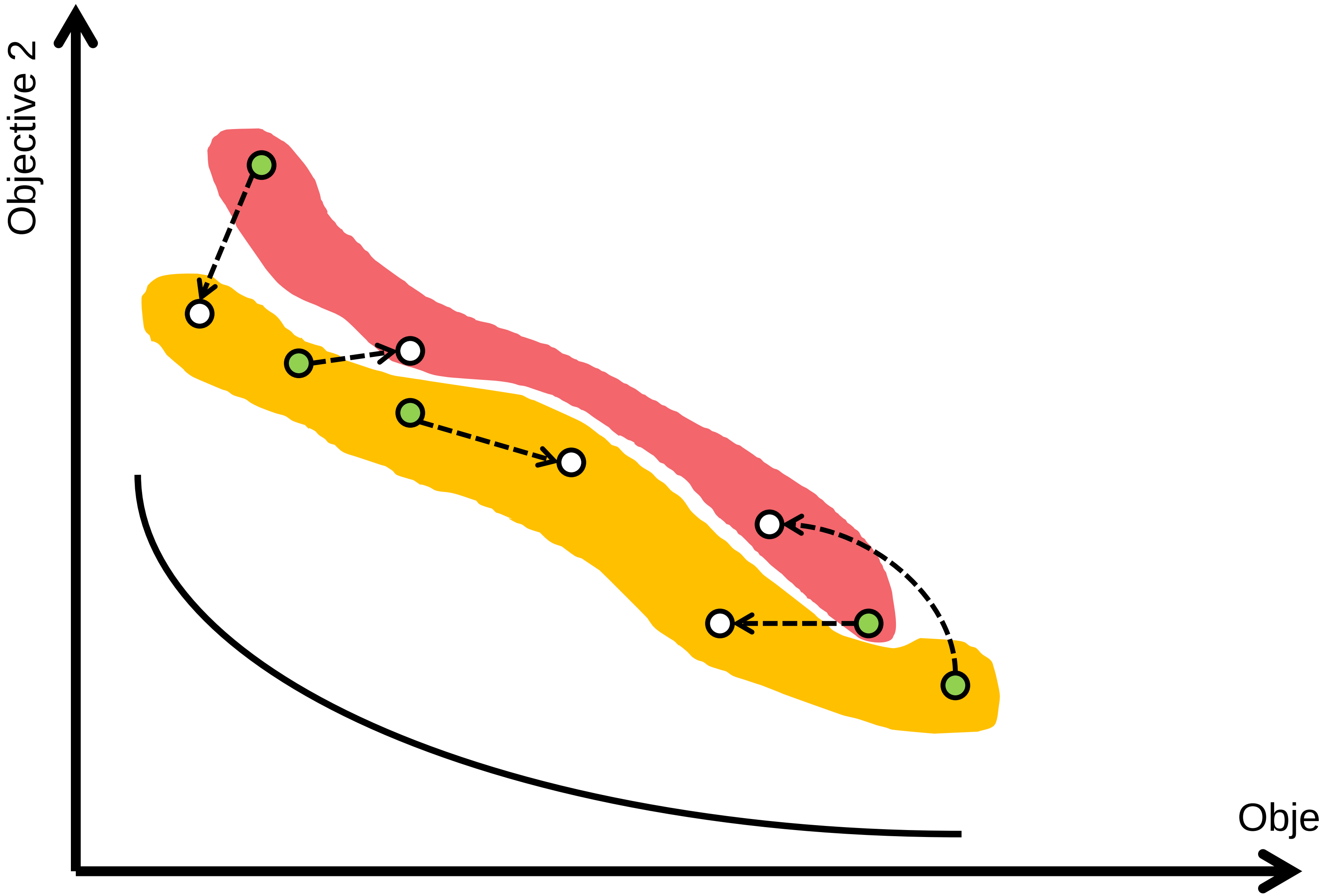
Concrete MO-CMA-ES Baseline Algorithm

μ x (1+1)-CMA-ES: $a_i^{(g)} = (\mathbf{x}_i^{(g)}, \overline{p}_{succ,i}^{(g)}, \sigma_i^{(g)}, \mathbf{p}_{c,i}^{(g)}, \mathbf{C}_i^{(g)})$



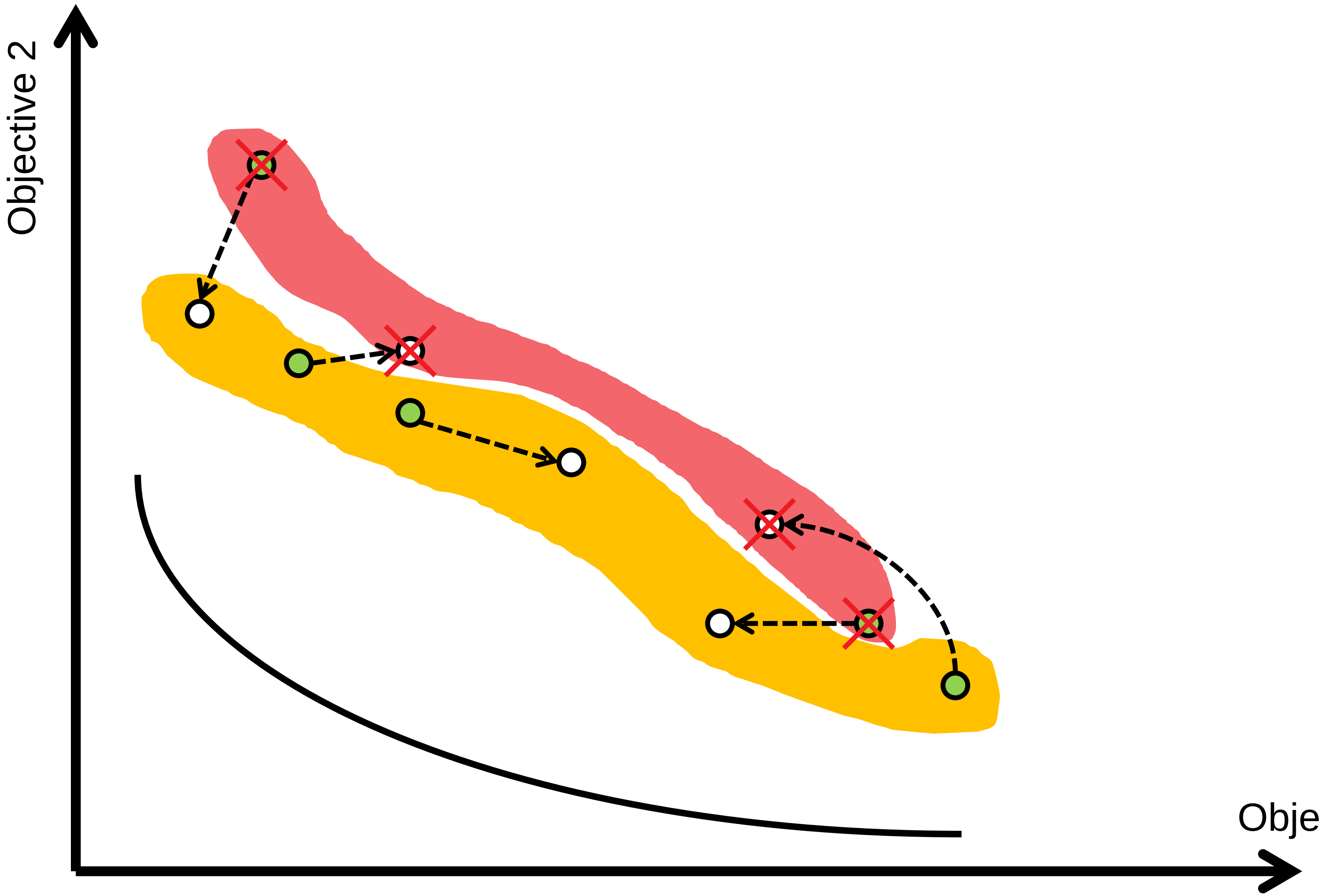
Concrete MO-CMA-ES Baseline Algorithm

μ x (1+1)-CMA-ES: $a_i^{(g)} = (\mathbf{x}_i^{(g)}, \overline{p}_{succ,i}^{(g)}, \sigma_i^{(g)}, \mathbf{p}_{c,i}^{(g)}, \mathbf{C}_i^{(g)})$



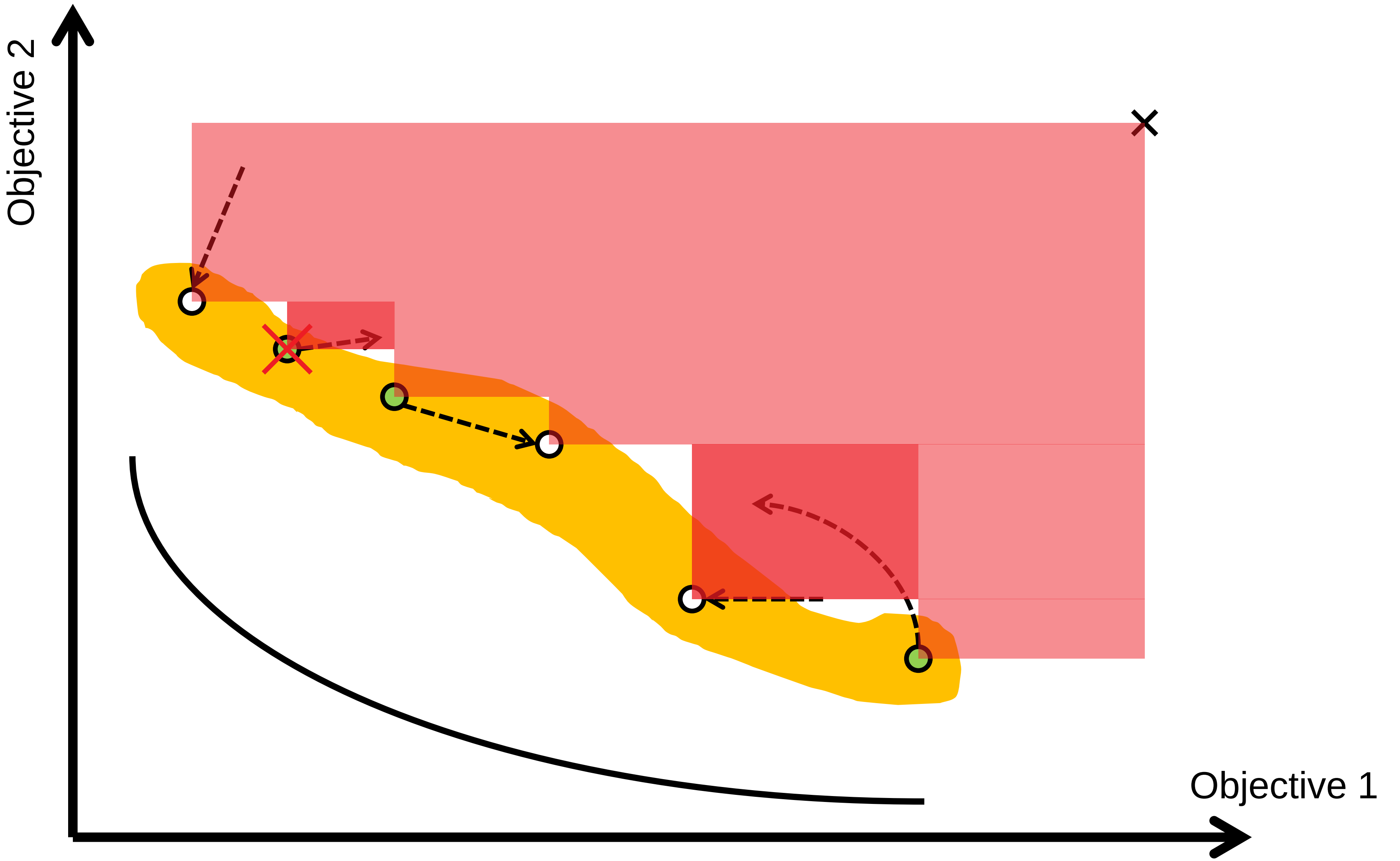
Concrete MO-CMA-ES Baseline Algorithm

μ x (1+1)-CMA-ES: $a_i^{(g)} = (\mathbf{x}_i^{(g)}, \overline{p}_{succ,i}^{(g)}, \sigma_i^{(g)}, \mathbf{p}_{c,i}^{(g)}, \mathbf{C}_i^{(g)})$



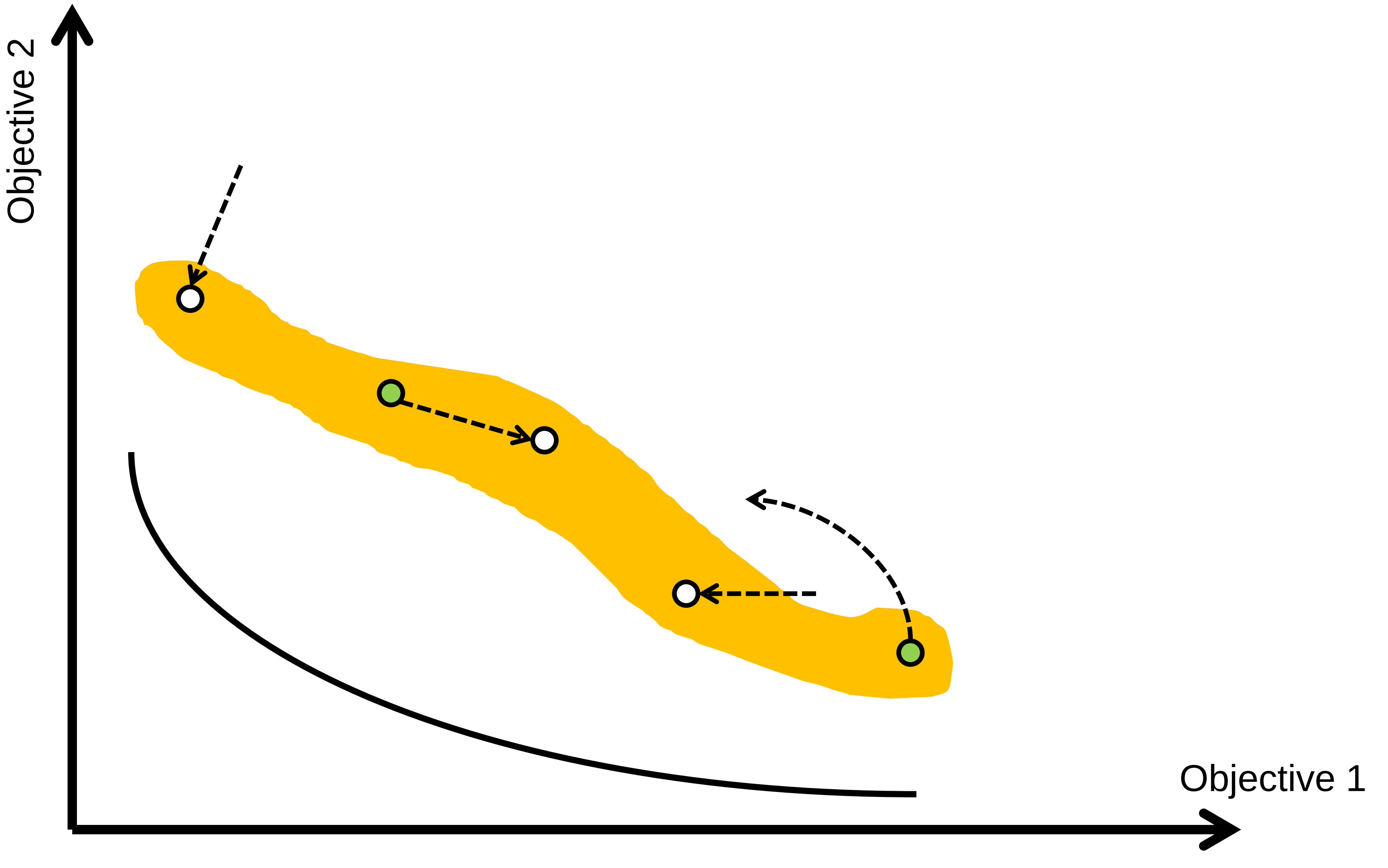
Concrete MO-CMA-ES Baseline Algorithm

μ x (1+1)-CMA-ES: $$a_i^{(g)} = (\mathbf{x}_i^{(g)}, \overline{p}_{succ,i}^{(g)}, \sigma_i^{(g)}, \mathbf{p}_{c,i}^{(g)}, \mathbf{C}_i^{(g)})$$



Concrete MO-CMA-ES Baseline Algorithm

μ x (1+1)-CMA-ES: $a_i^{(g)} = (\mathbf{x}_i^{(g)}, \overline{p}_{succ,i}^{(g)}, \sigma_i^{(g)}, \mathbf{p}_{c,i}^{(g)}, \mathbf{C}_i^{(g)})$



MO-CMA-ES baseline algorithm

- μ x (1+1)-CMA-ES
- hypervolume-based selection
- update of CMA strategy parameters based on different success notions

Success Definitions:

- original success [Igel et al. 2007]: if offspring dominates parent
- improved success [Voß et al. 2010]: if offspring selected into new population

Available Implementations:

- Baseline in Shark machine learning library (C++)
  - http://image.diku.dk/shark/
- Now also available in MATLAB
  - easy prototyping of new ideas
  - visualization of algorithm's state variables (similar to CMA-ES)

Overview

The Big Picture

Basic Principles of Multiobjective Optimization

- algorithm design principles and concepts
- performance assessment & benchmarking

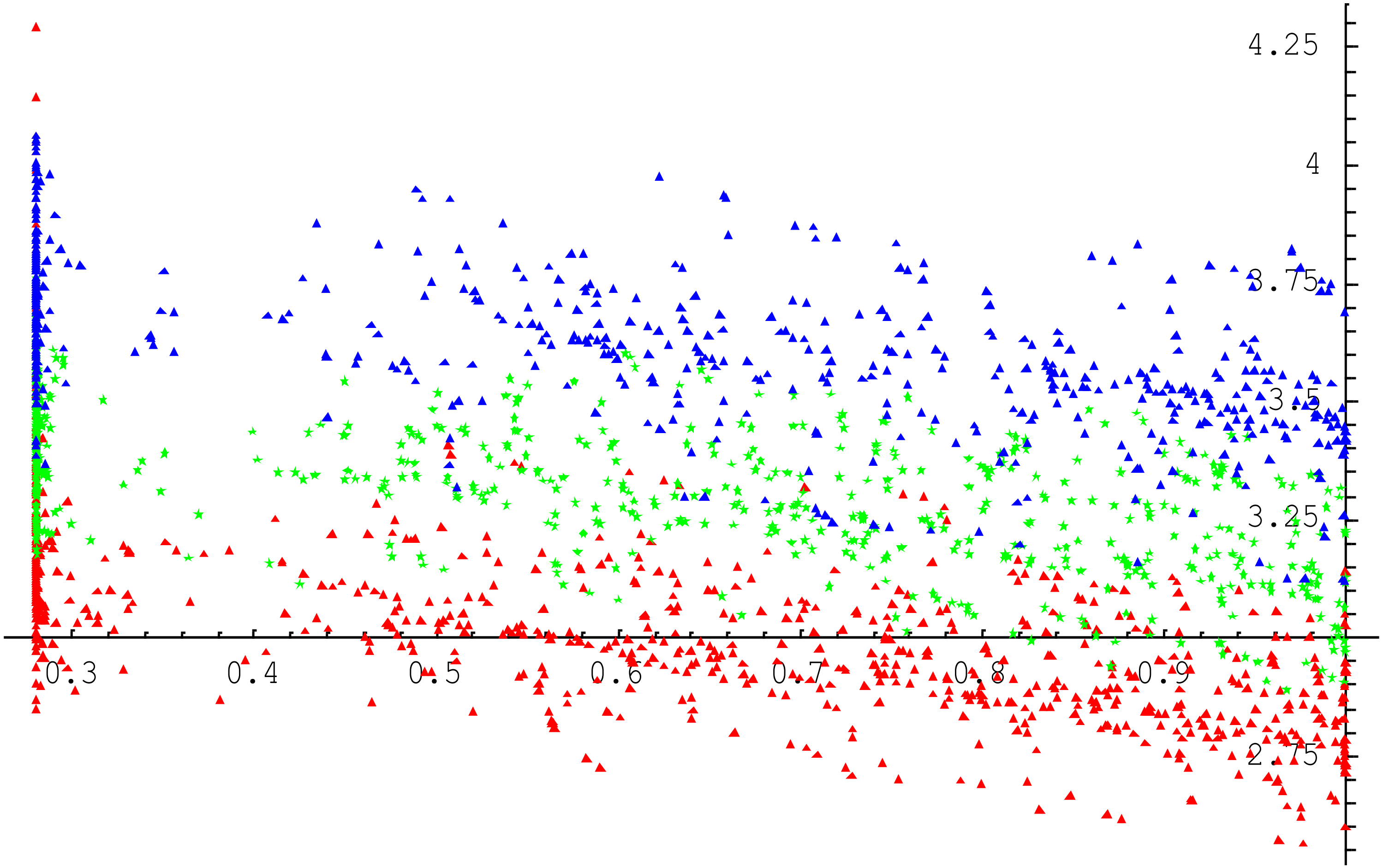
Why Benchmarking Algorithms?

