It is Time for a Revision of COCO BBOB

Hans-Georg Beyer
Hans-Georg.Beyer@evolutionary-computation.com
http://evolutionary-computation.com

Research Center Business Informatics

FHV
Vorarlberg University
of Applied Sciences

The Weaknesses of COCO BBOB

Caveat: the considerations presented refer to the test beds:
bbob, bbob-noisy, and bbob-largescale[1]

Basic Design Principle:
Given a minimization problem:

$$F(\mathbf{y}) \to \min, \quad \mathbf{y} \in \mathbb{R}^D \tag{1.1}$$

use parental $F^{(g)}$ dynamics to generate aggregated performance measures:

ERT: estimate the expected running time to reach a target value F_{target}, where the running time is defined as the number of F-function evaluations r (run length)

ECDF: considering the $F^{(g)}$-dynamics (of a number of independent runs) and construct the empirical cumulative distribution of the run lengths r:

1. under fixed F_{target} conditions (“horizontal view”)
2. under fixed F-function evaluation budget (“vertical view”)

[1] This does not exclude that parts are also valid for the other test suits.

Empirical Cumulated Distribution Function (ECDF) of Run Lengths r

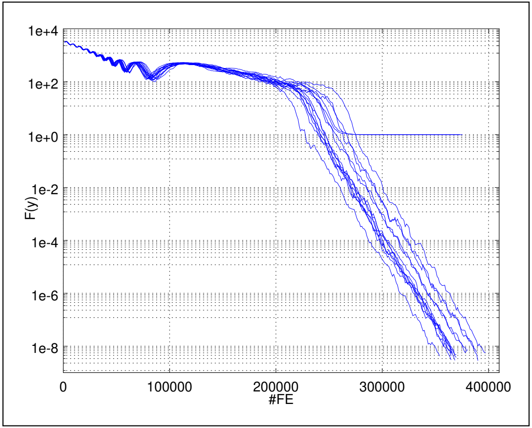

Figure 1.1: Fitness F vs. number of function evaluations #FE of a $(400/400_I, 1200)$-σSA-ES on the Rastrigin function in $D = 30$ dimensional search space. 15 independent runs are displayed using the *same* initialization.

- the run length r is the number of function evaluations (#FE) the algorithm consumes until reaching a certain goal
- e.g., F target value, σ mutation strength target, etc.
- runs may not reach the predefined F goal
- those runs that reach an F goal may have different run lengths r

$\Rightarrow$ r is a random variate, thus it has a distribution

The Fixed Target Scenario: Horizontal ECDF View

- the horizontal ECDF determines the percentage of runs that reach a certain F target value for a given number of function evaluations #FE
- technically, one has to go through the #trials F(#FE) curves and look for the first appearance of an $F(\#\mathrm{FE}) \le F_{\mathrm{target}}$ yielding the first step (of size $1/$#trials) in the ECDF plot
- removing the corresponding F curve, the process is repeated #trials times

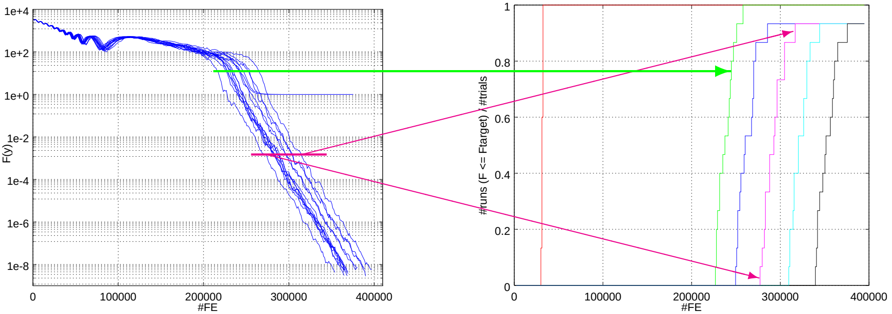

Figure 1.2: The ECDFs of the run lengths (right graph) derived from the fitness dynamics (left graph) for F target values 10^3, 10, 0.1, 10^{-3}, 10^{-5}, 10^{-7} (left to right).