- Understanding of algorithms
- Algorithm selection
- Putting algorithms to a standardized test
  - simplify judgement
  - simplify comparison
  - regression test under algorithm changes

We can measure performance on

- real world problems
  - expensive, often limited to certain domain
- "artificial" benchmark functions
  - cheap
  - controlled
  - data acquisition is comparatively easy
  - problem of representativity

... multiobjective EAs were mainly compared visually:



ZDT6 benchmark problem: IBEA, SPEA2, NSGA-II

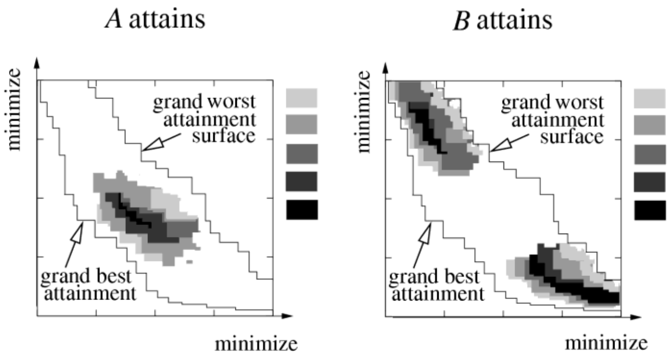
Two Approaches for Empirical Studies

Attainment function approach:

- Applies statistical tests directly to the samples of approximation sets
- Gives detailed information about how and where performance differences occur

Quality indicator approach:

- First, reduces each approximation set to a single value of quality
- Applies statistical tests to the samples of quality values

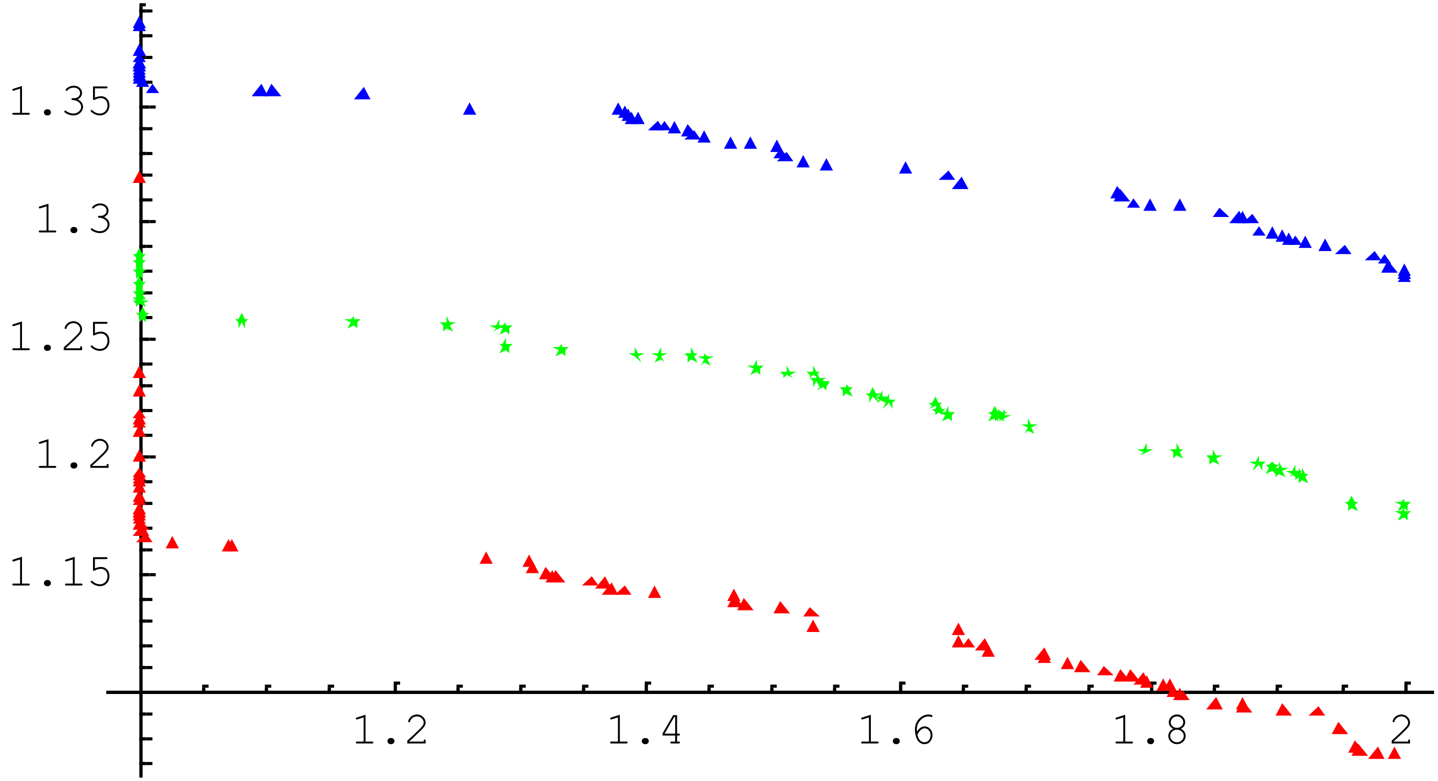


Indicator	A	B
Hypervolume indicator	6.3431	7.1924
ϵ-indicator	1.2090	0.12722
R_2 indicator	0.2434	0.1643
R_3 indicator	0.6454	0.3475

see e.g. [Zitzler et al. 2003]

Attainment Plots

50% attainment surface for IBEA, SPEA2, NSGA2 (ZDT6)



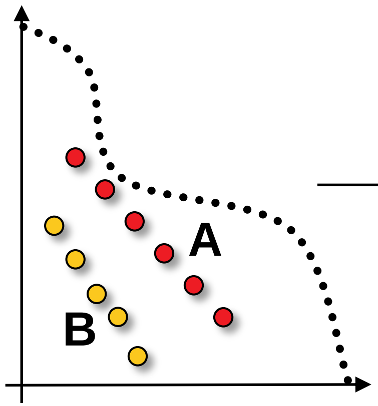
latest implementation online at

`http://eden.dei.uc.pt/~cmfonsec/software.html`

see [Fonseca et al. 2011]

Quality Indicator Approach

Goal: compare two Pareto set approximations A and B



	A	B
hypervolume	*432.34*	*420.13*
distance	*0.3308*	*0.4532*
diversity	*0.3637*	*0.3463*
spread	*0.3622*	*0.3601*
cardinality	*6*	*5*

→ "**A** better"

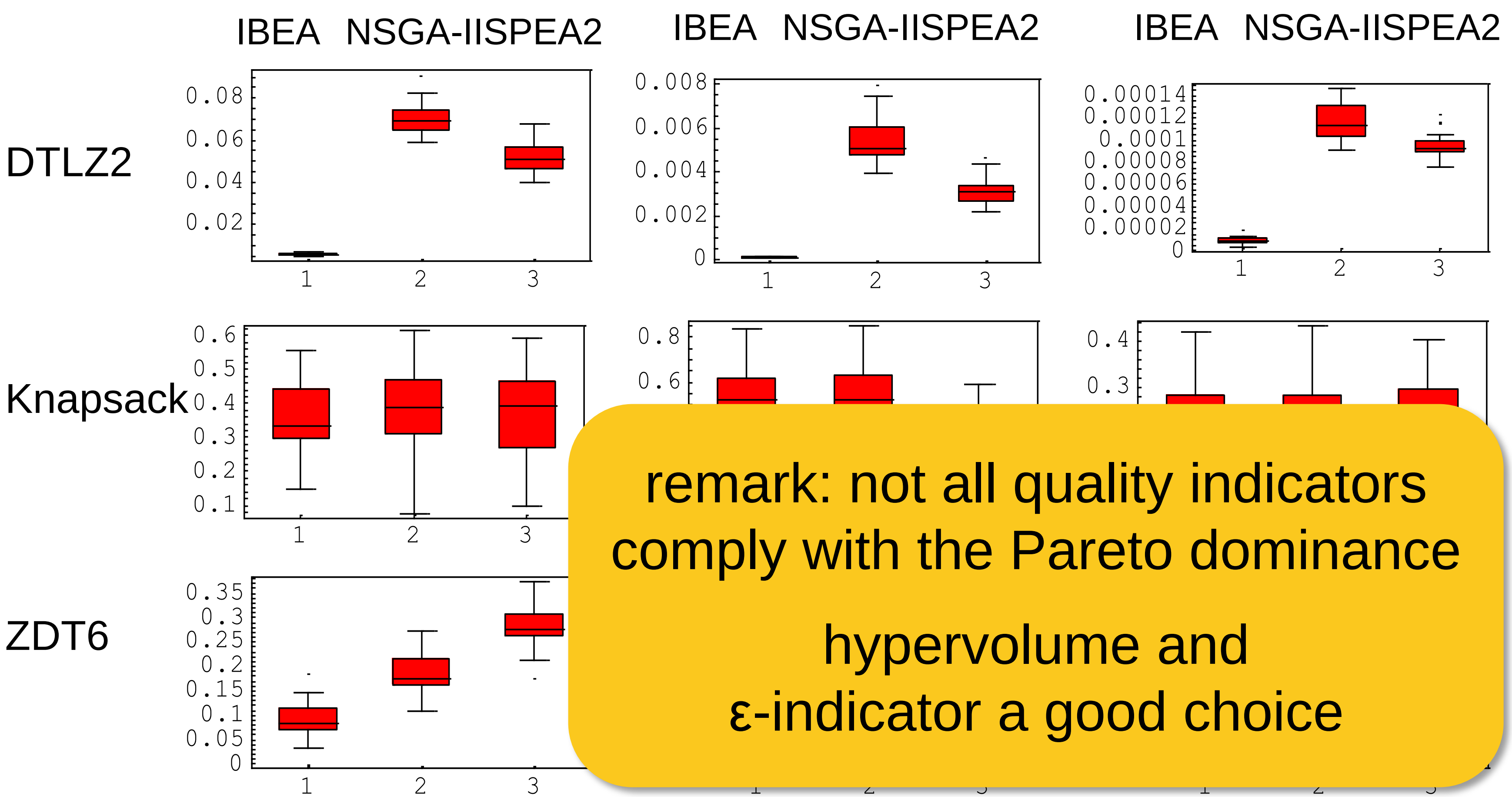
Comparison method C = quality measure(s) + Boolean function

$$A, B \xrightarrow{\text{quality measure}} \mathbb{R}^n \xrightarrow{\text{Boolean function}} \text{statement}$$

reduction **interpretation**

Example: Box Plots

epsilon indicator | hypervolume | R indicator



remark: not all quality indicators comply with the Pareto dominance

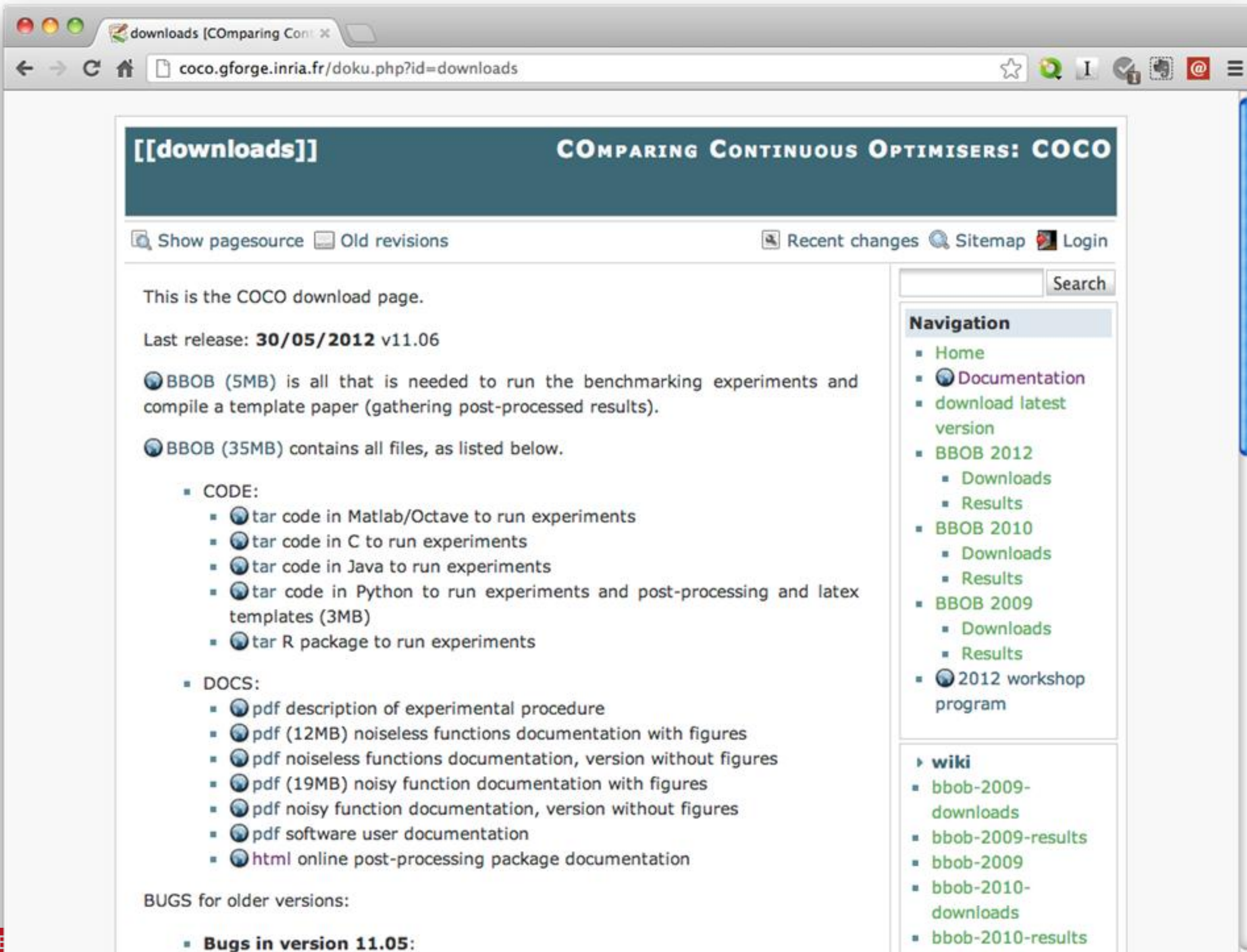
hypervolume and
ε-indicator a good choice

small excursion:

Single-objective Blackbox Optimization Benchmarking in Practice (using COCO)

COCO (COmparing Continuous Optimizers): a tool for black-box optimization benchmarking

[slides borrowed from Nikolaus Hansen]



downloads [COmparing Con

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

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¬
```

Interface: MY_OPTIMIZER(function_name, dimension, optional_args)

Running the experiment at an OS shell:

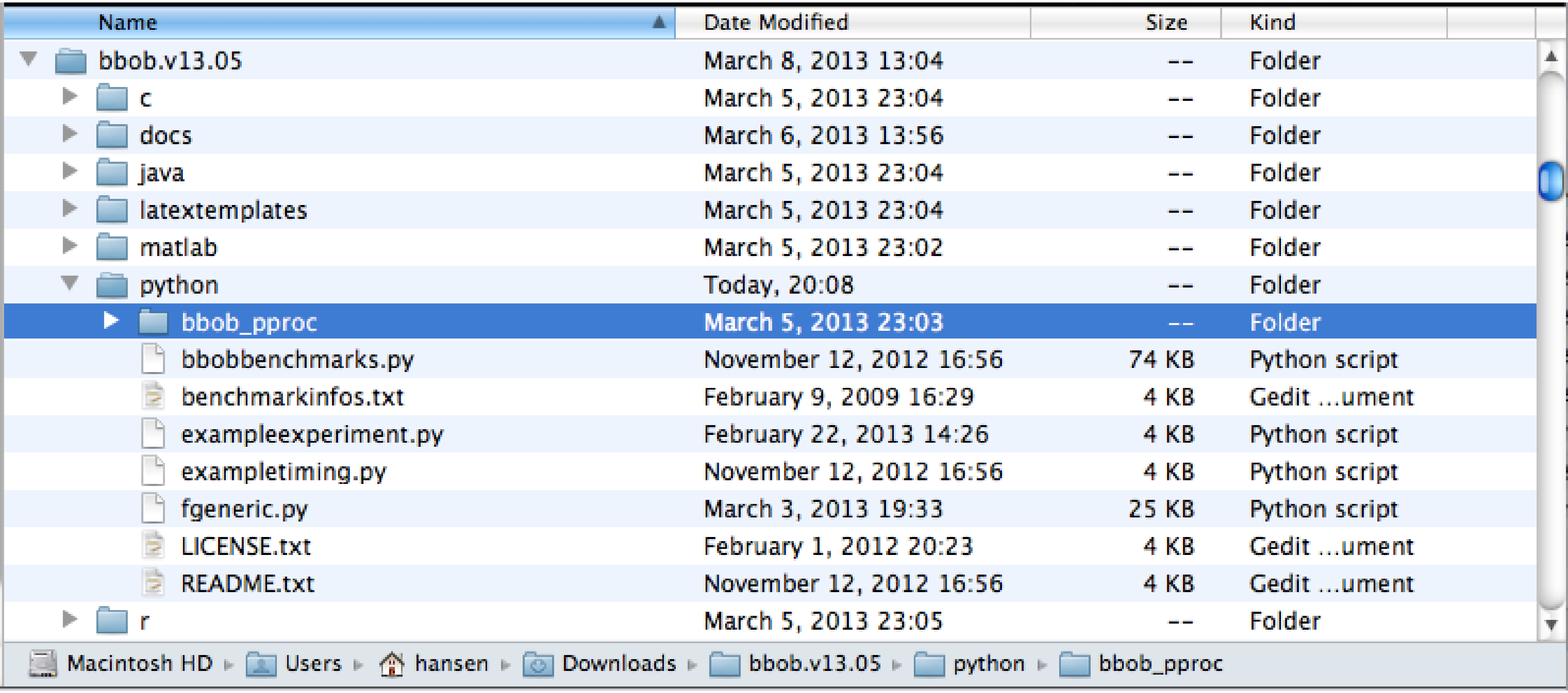
```
$ nohup nice octave < exampleexperiment.m > output.txt &
$ less output.txt
```

```
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



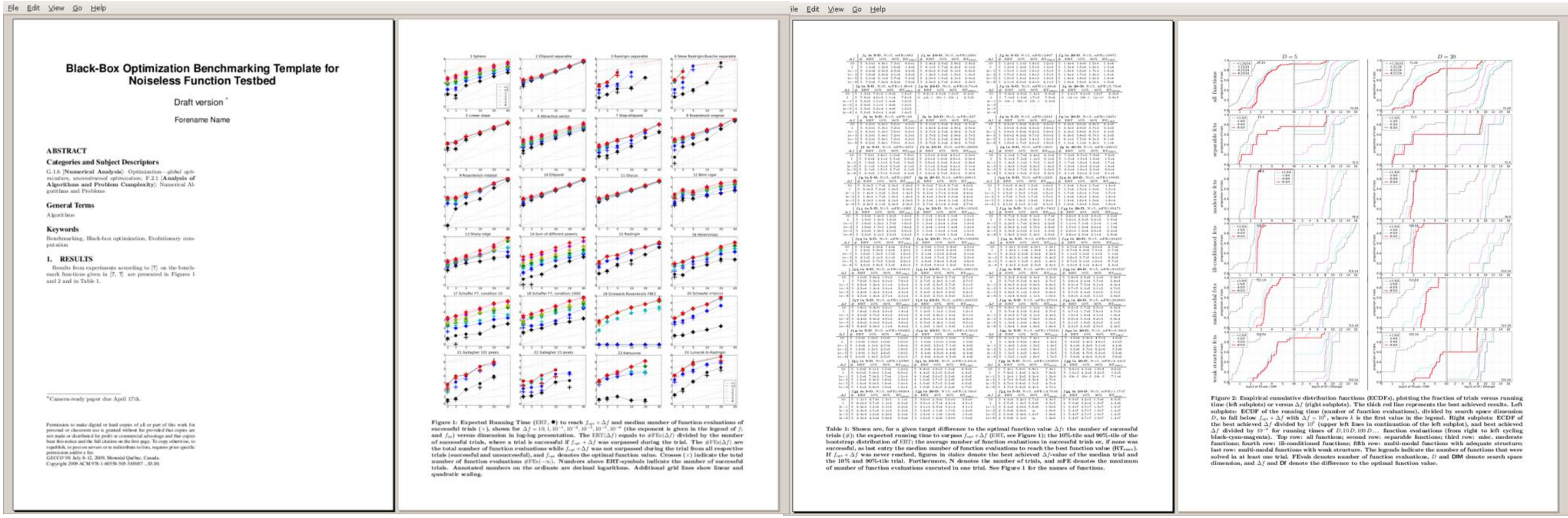
Post-processing at the OS shell:

```
$ python codepath/bbob_pproc/rungeneric.py datapath

[...]

$ pdflatex templateACMarticle.tex

[...]
```



LaTeX Templates

File 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.

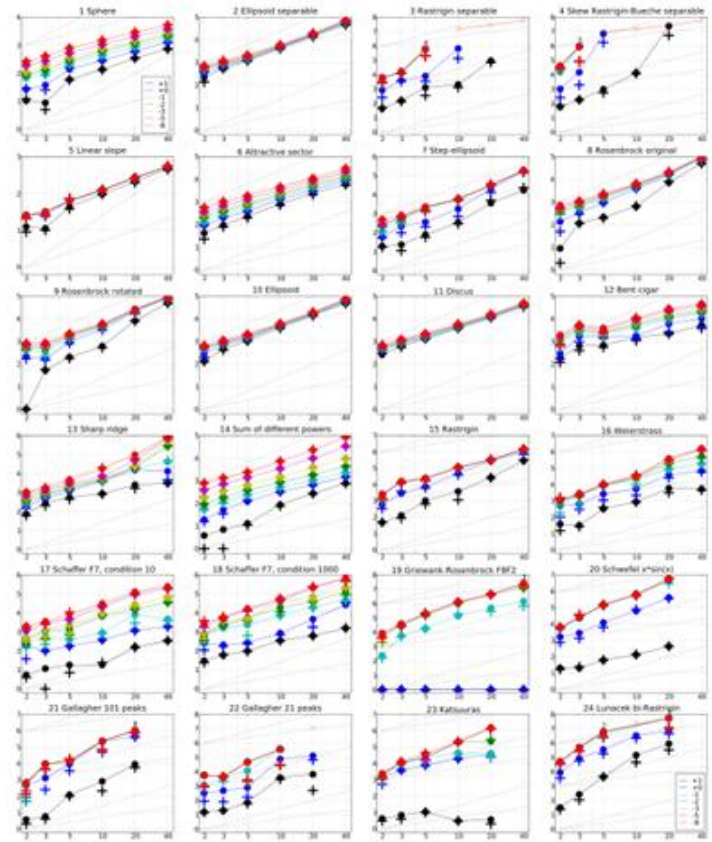
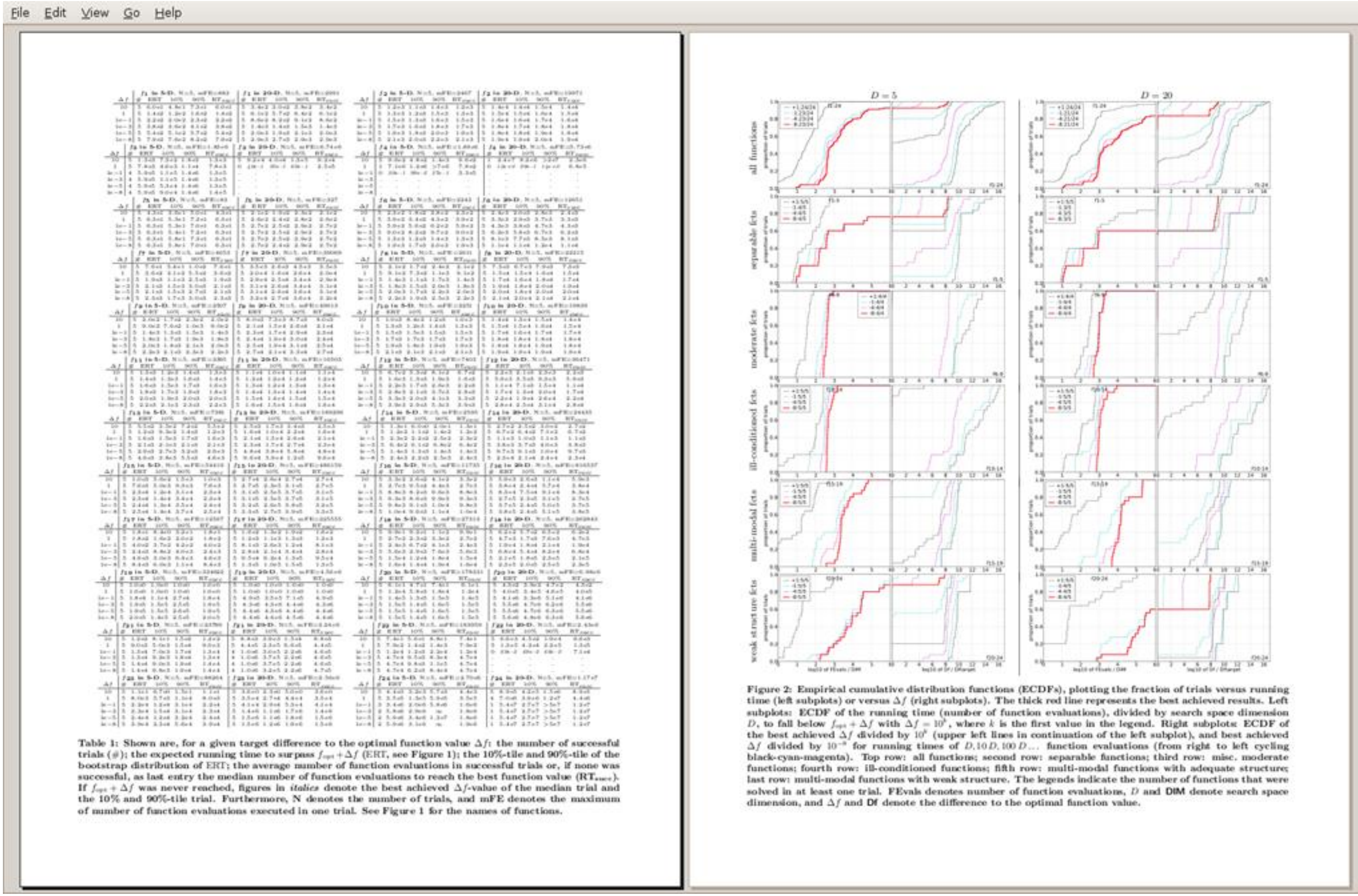


Figure 1: Expected Running Time (ERT, ●) to reach $f_{opt} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation.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

LaTeX Templates



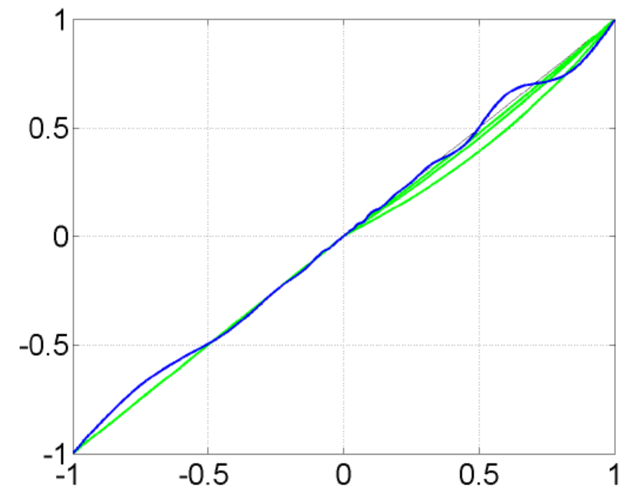
File Edit View Go Help

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 $f_{opt} + \Delta f$ (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 (RT_{succ}). If $f_{opt} + \Delta f$ 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 $f_{opt} + \Delta f$ 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.

The BBOB (Noiseless) Test Functions

- all define a "scientific question”
  the relevance can hardly be overestimated
- functions are not perfectly symmetric and are locally deformed
- 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

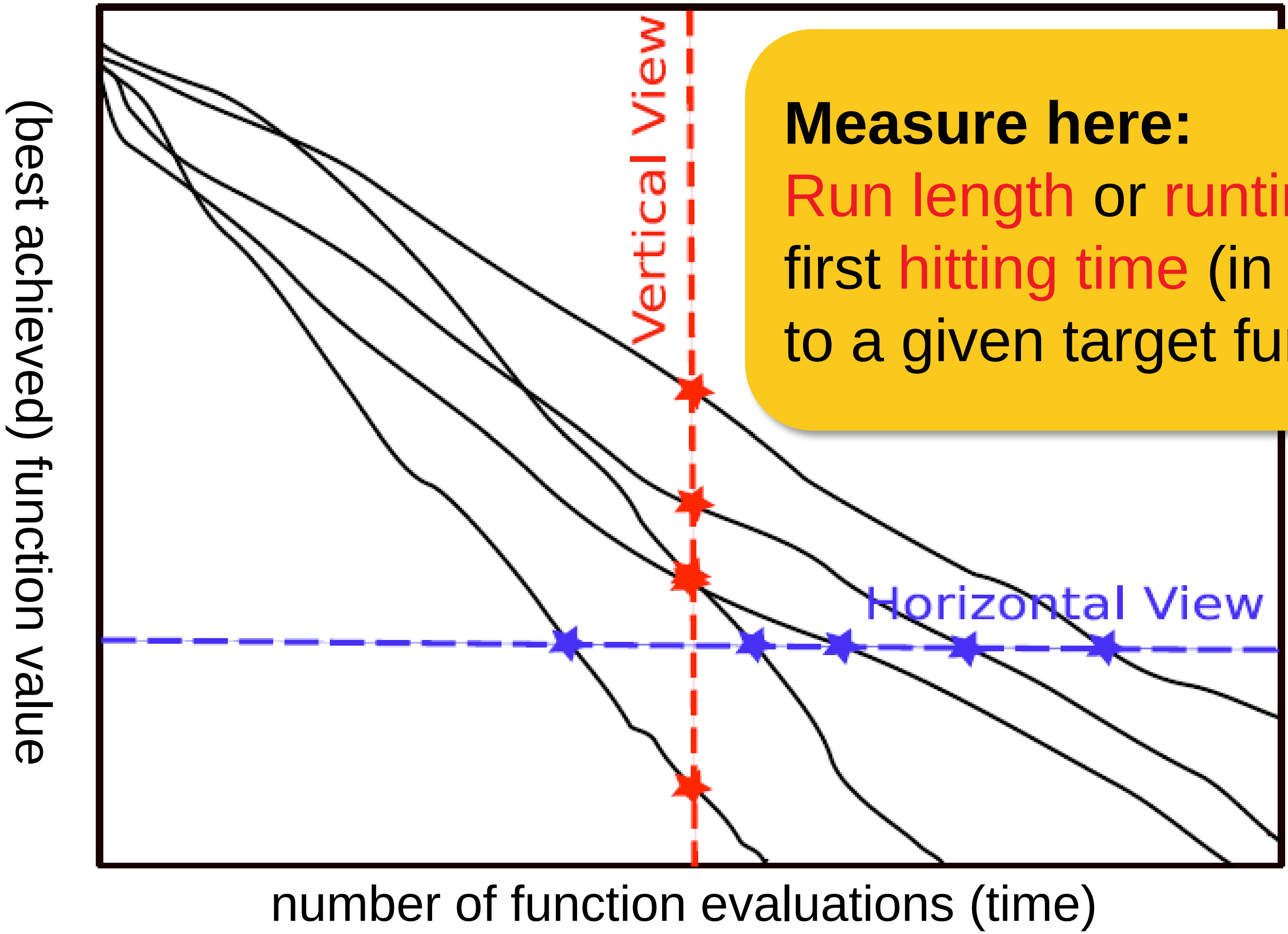


How do we measure performance?