The Fixed Cost Scenario: the Vertical ECDF View

- the vertical ECDF considers a *fixed* #FE budget and provides the percentage of runs that yield F values for which $F \leq f$
- technically, one searches for the generation in which #FE reaches the predefined #FE budget and sorts ascendingly F values of all #trials runs
- these values are the start values (horizontal) for the steps of size $1/\#\text{trial}$

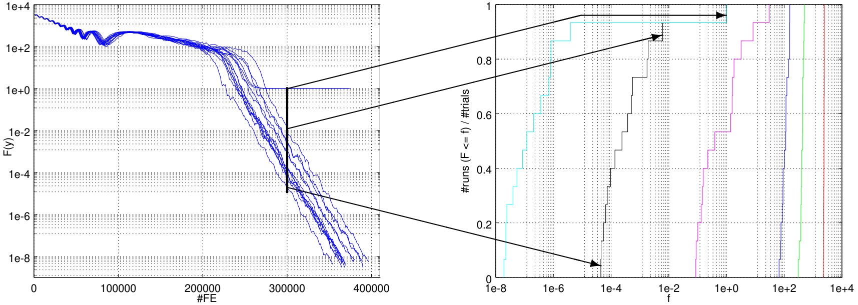

Figure 1.3: ECDFs (right graph) of F values reached in the fitness dynamics (left graph) for given #FE budgets 10^4, 10^5, $2\cdot10^5$, $2.5\cdot10^5$, $3\cdot10^5$, $3.5\cdot10^5$ (right to left).

- the ECDF graphs can be also used for *aggregated* performance evaluation considering more than one test function at a time
  - the only difference to Figs. 1.2 and 1.3 is that the fitness dynamics of the algorithm for more than one test function are considered simultaneously
  - *however, note that in doing so, one compares apples and oranges*
- in the COCO framework, one even aggregates *different F* targets of one or more test functions to get a global ECDF measure:

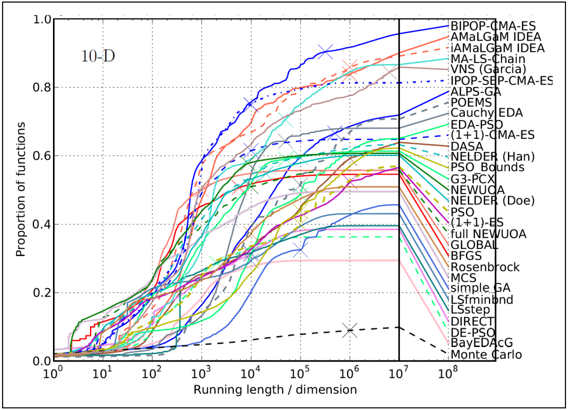

Figure 1.4: Aggregated ECDF of the 24 BBOB test functions over all targets in $D = 10$ dimensions taken from the 2010 GECCO paper: *Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking BBOB-2009*.

Critique:

- the only information that can be drawn from such aggregations is that given a function budget #FE, a percentage of F targets has been reached
- this does not provide any information how an algorithm really performs
- yet, it is a simple measure to isolate a winner in black box optimization competitions, such as the BBOB contest

Benefit:

- the ECDF graphs can be well used to compare different algorithms if carefully crafted

☞ this is the only methodically sound usage of ECDF graphs

There are certain COCO design decisions that should be questioned:

1. How does the algorithm perform in search space?
   Instead of considering the F-dynamics, one should *alternatively* consider the approach of $\mathbf{y}$ towards the optimizer $\hat{\mathbf{y}} = \mathrm{argopt}_{\mathbf{y}} F(\mathbf{y})$.
   (impossible to be realized in the COCO BBOB framework)
2. For the noisy test bed, considering the noisy F-values for performance evaluation is problematic: the noise-free F-values should be used.
   (difficult to be realized in the COCO BBOB framework)
3. The linear distortions and rotations of the search space are biased toward CMA-ES like algorithms.
4. The nonlinear distortions applied are rather weak perturbations that can be treated well by CMA-ES

5. Considering target differences of 10^{-8} are far too precise for most RWA:
   - In COCO BBOB 51 (or 41) F_{target} are considered to calculate the ECDF
   - Between 10^{-6} and 10^{-8} there are 11 targets that almost always correspond to the final convergence in a global attractor.
   - In that case the local fitness landscape is often a quadratic one, well suited for CMA-like algorithms.
   - ☞ As a result, more than 20% of the targets "improve" the *global ECDF of* such EAs without any practical relevance.
6. How is the choice of the 24 test functions related to real-world problems?
7. The test function set has not been changed for more than 15 years.
8. Most of these functions are not a challenge for CMA-ES, or the function parameters are chosen in such a manner that CMA-like EAs converge (at least for the BiPop-version).