Measuring Performance

...empirically...

convergence graphs is all we have to start with



Measure here:
Run length or runtime or
first hitting time (in #funevals)
to a given target function value

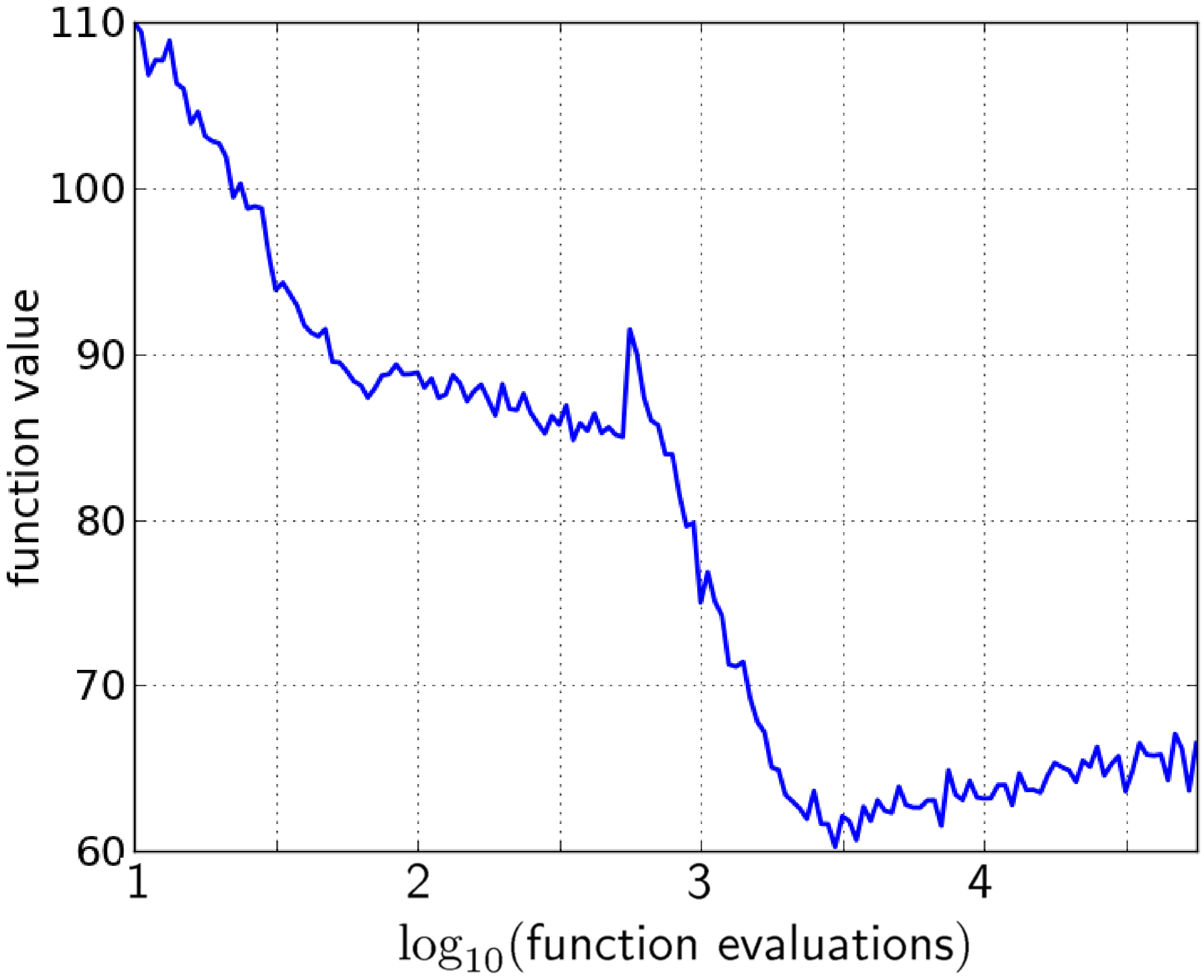
ECDF:

Empirical Cumulative Distribution Function of the Runtime

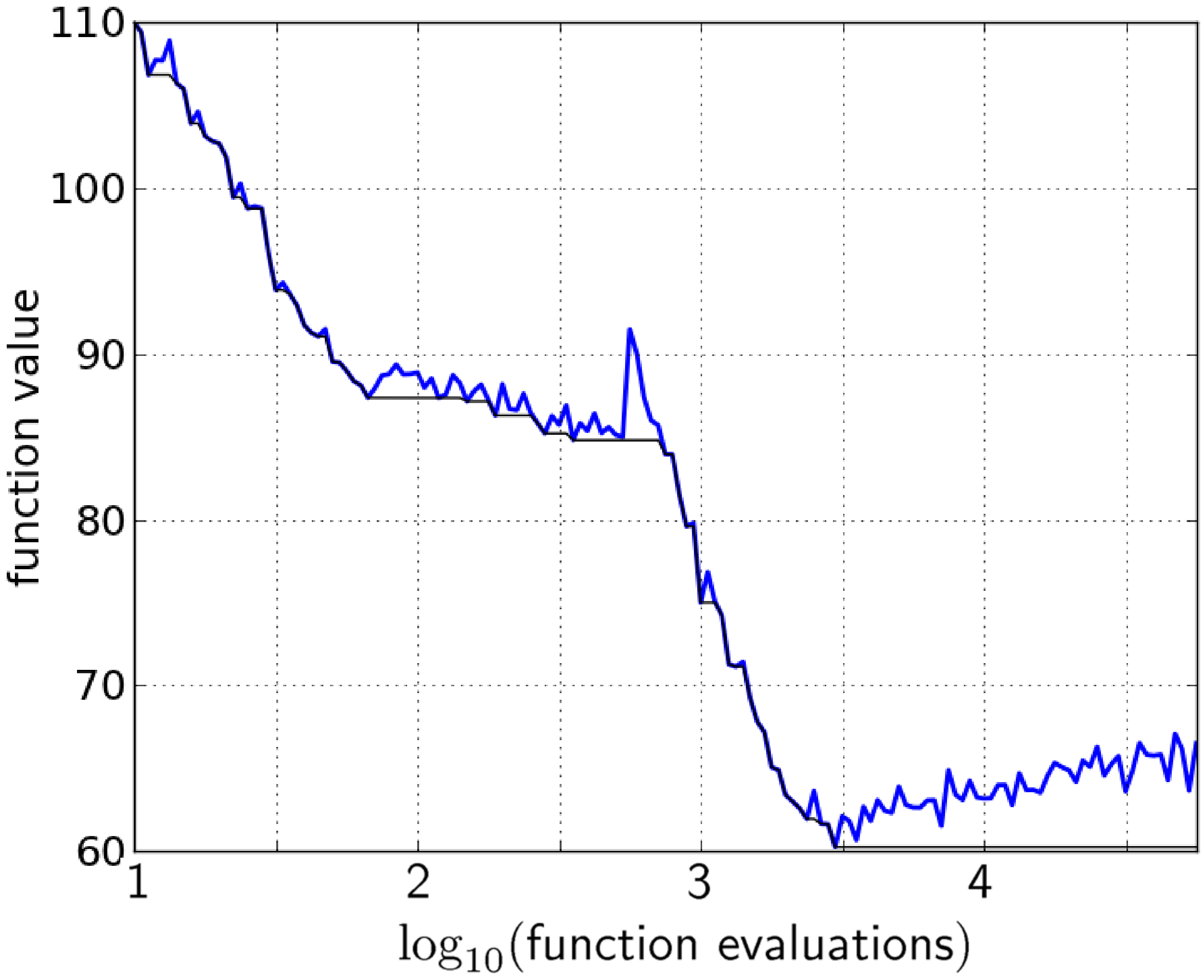
aka

Data Profiles

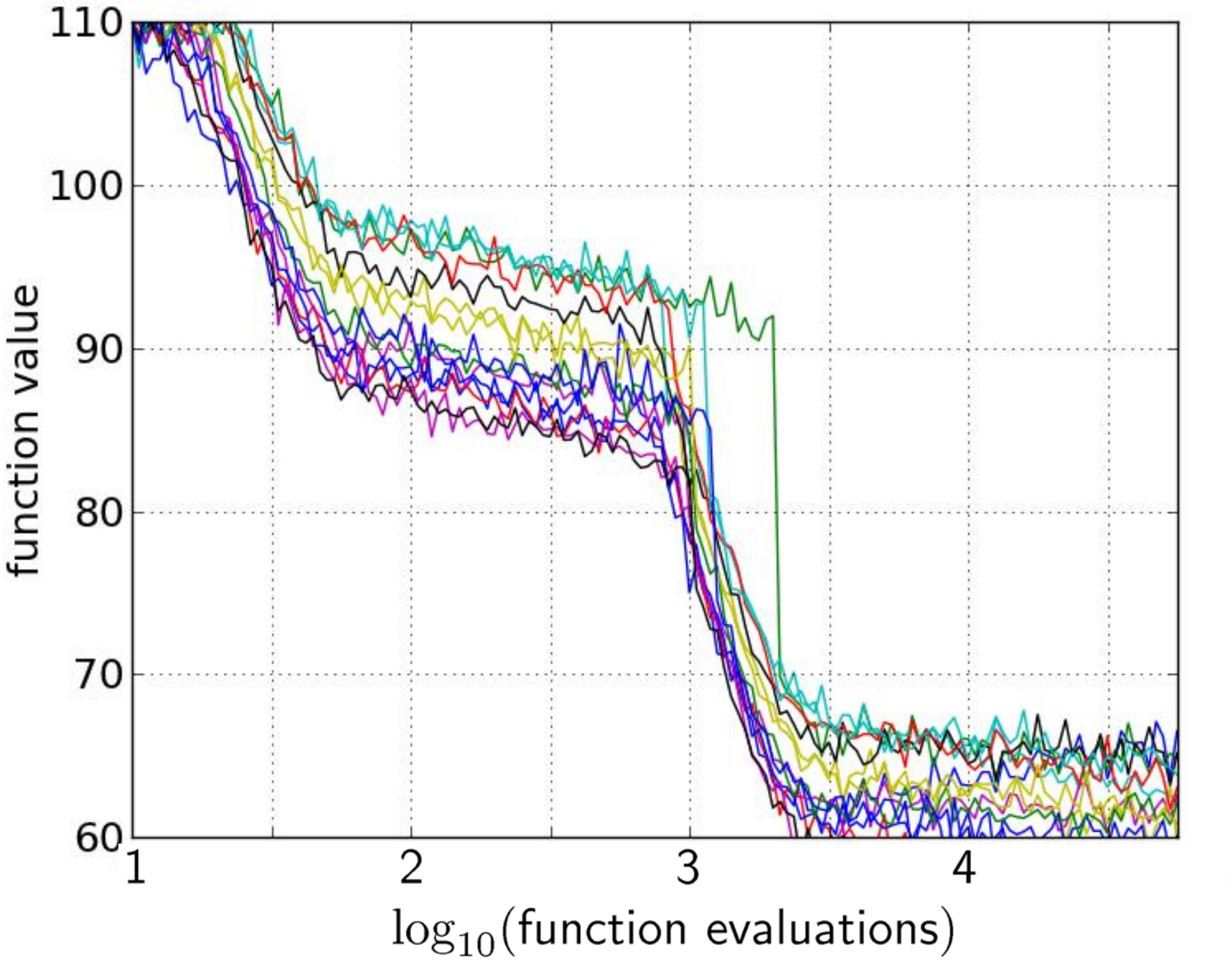
A Convergence Graph



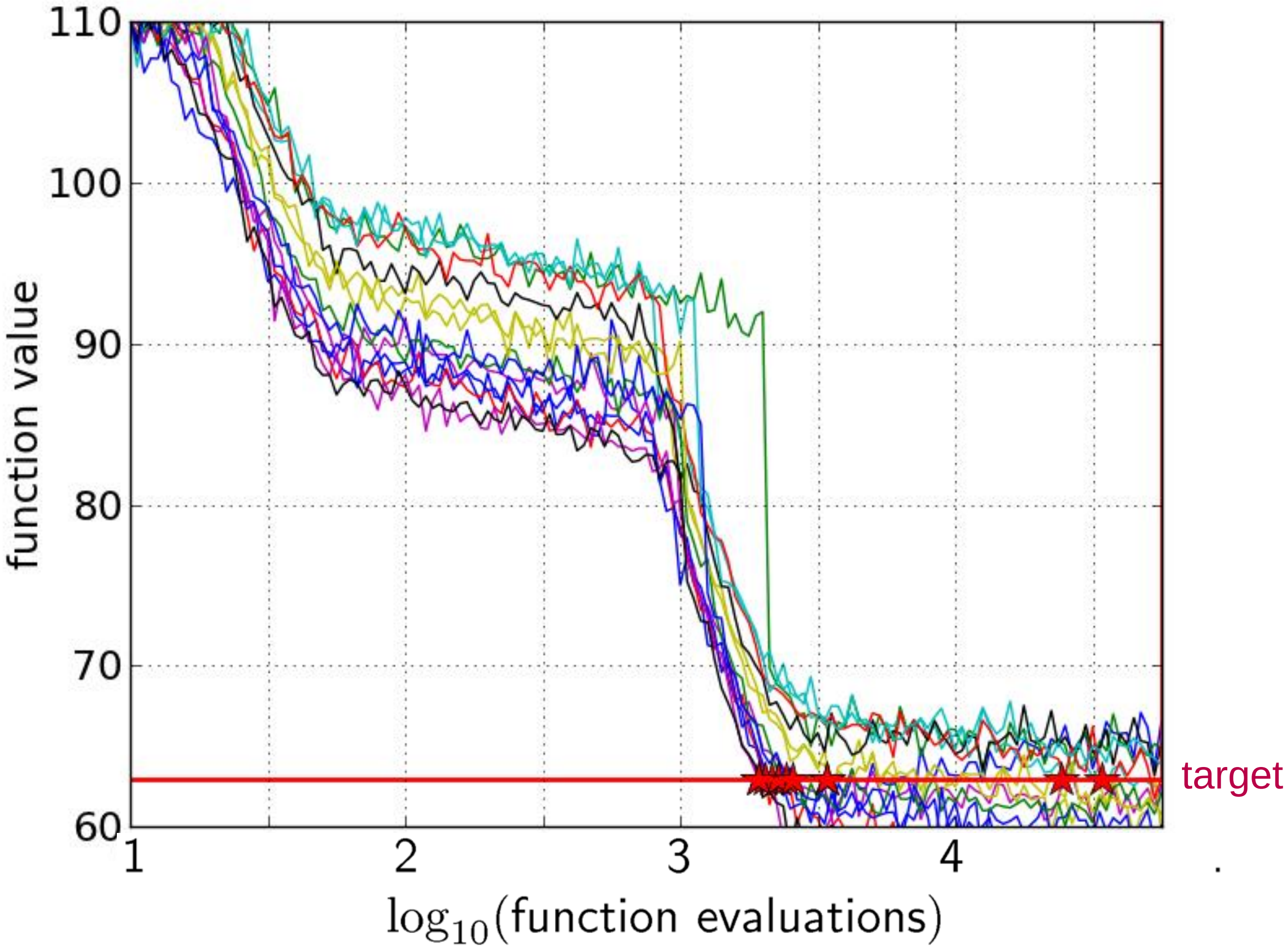
First Hitting Time is Monotonous



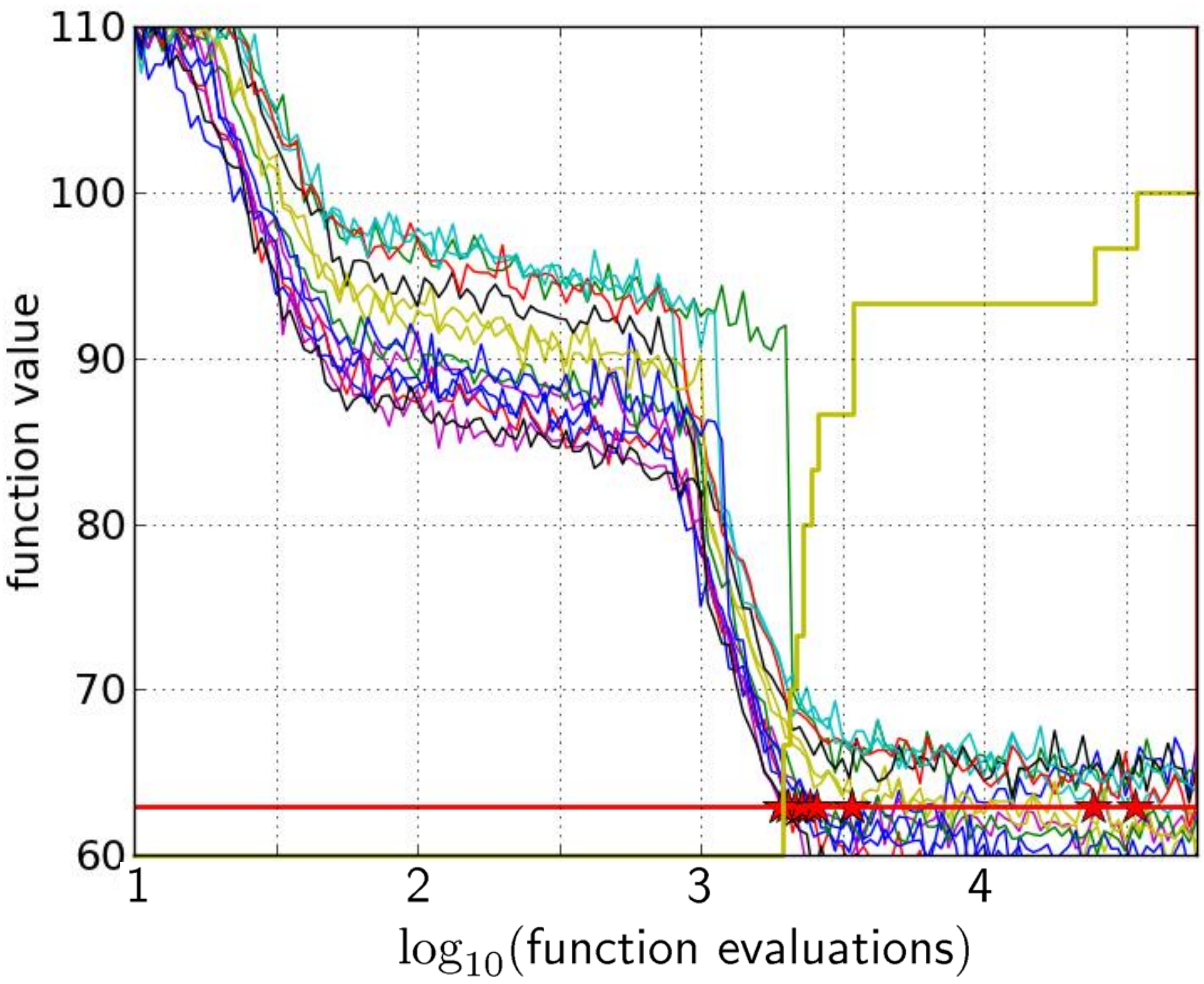
15 Algorithm Runs



15 Algorithm Runs



Empirical CDF



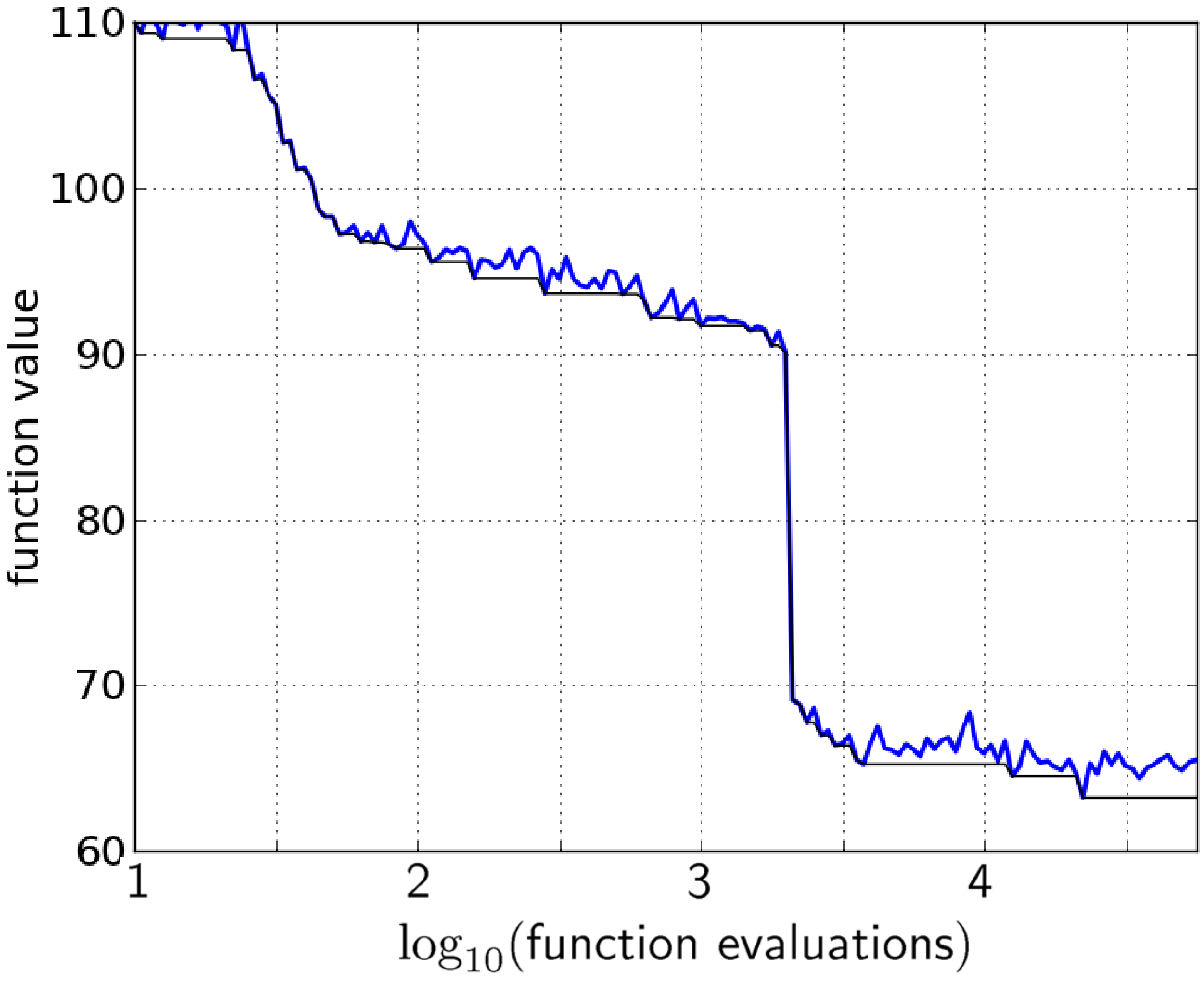
the **ECDF** of run lengths (runtimes) to reach the target