On the Choice of the Test Functions

Example: Ellipsoid f_2 and f_{10}

$$F(\mathbf{x}) := \sum_{i=1}^{D} a_i x_i^2 \quad \text{with} \quad a_i = 10^{\alpha \frac{i-1}{D-1}} \quad \text{and} \quad \alpha = 6 \tag{2.1}$$

$$f_2(\mathbf{y}) := F\big(\mathrm{T}_{\mathrm{osz}}(\mathbf{y} - \mathbf{y}^{\mathrm{opt}})\big) \tag{2.2}$$

$$f_{10}(\mathbf{y}) := F\big(\mathrm{T}_{\mathrm{osz}}(\mathbf{R}(\mathbf{y} - \mathbf{y}^{\mathrm{opt}}))\big) \tag{2.3}$$

- condition number of (2.1): $\mathrm{cond} = 10^6$ (highly conditioned)
- however, cond does not depend on search space dimensionality D

☞ function (2.1) is not suited for evaluating the complexity of the covariance matrix learning

- as has been shown by Beyer & Melkozerov (for σSA) and Beyer & Hellwig (for CSA) the ERT of general quadratic models

$$F(\mathbf{x}) := \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x}, \quad \mathbf{A} \in \mathbb{R}^{D\times D}, \;\; \mathbf{A} \succ 0 \tag{2.4}$$

can be estimated for $(\mu/\mu_I, \lambda)$-ES by

$$\mathrm{ERT}(D) = \mathcal{O}\left(\frac{\mathrm{Tr}[\mathbf{A}]}{\min_i(a_i)}\right) \tag{2.5}$$

where $\mathrm{Tr}[\mathbf{A}]$ is the trace of $\mathbf{A}$ and $\min_i(a_i)$ is the *smallest* eigenvalue of $\mathbf{A}$

- since $\min_i(a_i) = 1$ and $\mathrm{Tr}[\mathbf{A}] = \sum_{i=0}^{D-1} 10^{\frac{\alpha i}{D-1}} = \frac{10^{\frac{\alpha D}{D-1}}-1}{10^{\frac{\alpha}{D-1}}-1} \overset{D\to\infty}{\simeq} \frac{10^{\alpha}-1}{\alpha\ln(10)}D$

$\Rightarrow$ $\mathrm{ERT}(D) = \mathcal{O}(D)$

- the ERT(D) scales linearly even for isotropic mutations as the sphere!

☞ if D is sufficiently large, optimizing the BBOB ellipsoid is from viewpoint of computational complexity not harder than the sphere model

☞ since the covariance learning needs $\mathcal{O}(D^2)$ black-box queries, i.e. F-evaluations, there is a D above which the CMA-ES has reached the final target *without* having completed the covariance matrix learning!

- while this is not an issue considering small dimensionalities D, if D is sufficiently large, the covariance learning is by far not completed before the fitness target has been reached

☞ include test functions with cond $= f(D)$ with $a_i = i^\beta$ in BBOB

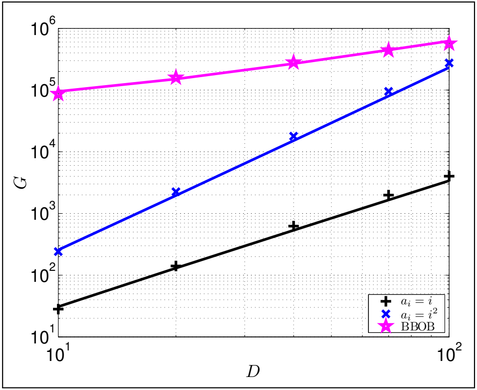

Figure 2.1: Expected running time (measured in generations, i.e., #FE $= \lambda G$) of the ellipsoid (2.1) with $\alpha = 5$ compared to $a_i = i$ and $a_i = i^2$ ellipsoids using a $(3/3_I, 10)$-σSA-ES with isotropic mutations. Plot taken from Beyer & Melkozerov (2013): *The Dynamics of Self-Adaptive Multi-Recombinant Evolution Strategies on the General Ellipsoid*.

BBOB Needs More Challenging Test Functions

- should be *unimodal* and *scalable* w.r.t. dimensionality D and hardness β
- the optimizer should be known and easily visible in $D = 2$

The HappyCat Function:

$$F_{\mathrm{HC}}(\mathbf{y}) := \sqrt[\beta]{(\|\mathbf{y}\|^2 - D)^2} + \frac{1}{D}\left(\frac{\|\mathbf{y}\|^2}{2} + \sum_{i=1}^{D} y_i\right) + \frac{1}{2} \quad \text{with } \beta = 8 \quad (3.1)$$

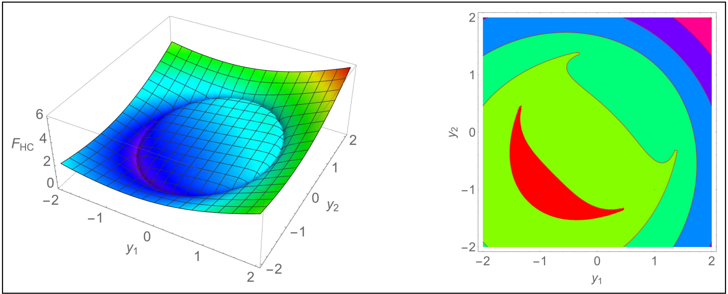

Figure 3.1: HappyCat in two dimensions; optimizer at: $\hat{\mathbf{y}} = (-1, \ldots, -1)^{\mathrm{T}}$ (see Beyer & Finck, PPSN XII, 2012) ☞ live simulation.

- a human can see the solution for $D = 2$, however, reaching the optimizer $\hat{\mathbf{y}}$ seems hard for CMA-ES (and other algorithms)
- for $D > 2$ the distance to $\hat{\mathbf{y}}$ increases further
- hardness can be controlled by $\beta \uparrow$ ($\beta = 2$ is easy)

- HappyCat is used in regular CEC contests, however, embedded in composite functions
- up to now, this test function has been exclusively used in the 2019 CEC 100-Digit Challenge with $D = 10$
- only jDE100 was able to master the challenge with HappyCat

The HGBat Function:

$$F_{\mathrm{HGB}}(\mathbf{y}) := \sqrt[\beta]{\left(\|\mathbf{y}\|^4 - \left(\sum_{i=1}^{D} y_i\right)^2\right)^2} + \frac{1}{D}\left(\frac{\|\mathbf{y}\|^2}{2} + \sum_{i=1}^{D} y_i\right) + \frac{1}{2} \quad \text{with } \beta = 4 \tag{3.2}$$

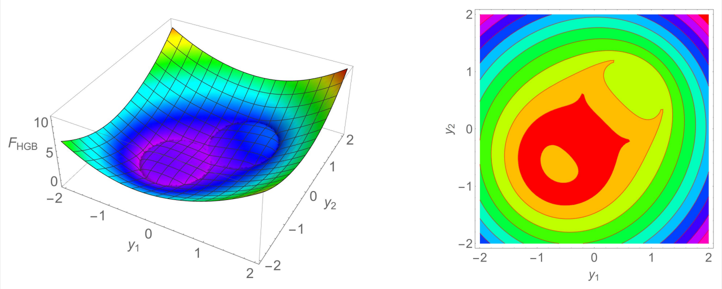

Figure 3.2: HGBat in two dimensions; optimizer at: $\hat{\mathbf{y}} = (-1, \ldots, -1)^{\mathrm{T}}$ (see Beyer & Finck, PPSN XII, 2012) ☞ live simulation.

- hardness can be controlled by $\beta \uparrow$ ($\beta = 1$ is easy)

- HGBat is even harder than HappyCat
- there are only a few attempts to design EAs to solve HappyCat and HGBat
- one attempt that also addressed HGBat is the μDER proposed by Olguin-Carbajal et al. in IEEE Access yielding performance curves:

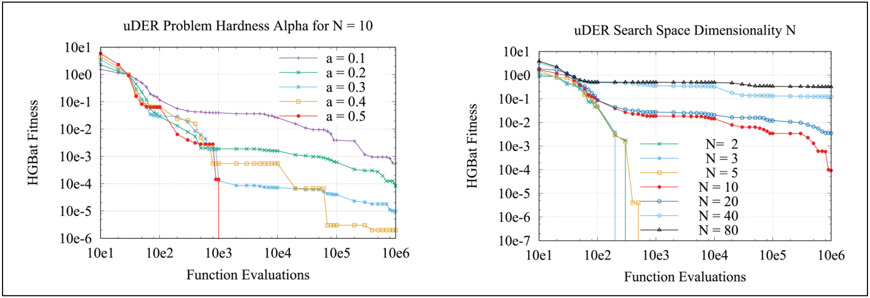

Figure 3.3: Performance of μDER on HGBat Olguin-Carbajal et al.. Note, it holds $N = D$ and $a = 1/\beta$. Left: Influence of the hardness parameter $\beta = 1/a$. Right: Influence of the search space dimensionality for $\beta = 8$.