- 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)

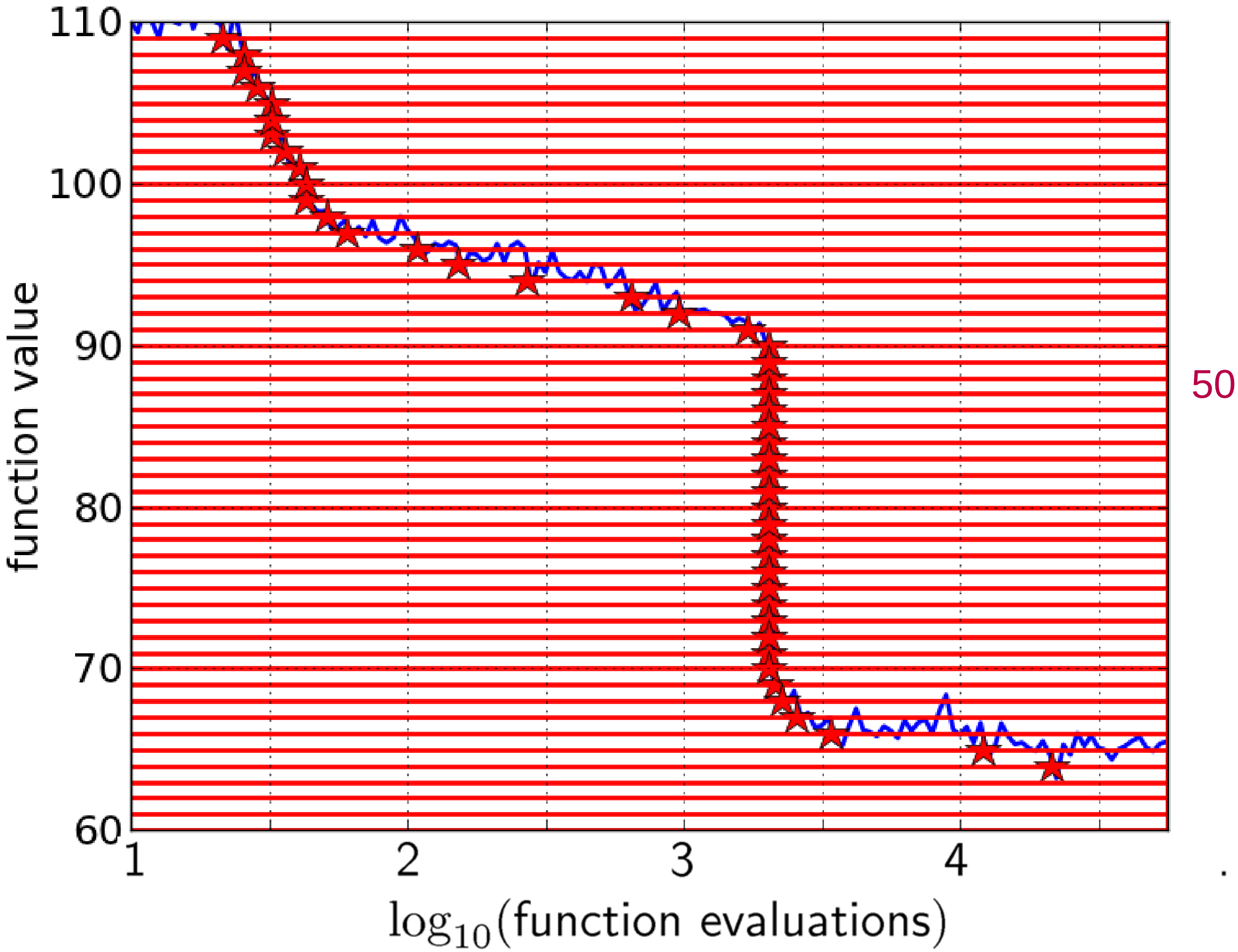
60% of the runs need between 2000 and 4000 evaluations

80% of the runs reached the target

Reconstructing Single Runs: Recording More Targets

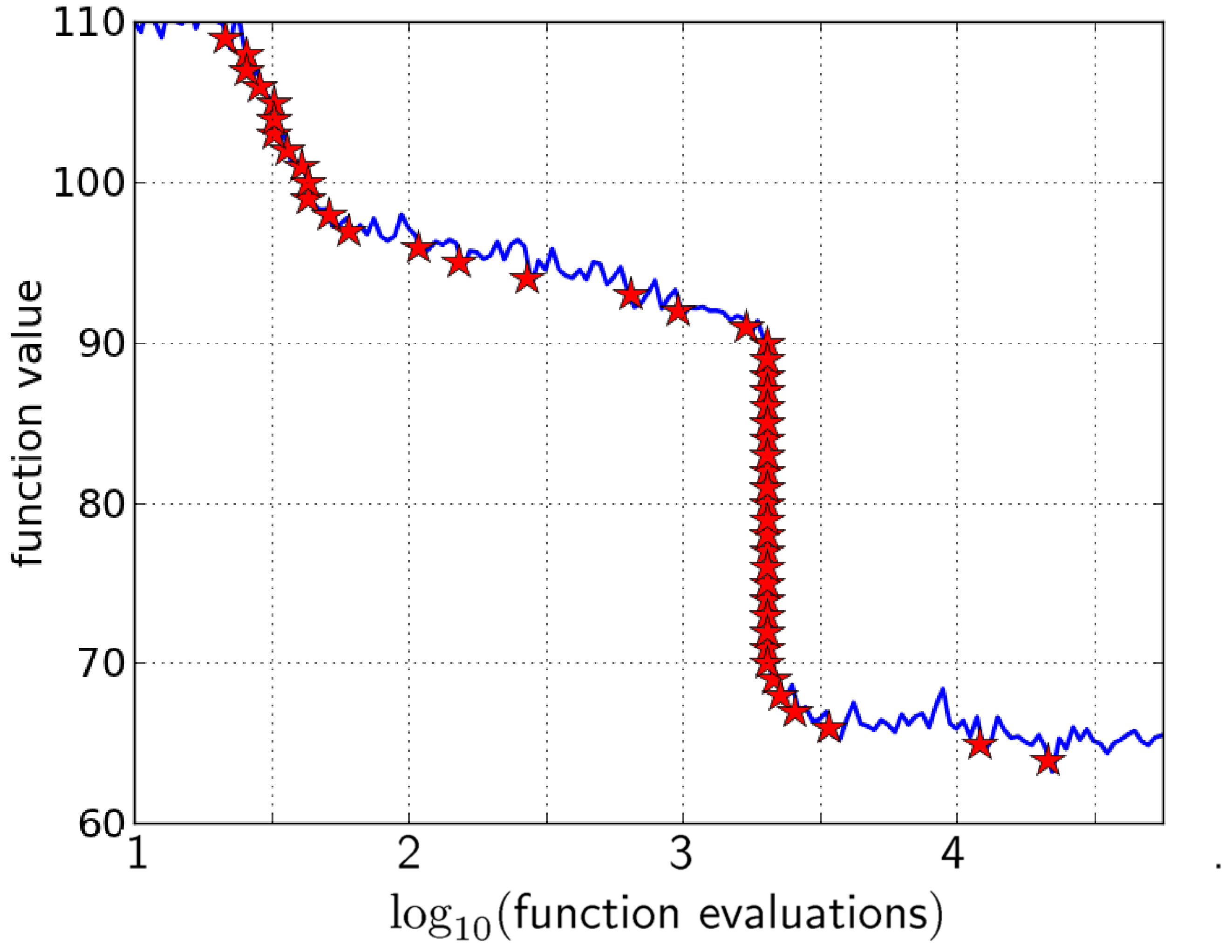


Reconstructing Single Runs: Recording More Targets

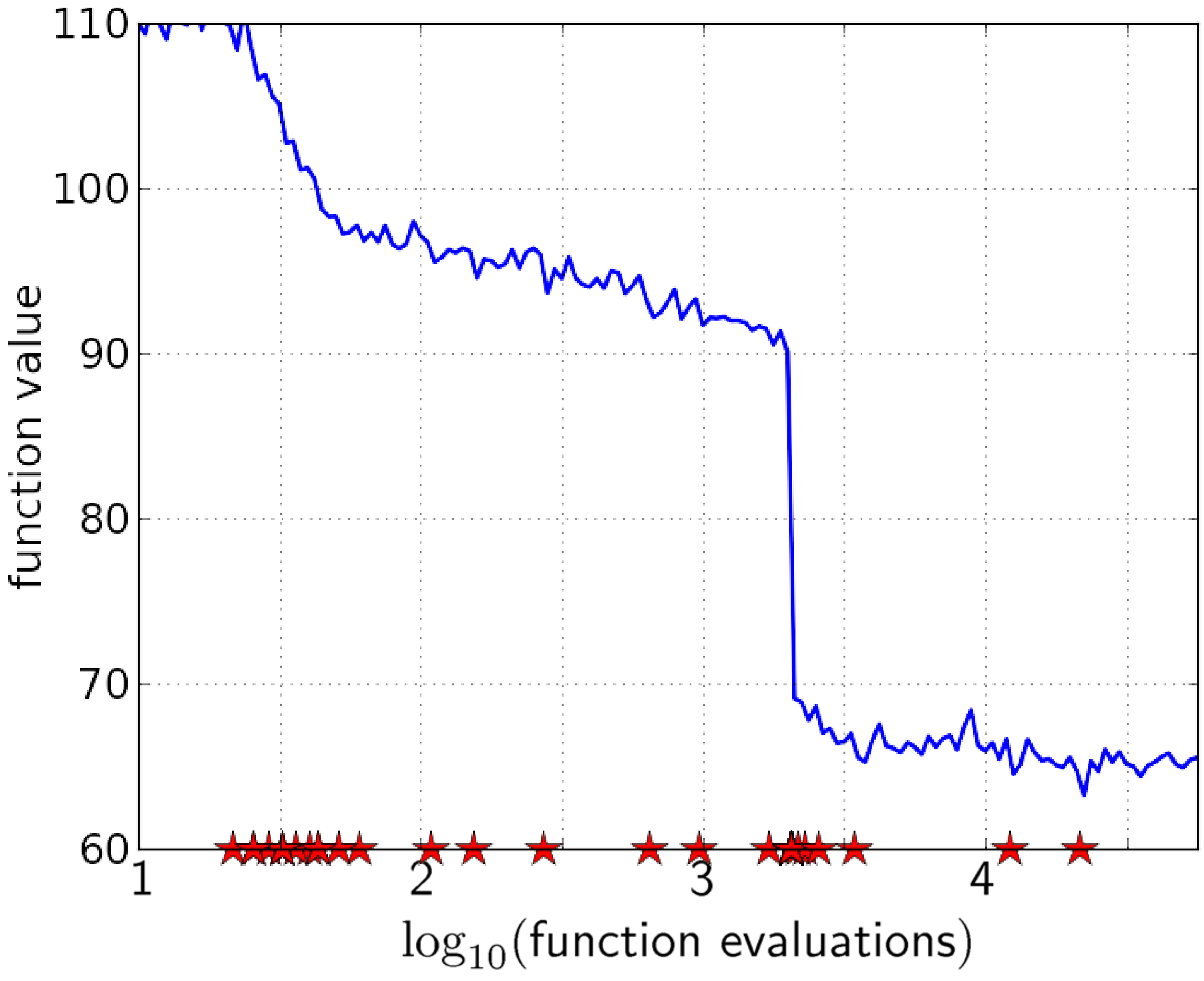


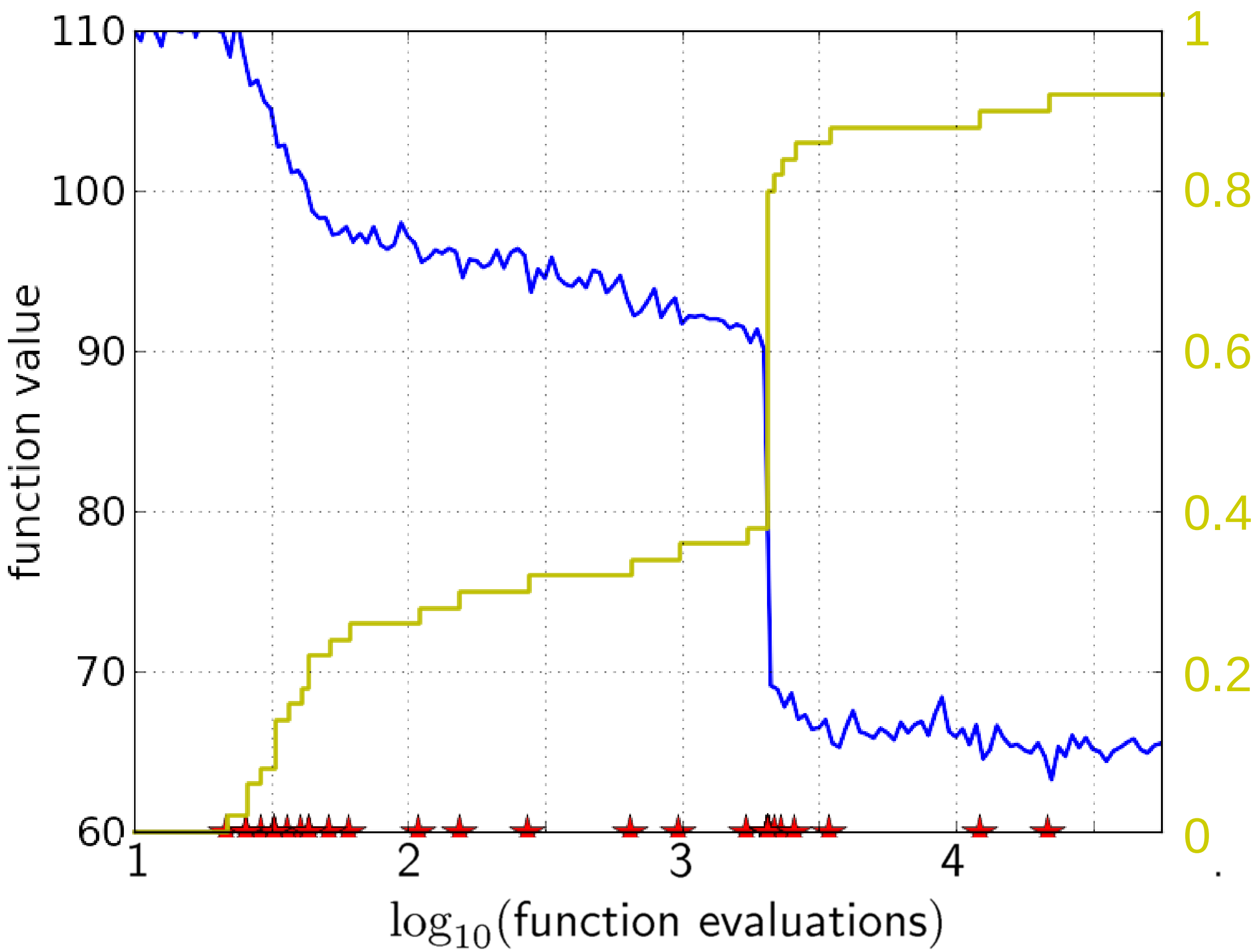
50 equally spaced targets

Reconstructing Single Runs: Recording More Targets



Reconstructing Single Runs: Recording More

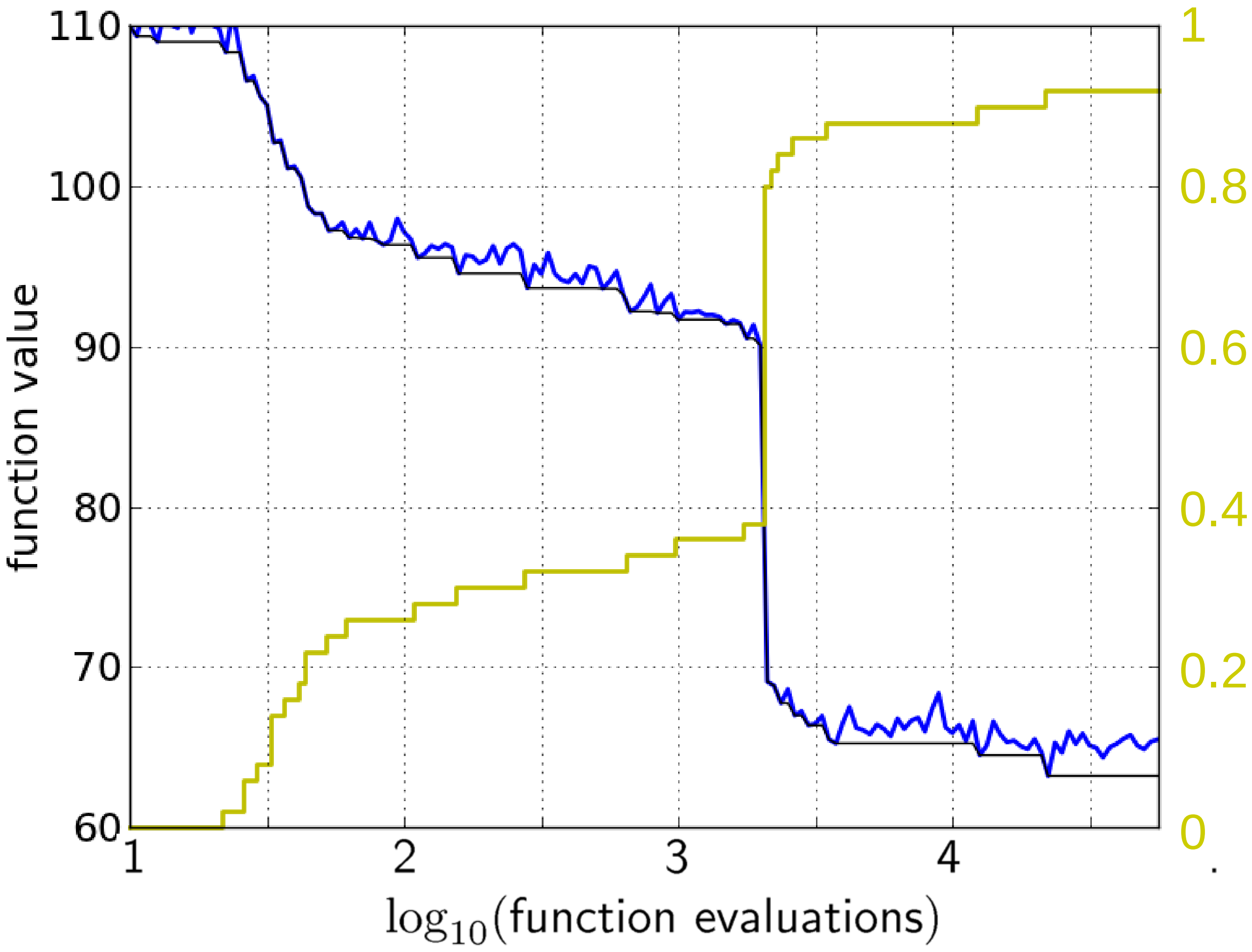




the empirical CDF makes a step for each star, is monotonous and displays for each budget the fraction of targets achieved within the budget