The ThreeButton Function:

$$F_{\mathrm{TB}}(\mathbf{y}) := \sqrt[\beta]{\left(\|\mathbf{y}\|^2 - D\right)^2 \left(\|\mathbf{y} - \mathbf{2}\|^2 - D\right)^2 \left(\|\mathbf{y} - \mathbf{4}\|^2 - D\right)^2} + \frac{\gamma}{D}\left(\frac{\|\mathbf{y}\|^2}{2} + \sum_{i=1}^{D} y_i\right) + \frac{\gamma}{2} \qquad \text{with } \beta = 8, \gamma = 1 \qquad (3.3)$$

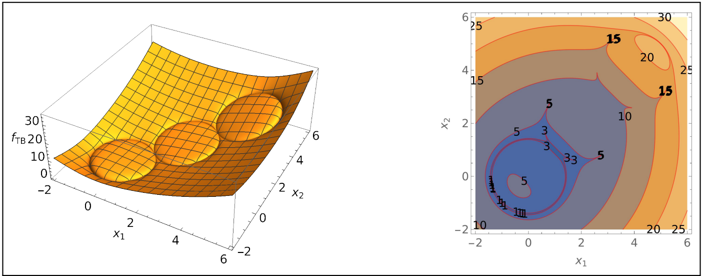

Figure 3.4: $F_{\mathrm{TB}}(\mathbf{y})$ for $D = 2$; optimizer at: $\hat{\mathbf{y}} = (-1, \ldots, -1)^{\mathrm{T}}$ ☞ live simulation.

- hardness can be controlled by $\beta \uparrow$ and $\gamma \downarrow$
- even the $D = 10$ case with $\beta = 1$ seems a challenge for IPOP-CMA-ES
- generalization to M-Button functions is trivial

Optimizing Audi:

$$F_{\mathrm{Au}}(\mathbf{y}) := \sqrt[\beta]{(\|\mathbf{y}\|^2 - D)^2(\|\mathbf{y} - \mathbf{1.3}\|^2 - D)^2(\|\mathbf{y} - \mathbf{2.6}\|^2 - D)^2(\|\mathbf{y} - \mathbf{3.9}\|^2 - D)^2} + \frac{\gamma}{D}\left(\frac{\|\mathbf{y}\|^2}{2} + \sum_{i=1}^{D} y_i\right) + \frac{\gamma}{2} \qquad \text{with } \beta = 8, \gamma = 1 \tag{3.4}$$

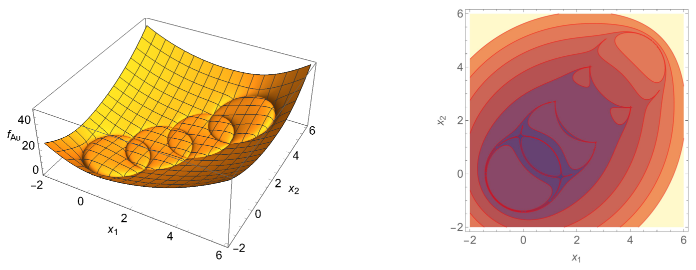

Figure 3.5: Audi 2D; global minimum at: $\hat{\mathbf{y}} = (-1, \ldots, -1)^{\mathrm{T}}$ ☞ live simulation.