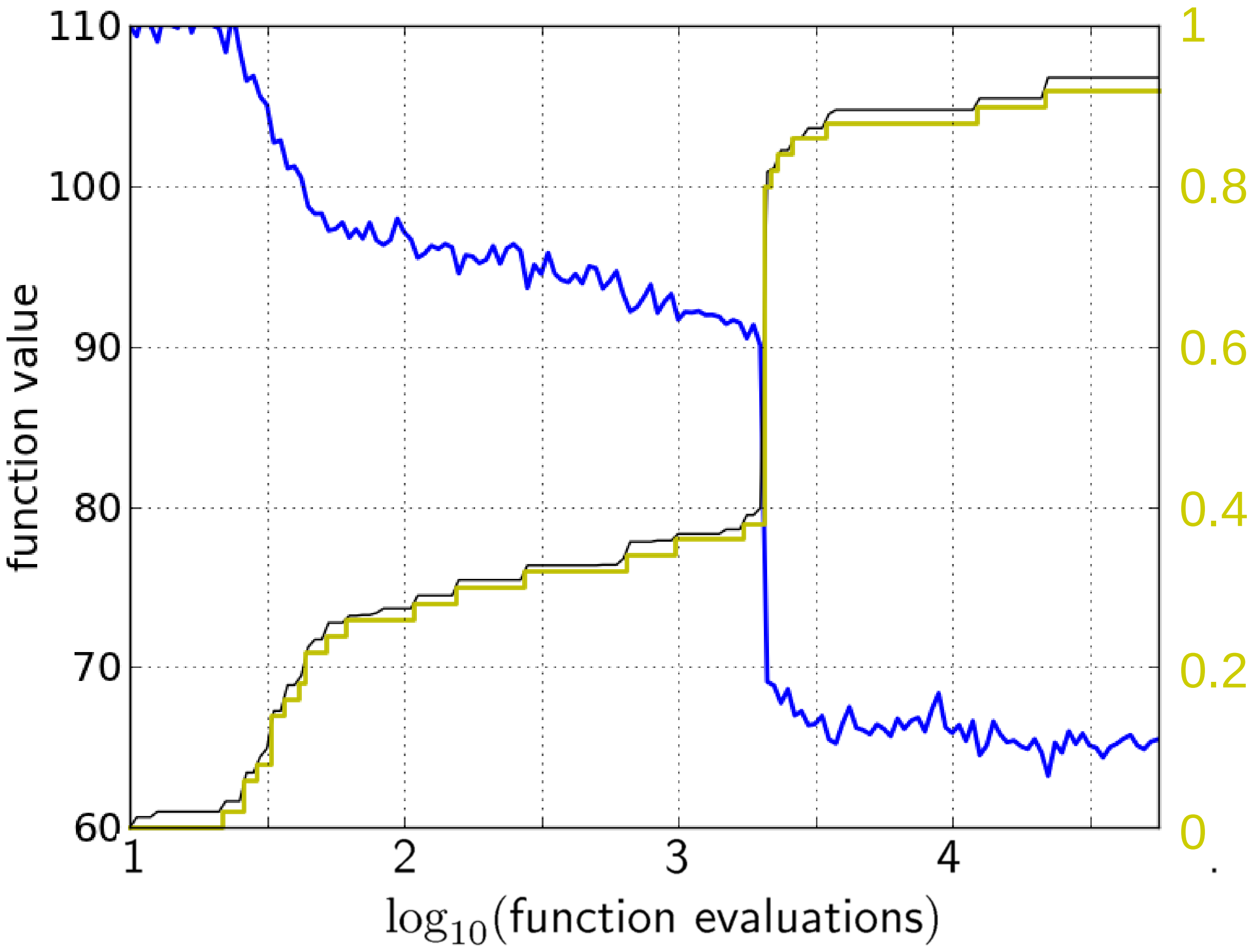
ECDF Recovers the Convergence Graph

...discretized and flipped

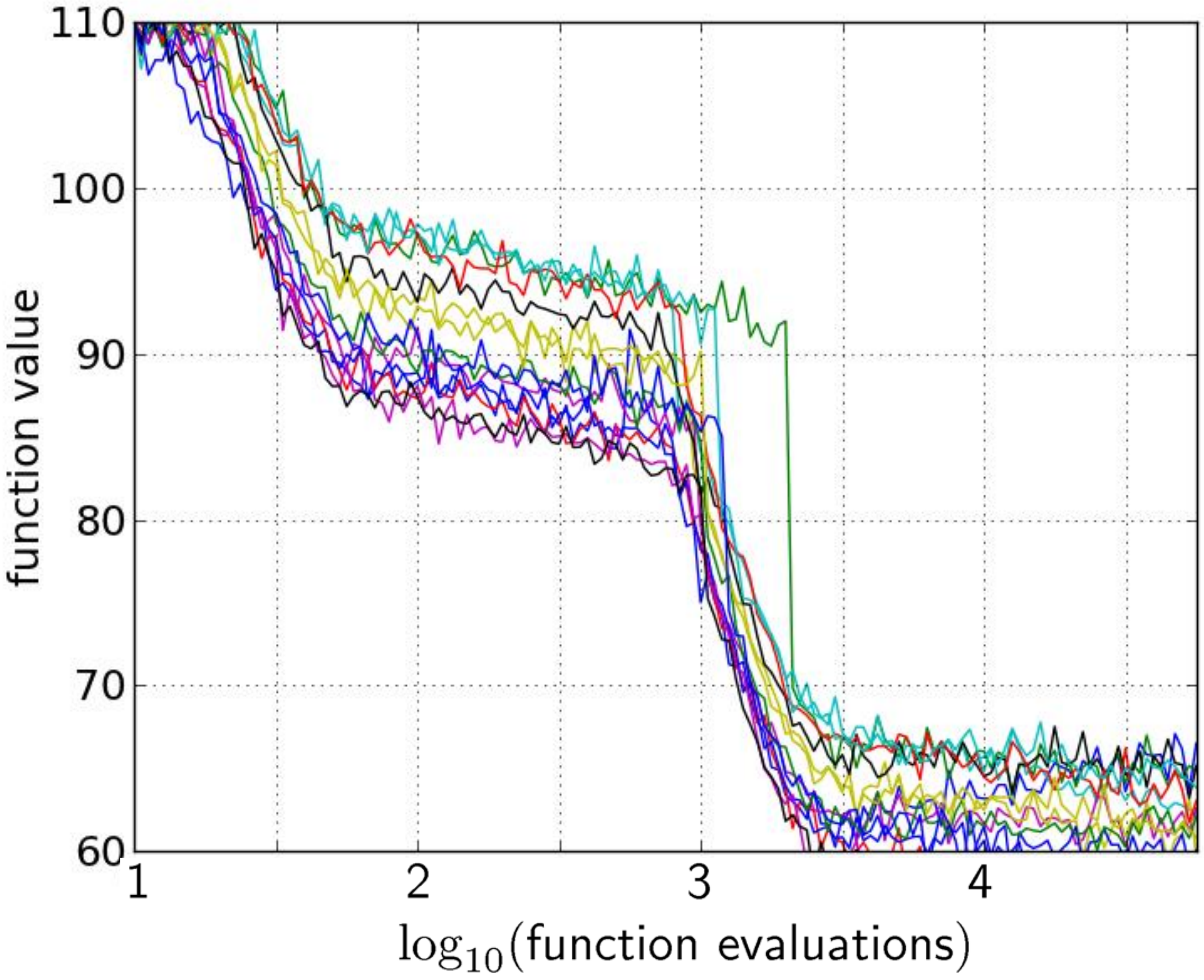


ECDF Recovers the Convergence Graph

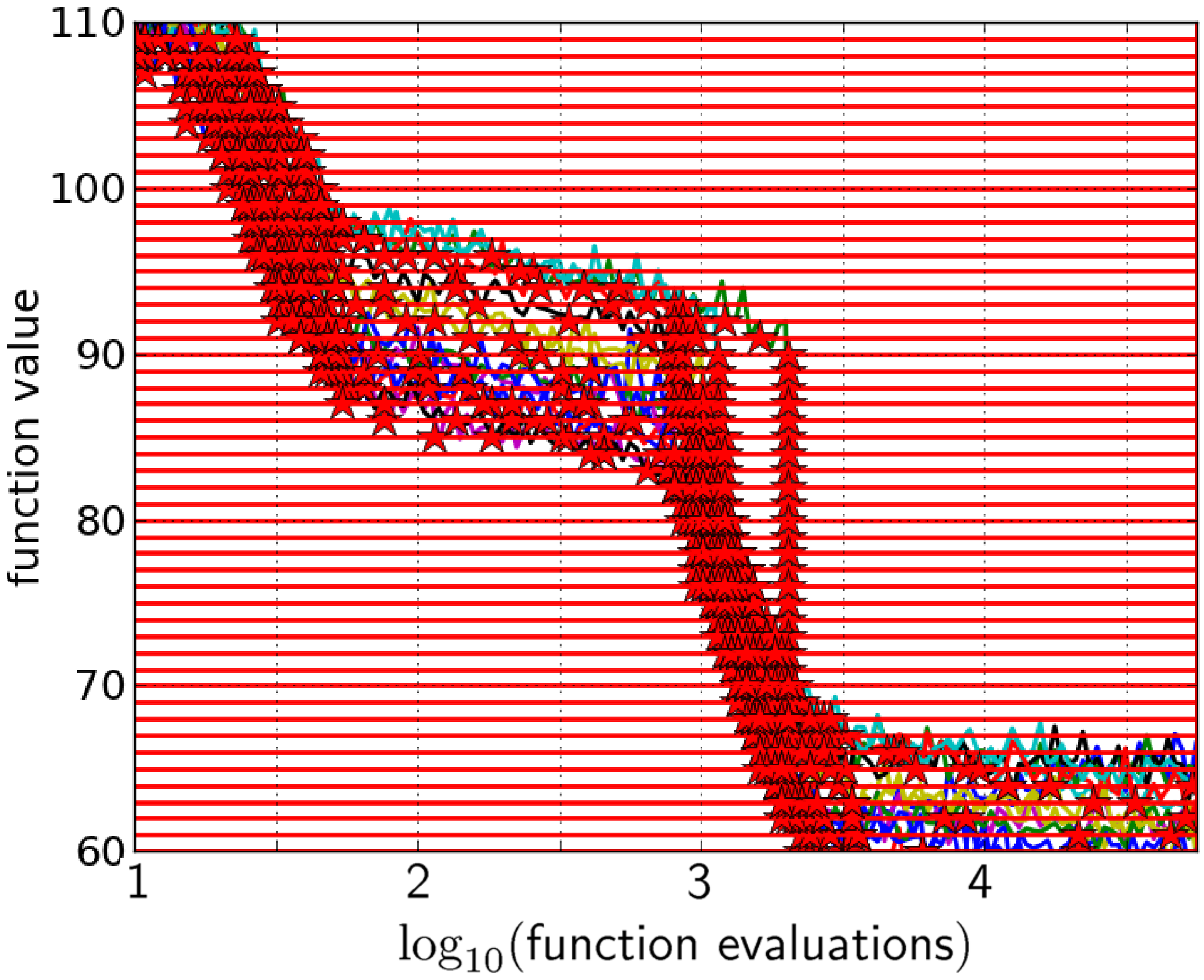
...discretized and flipped



Aggregation of Optimization Runs

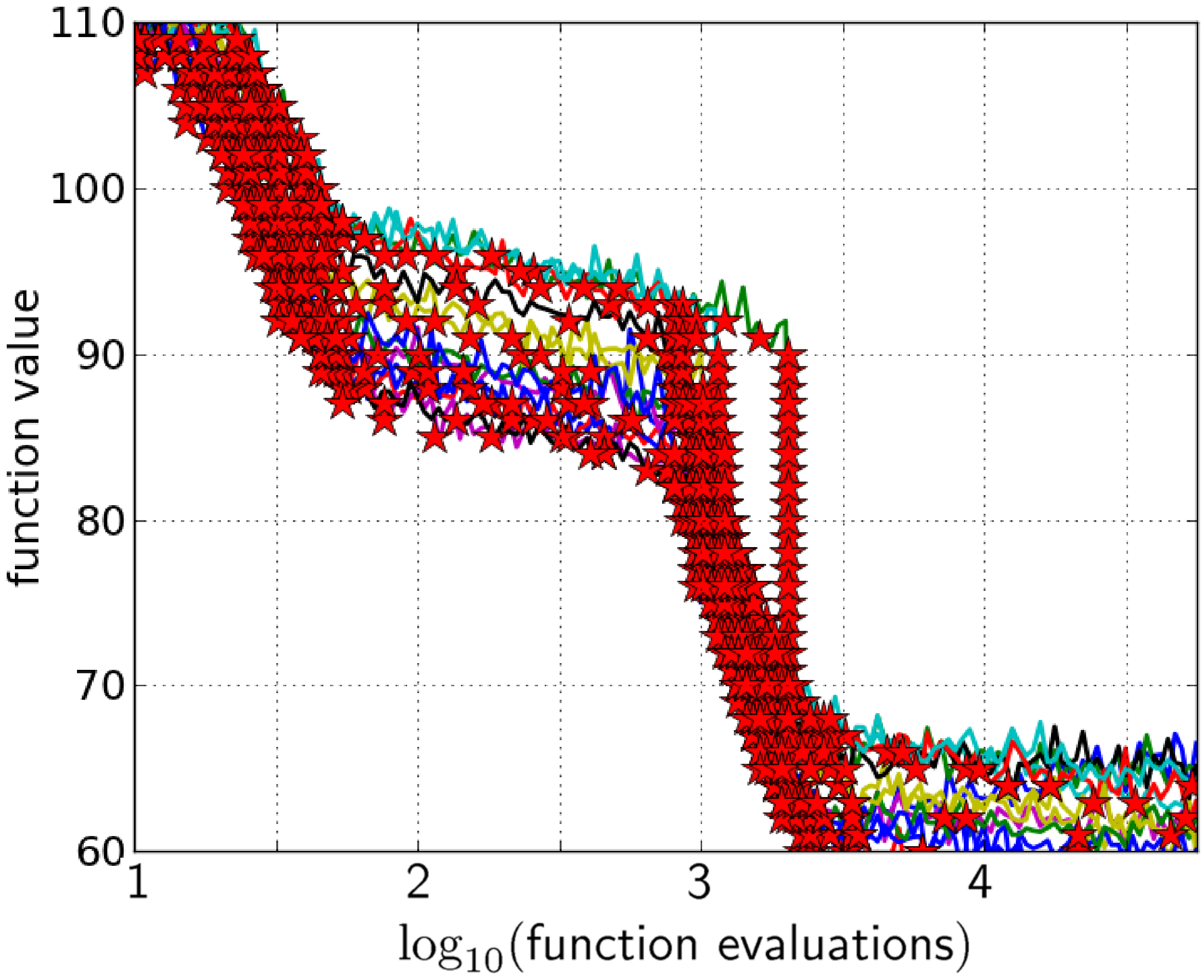


Aggregation of Optimization Runs



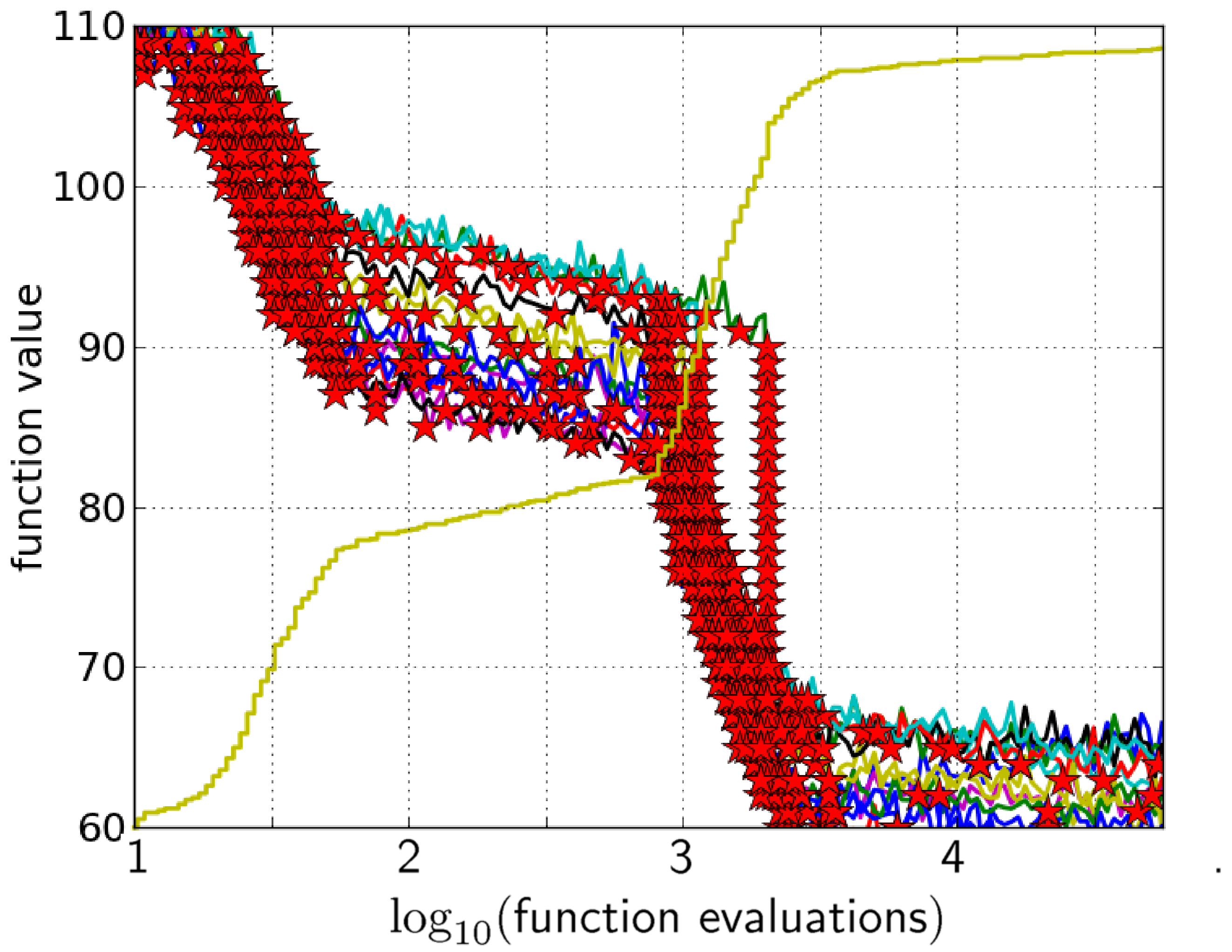
.

Aggregation of Optimization Runs

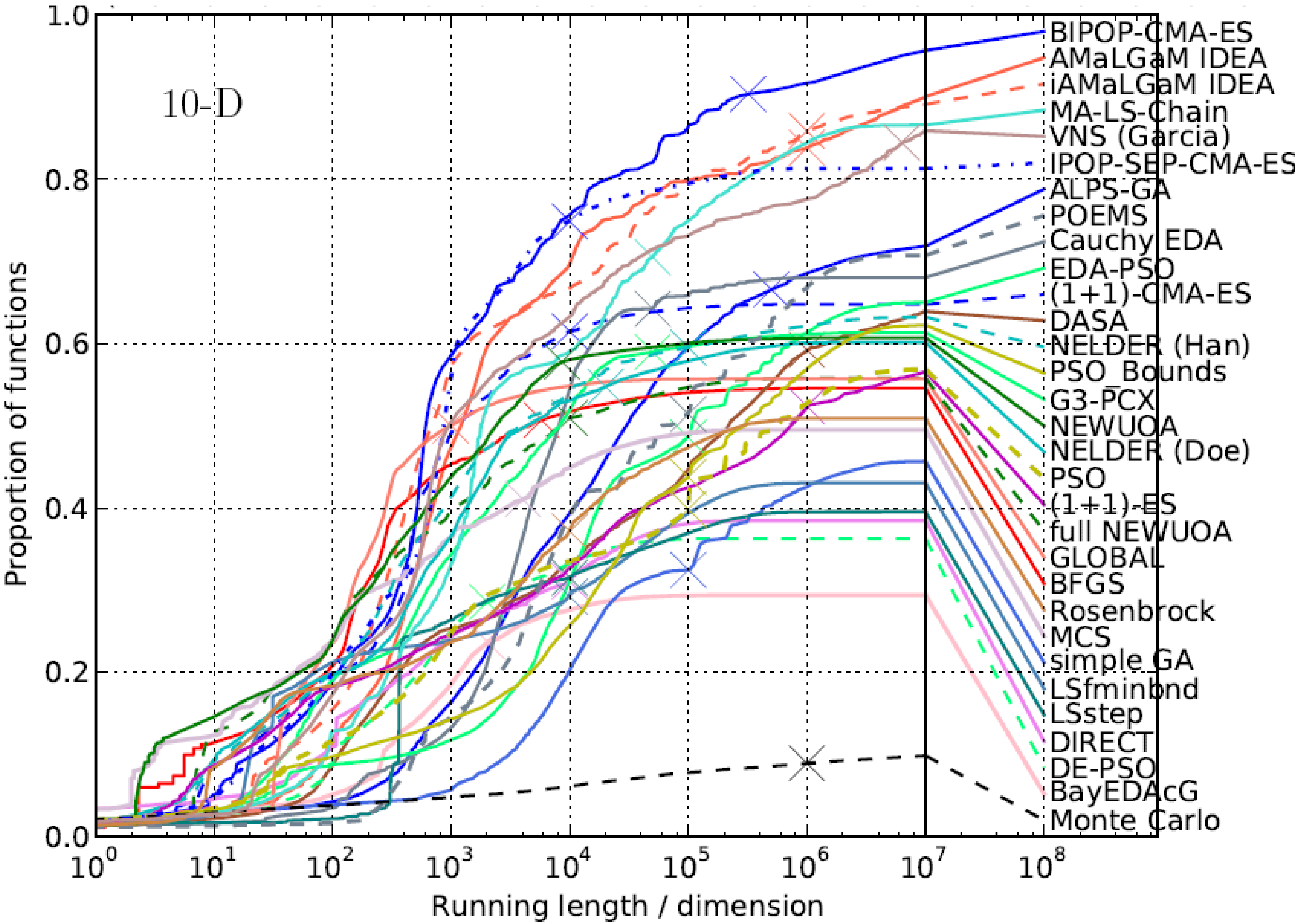


.

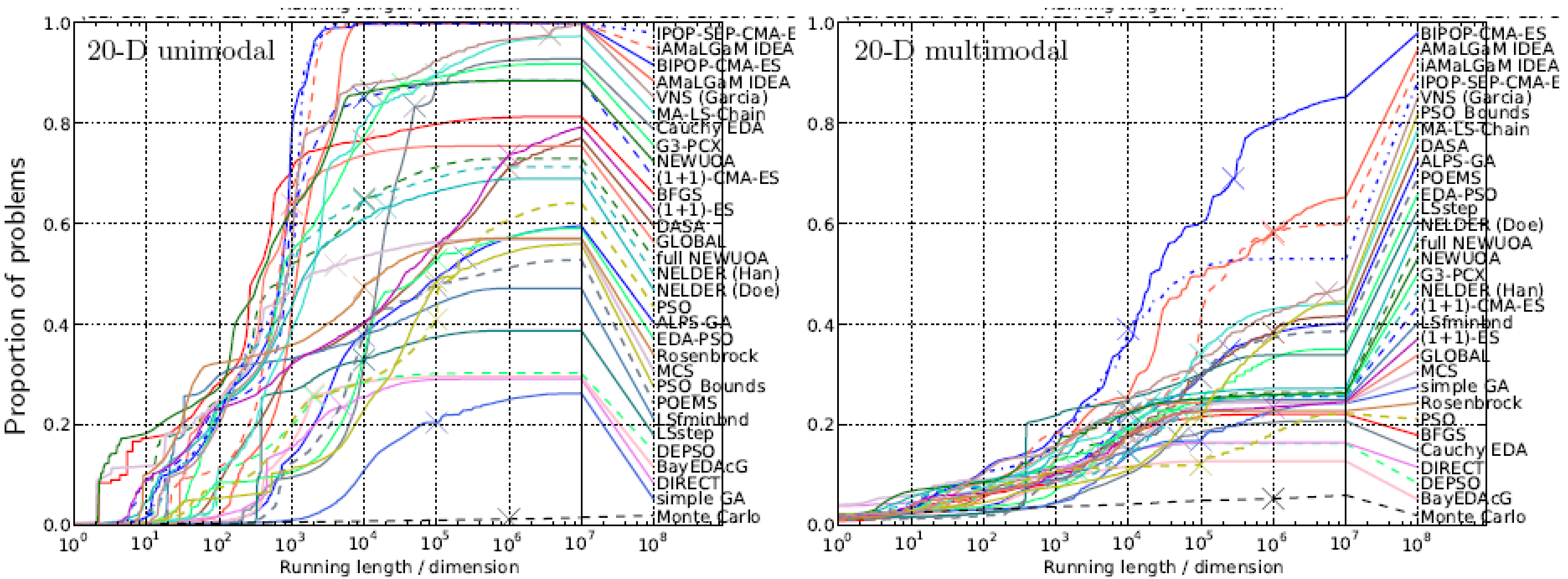
Aggregation of Optimization Runs



Exemplary Results



Unimodal vs. Multimodal



Announcement

CEC Special Session on

Black Box Optimization Benchmarking (CEC-BBOB’2015)

organizers:

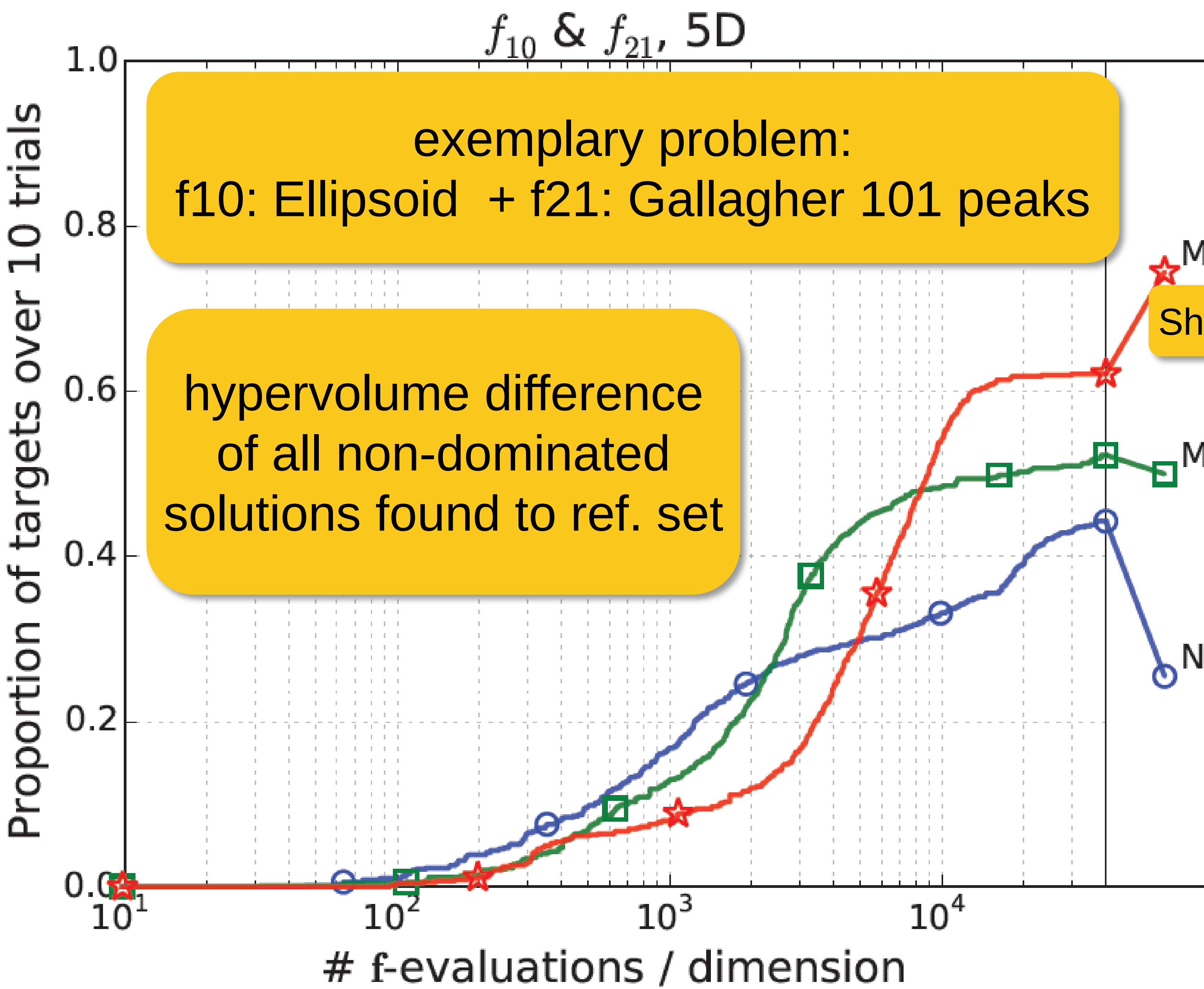
Y. Akimoto, A. Auger, D. Brockhoff, N. Hansen, O. Mersmann, P. Pošík

submission deadline: **December 19, 2014**

http://coco.gforge.inria.fr/

Back to
Multi-Objective Benchmarking

Data Profiles

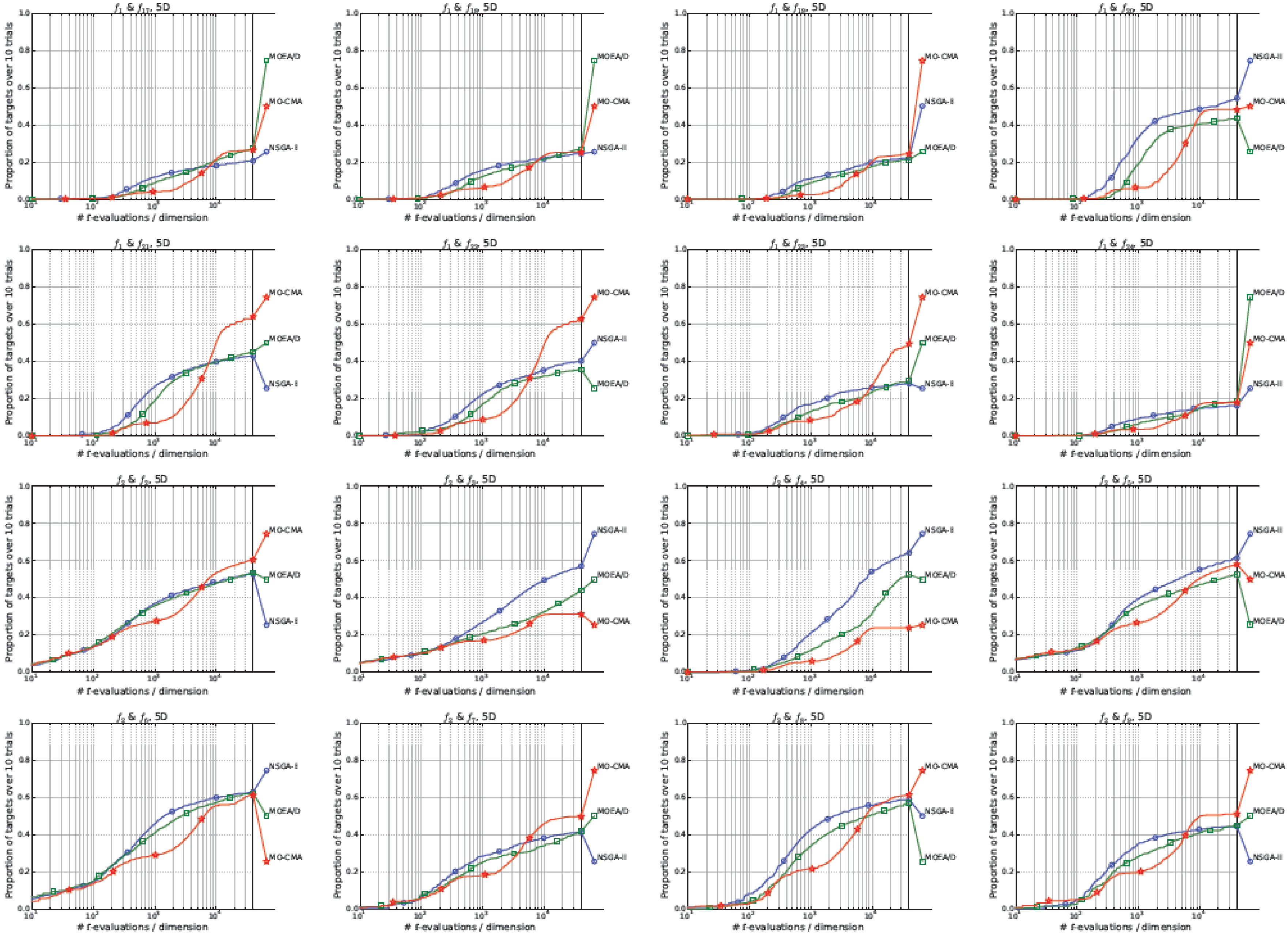


exemplary problem:
f10: Ellipsoid + f21: Gallagher 101 peaks

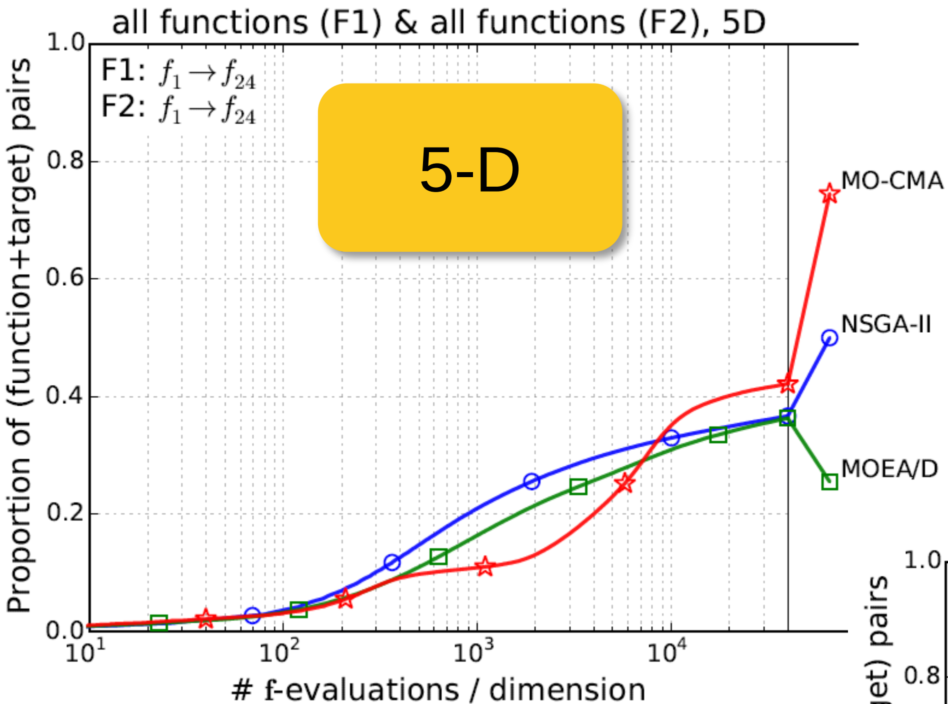
hypervolume difference of all non-dominated solutions found to ref. set