- NB: there are local minima depending on β and γ
- even in $D = 2$ with $\beta = 1$, a large γ (about 60) is needed to get the IPOP-CMA-ES to converge to the global minimizer

The Thomson-Function

- calculate the minimum electrostatic energy configuration of P equally charged particles Q on a sphere (of radius 1)
- this is known as Thomson's problem
- $W_{\mathrm{el}}(\mathbf{x}_1, \ldots, \mathbf{x}_P) = \frac{Q^2}{4\pi\epsilon} \sum_{i=1}^{P-1} \sum_{j=i+1}^{P} \|\mathbf{x}_i - \mathbf{x}_k\|^{-1}$ s.t. $\forall k, i : \|\mathbf{x}\| = 1$
- since $\mathbf{x} \in \mathbb{R}^3$, the decision variable vector $\mathbf{y} \in \mathbb{R}^{3P}$ is 3P-dimensional, the Thomson function reads

$$F_{\mathrm{Th}}(\mathbf{y}) := \sum_{p=0}^{P-2} \sum_{k=p+1}^{P-1} \left[\sum_{d=1}^{3} \left(\frac{y_{3p+d}}{\sqrt{\sum_{i=1}^{3} y_{3p+i}^2}} - \frac{y_{3k+d}}{\sqrt{\sum_{i=1}^{3} y_{3k+i}^2}} \right)^2 \right]^{-\frac{\alpha}{2}} - F_{\mathrm{Th}}^*(P) \tag{3.5}$$

  where $\alpha = 1$ and $F_{\mathrm{Th}}^*(P)$ are the global minima know from literature
- $F_{\mathrm{Th}}^*(P)$ are available to at least $P = 200$ charges

☞ could be used in large-scale BBOB

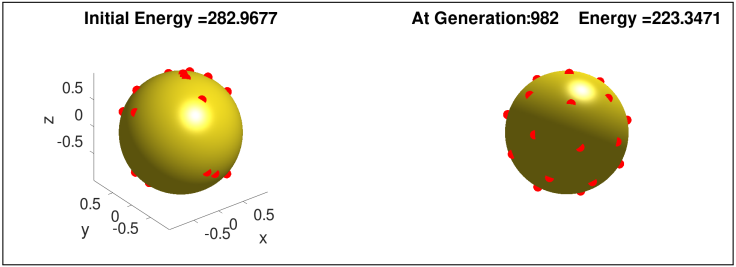

Figure 3.6: Initial random configuration (left) vs. final minimized energy configuration for $P = 24$ charges. A $(8/8, 16)$-MA-ES has been used to find the minimum configuration.

- closely related: **Tammes' problem** of finding the maximum of the minimal distance of P points on the unit sphere ($\alpha = \infty$ in (3.5))
- this is equivalent to minimize

$$F_{\mathrm{Ta}}(\mathbf{y}) := \min_{p,k\in\{1,\ldots,P\}} \left[-\sqrt{\sum_{d=1}^{3} \left(\frac{y_{3p+d}}{\sqrt{\sum_{i=1}^{3} y_{3p+i}^2}} - \frac{y_{3k+d}}{\sqrt{\sum_{i=1}^{3} y_{3k+i}^2}} \right)^2} \right] - F_{\mathrm{Ta}}^*(P) \tag{3.6}$$

The Lennard-Jones Energy Function

- minimal energy configuration of P atoms of the same type
- interaction energy is given by the Lennard-Jones potential
  $V(r) := 4\varepsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]$
- using reduced units $\varepsilon = 1$ and $\sigma = 1$, the energy function of P atoms is given by

$$F_{\mathrm{LJ}}(\mathbf{y}) := 4\sum_{p=0}^{P-2}\sum_{k=p+1}^{P-1}\left[\left(\sum_{d=1}^{3}(y_{3p+d} - y_{3k+d})^2\right)^{-6} - \left(\sum_{d=1}^{3}(y_{3p+d} - y_{3k+d})^2\right)^{-3}\right] - F^*_{\mathrm{LJ}}(P) \qquad (3.7)$$

- global optima are known for 100+ atoms (see Wales & Doye, 1997)
- for several P – especially for larger P – it appears hard for CMA-ES (and other general purpose EAs) to find the global optimizer

Summary: COCO BBOB Needs Renovations

- there are several weaknesses in the COCO BBOB design
- some of them are difficult to remove, however,
  a renovation regarding the test functions can be done easily
- Patrick Spettel has done such a renovation of BBOB in context of constrained optimization
- the coding of a problem problem class has not taken more than two days
- the code can be found on github: Extension of the Comparing Continuous Optimizers (COCO) framework
- doing the same for the unconstrained case should be even easier
- **the main problem is dissemination and application in the community**

(COCO Extension)

Thank You

(pdf of this talk)