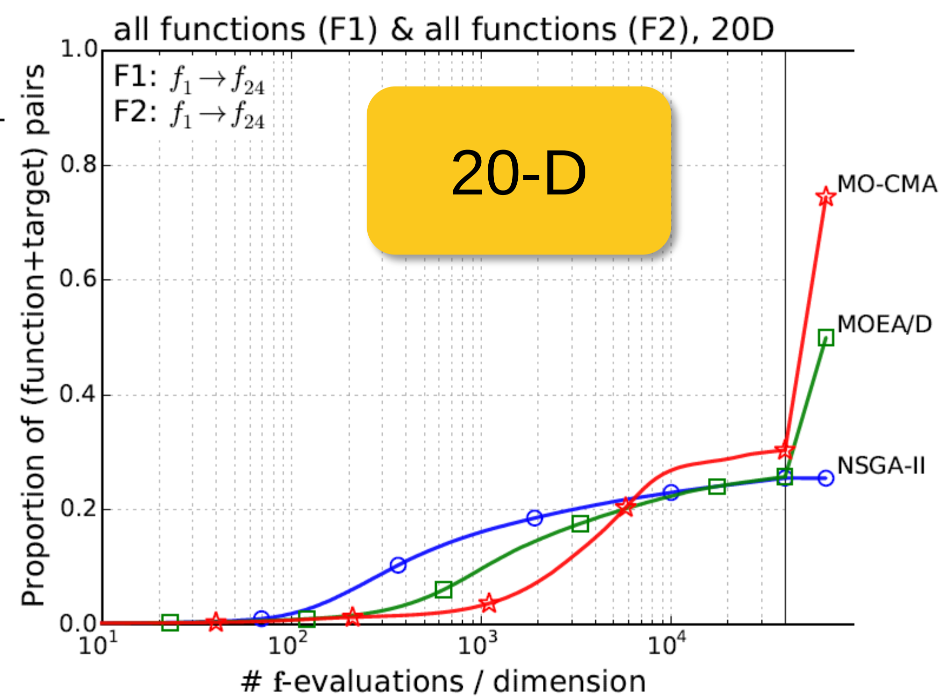
Shark version

More Problems

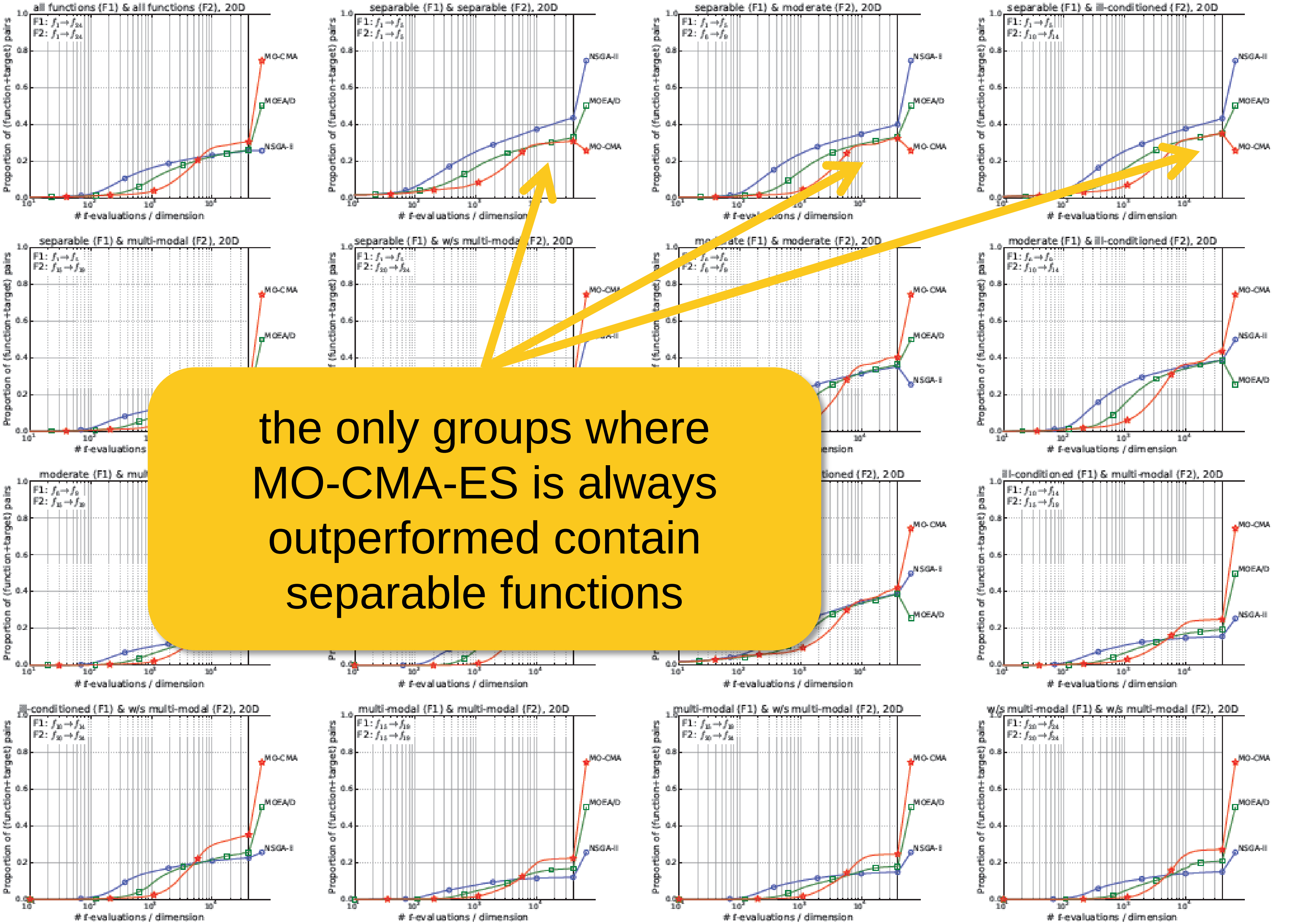


Aggregation Over All BBOB Problems (300 total)

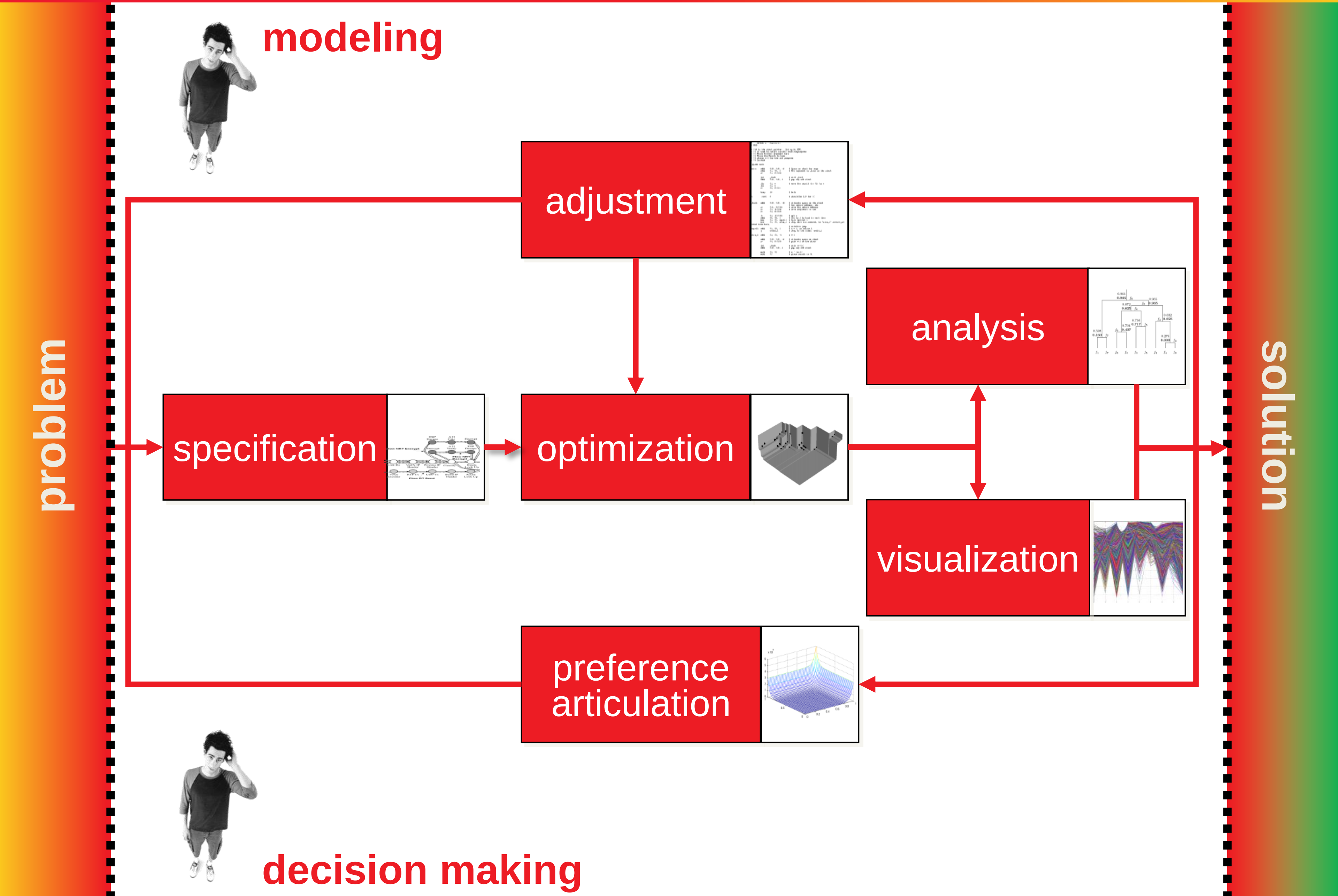




Aggregation Over Function Groups



Conclusions: EMO as Interactive Decision Support



The EMO Community

Links:

- EMO mailing list: *https://lists.dei.uc.pt/mailman/listinfo/emo-list*
- MCDM mailing list: *http://lists.jyu.fi/mailman/listinfo/mcdm-discussion*
- EMO bibliography: *http://www.lania.mx/~ccoello/EMOO/*
- EMO conference series: *http://www.dep.uminho.pt/EMO2015/*

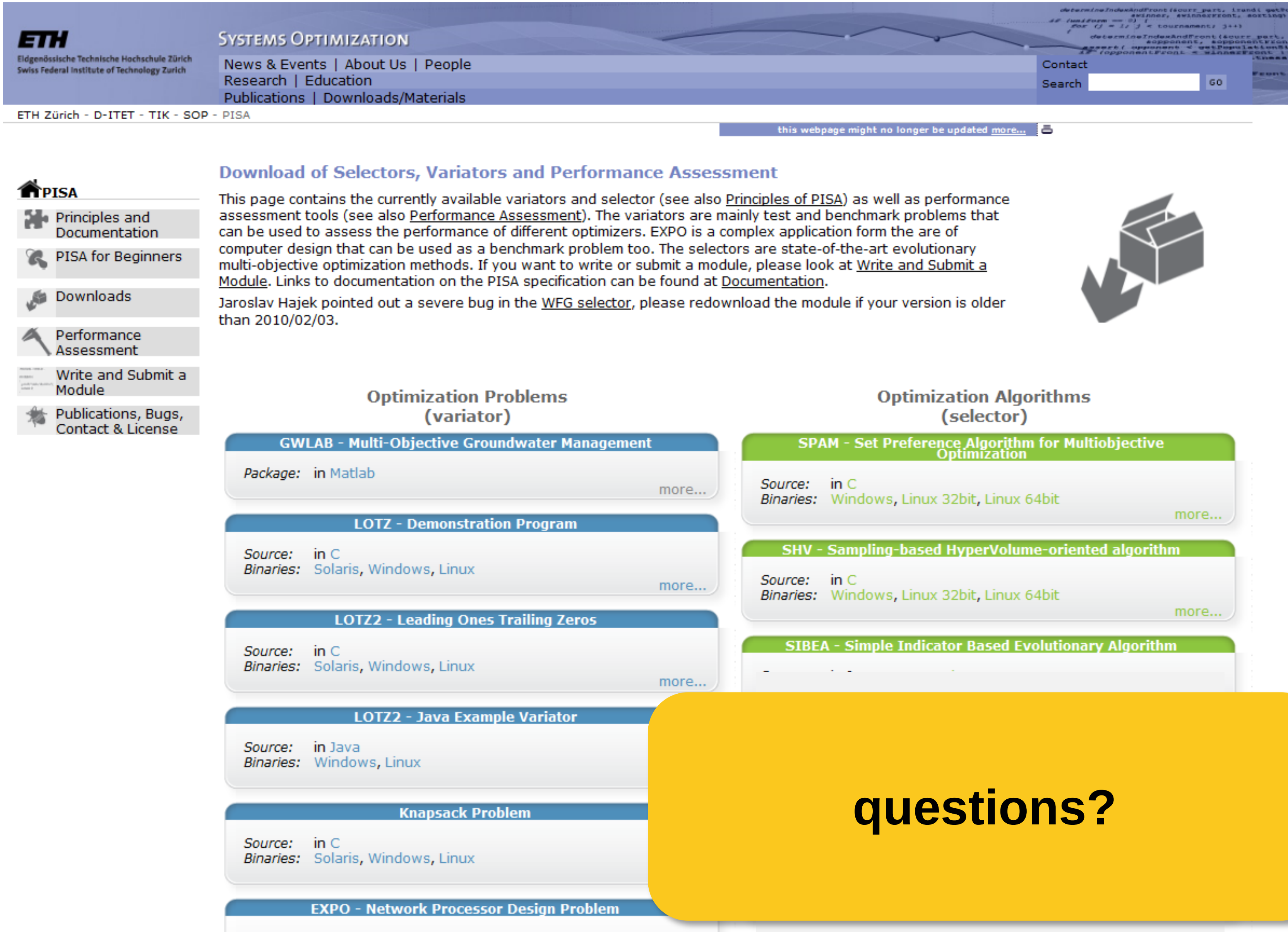
Books:

- ***Multi-Objective Optimization using Evolutionary Algorithms***
  Kalyanmoy Deb, Wiley, 2001
- ***Evolutionary Algorithms for Solving Multi Evolutionary Algorithms for Solving Multi-Objective Problems Objective Problems***, Carlos A. Coello Coello, David A. Van Veldhuizen & Gary B. Lamont, Kluwer, 2nd Ed. 2007
- **Multiobjective Optimization—Interactive and Evolutionary Approaches**, J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, volume 5252 of *LNCS*. Springer, 2008
- and more…

PISA: http://www.tik.ee.ethz.ch/pisa/

and many more:
jmetal, Shark,
MOEA Framework,
...

PISA: http://www.tik.ee.ethz.ch/pisa/



questions?

References

References

[Bader and Zitzler 2011] J. Bader and E. Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. Evolutionary Computation 19(1):45-76, 2011.

[Bringmann and Friedrich 2009] K. Bringmann and T. Friedrich. Approximating the Least Hypervolume Contributor: NP-hard in General, But Fast in Practice. In M. Ehrgott et al., editors, Conference on Evolutionary Multi-Criterion Optimization (EMO 2009),pages 6–20. Springer, 2009

[Fonseca et al. 2011] C. M. Fonseca, A. P. Guerreiro, M. López-Ibáñez, and L. Paquete. On the computation of the empirical attainment function. In Conference on Evolutionary Multi-Criterion Optimization (EMO 2011). Volume 6576 of LNCS, pp. 106-120, Springer, 2011

[Goldberg 1989] D. E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Massachusetts, 1989

[Hansen and Ostermeier 1996] N. Hansen and A. Ostermeier. **Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation**. In *Congress on Evolutionary Computation (CEC 1996)*, pages 312–317, Piscataway, NJ, USA, 1996. IEEE. [476 —

[Hansen and Ostermeier 2001] N. Hansen and A. Ostermeier. **Completely Derandomized Self-Adaptation in Evolution Strategies**. *Evolutionary Computation*, 9(2):159–195, 2001.

[Hansen et al 2010] N. Hansen, A. Auger, R. Ros, S. Finck, and P. Posik. **Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009**. In *Genetic and Evolutionary Computation Conference (GECCO 2010)*, pages 1689–1696, 2010. ACM

References

[Igel et al. 2007] C. Igel, N. Hansen, and S. Roth. Covariance Matrix Adaptation for Multi-objective Optimization. Evolutionary Computation, 15(1):1–28, 2007

[Judt et al. 2011] L. Judt, O. Mersmann, and B. Naujoks. Non-monotonicity of obtained hypervolume in 1-greedy S-Metric Selection. In: Conference on Multiple Criteria Decision Making (MCDM 2011), abstract, 2011.

[Miettinen 1999] K. Miettinen. Nonlinear Multiobjective Optimization. Kluwer, Boston, MA, USA, 1999

[Voß et al. 2009] T. Voß, N. Hansen, and C. Igel. Recombination for Learning Strategy Parameters in the MO-CMA-ES. In *Evolutionary Multi-Criterion Optimization (EMO 2009)*, pages 155–168. Springer, 2009.

[Voß et al. 2010] T. Voß, N. Hansen, and C. Igel. Improved Step Size Adaptation for the MO-CMA-ES. In J. Branke et al., editors, Genetic and Evolutionary Computation Conference (GECCO 2010), pages 487–494. ACM, 2010

[Zhang and Li 2007] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation, 11(6):712--731, 2007

[Zitzler et al. 2003] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation, 7(2):117–132, 2003