

Performance of modular CMA-ES on strict box constrained problems in the SBOX-COST benchmarking suite

Diederick Vermetten
LIACS, Leiden University
Leiden, The Netherlands
d.l.vermetten@liacs.leidenuniv.nl

Olaf Mersmann
TH Köln
Köln, Germany
olaf.mersmann@th-koeln.de

Manuel López-Ibáñez
University of Manchester
Manchester, UK
manuel.lopez-ibanez@manchester.ac.uk

Youngmin Kim
University of Manchester
Manchester, UK
youngmin.kim@manchester.ac.uk

Richard Allmendinger
University of Manchester
Manchester, UK
richard.allmendinger@manchester.ac.uk

Anna V. Kononova
LIACS, Leiden University
Leiden, The Netherlands
a.kononova@liacs.leidenuniv.nl

ABSTRACT

This paper evaluates several versions of CMA-ES on the strict-box-constrained benchmarking suite (SBOX-COST). SBOX-COST is a variant of the well-known BBOB benchmark suite that enforces box-constraints by returning an invalid evaluation value for solutions violating any of the constraints on the bounds of decision variables. We compare four variants of CMA-ES: two initialization methods (centre of mass near the origin versus uniformly at random) and two box-constraint handling methods (saturate versus no handling at all). Contrary to what may be expected, handling box-constraints by saturation is not always better than not handling them at all. However, over all BBOB functions, saturation is better than not handling and the difference increases with the number of dimensions. Enforcing sbox-constraints also has a clear negative effect on the performance of classical CMA-ES (with uniform random initialization and no constraint handling), especially as problem dimensionality increases.

CCS CONCEPTS

• **Theory of computation** → **Theory of randomized search heuristics**; **Algorithm design techniques**; **Discrete optimization**; • **Mathematics of computing** → *Evolutionary algorithms*.

KEYWORDS

keyword, keyword

ACM Reference Format:

Diederick Vermetten, Olaf Mersmann, Manuel López-Ibáñez, Youngmin Kim, Richard Allmendinger, and Anna V. Kononova. 2023. Performance of modular CMA-ES on strict box constrained problems in the SBOX-COST benchmarking suite. In *2023 Genetic and Evolutionary Computation Conference (GECCO '23)*, July 10–14, 2023, Lisbon, Portugal. ACM, New York, NY, USA, 4 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

Box-constraints impose limits on the domain of decision variables and are perhaps the most typical type of constraints that arise in black-box continuous optimization. In almost all real-world problems, the range of decision variables will be limited by physical,

design, resource or policy bounds that are known a priori. As a result of these bounds, solutions outside those bounds, i.e., violating the box-constraints, are not only unacceptable, but they may not have a defined value of the objective function. Unfortunately, many optimization algorithms do not follow this assumption and, instead, allow the evaluation of solutions violating box-constraints. Moreover, benchmarking suites used for comparing algorithms, such as COCO/BBOB [4], return a valid objective value for such solutions, which is used by the algorithms to steer their search.

Here, we consider a benchmark SBOX-COST that enforces box-constraints by returning the same invalid value (∞) for any infeasible solution, thus the algorithm cannot use infeasible solutions to inform the search. We evaluate the effect that such strict-box-constraints have on the performance of some variants the Covariance Matrix Adaptation Evolution Strategy (CMA-ES).

The Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [5] is a very popular heuristic optimisation algorithm for continuous optimisation problems [7]. CMA-ES is considered state-of-the-art in evolutionary computation and has been adopted as one of the standard tools for continuous optimisation in many research labs. There are many flavours and variants of CMA-ES developed through the years and different implementations of sub-components such as the sampling strategy and the boundary correction method.

In recent works, the different modules and configurations of CMA-ES are explored and analysed based on their performance [1]. In that research a modular CMA-ES framework is presented, representing a plethora of different CMA-ES configurations. In this paper, the modular CMA-ES framework is used to analyse the effect of introducing boundaries to standard BBOB problems, as done in the new SBOX-COST benchmark.

This paper serves as an *example* of a short paper for the ACM Workshop on Strict Box-Constrained Optimization¹ (SBOX-COST) within GECCO 2023.

2 METHODOLOGY

2.1 SBOX-COST benchmarking suite

For our benchmarking, we make use of the newly proposed Strict Box-Constrained Optimization Studies benchmark² (SBOX-COST),

GECCO '23, July 10–14, 2023, Lisbon, Portugal
2023. ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$15.00
<https://doi.org/10.1145/nnnnnnn.nnnnnnn>

¹<https://sbox-cost.github.io/>
²<https://github.com/sbox-cost>

which is a modification of the original BBOB suite [4]. The changes made are as follows:

- *hard box-constraints*: points evaluated outside of the domain $[-5, 5]^d$ are considered infeasible and evaluated to ∞ ;
- Location of the optima: SBOX-COST has optima across instances located uniformly within the full domain – except for, as in BBOB, F4, F8, F9, F19, F20, F24 which have optima uniform in $[-4, 4]^d$ and F5 - in some corners of $[-5, 5]^d$.

2.2 Modular CMA-ES

To investigate several commonly and less commonly used configurations and variants of the CMA-ES algorithm [5], we use the Modular CMA-ES framework [1, 6]. The framework is open-source and available³ as part of the IOHprofiler [3] environment.

Modular CMA-ES framework is currently made up of 11 modules, each having a number of implemented options: Active update (2), Elitism (2), Orthogonal Sampling (2), Sequential Selection (2), Threshold Convergence (2), Step-Size adaptation (7), Mirrored Sampling (3), Quasi-Gaussian Sampling (3), Recombination Weights (2), Restart Strategy (3) and Boundary Correction (6). The framework allows a total of 72.576 possible configurations. A detailed description of each module and their settings can be found in [1].

In this paper, to investigate the *effect of the introduction of strict boundaries* on the BBOB problems in the SBOX-COST benchmarking suite, we compare the behaviour of 4 different variants withing modular CMA-ES by considering:

- two *initialization* methods for the centre of mass (set to the origin of the space, centre, or set uniformly at random in the domain),
- enabling or disabling the method for dealing with *box-constraints* (saturate, which places the infeasible coordinates to the closest corresponding bound or none which allows infeasible points).

All *other modules* and parameter settings are set to their defaults specified in [1].

3 EXPERIMENTAL SETUP

We evaluate algorithms on the same 24 functions of both BBOB and SBOX-COST. The only difference between a function in both benchmark suites is the objective function value returned for solutions violating a box constraint. We use an identical setup for BBOB and SBOX-COST: 15 instances per function, 1 run per instance, dimensionality $d \in \{5, 20, 40\}$, budget $10000 \times d$.

Experiments reported in this study are carried out in the IOHexperimenter environment [2], which implements both sets of benchmarking functions.

4 RESULTS

The data from the described experiments is visualized using IOHAnalyzer [8]. We start by aggregating the performance of each of the 8 settings across all 5-dimensional problems. This is done by making use of the Empirical Cumulative Distribution function, which aggregates the fraction of targets (we use the default of 51 logarithmically spaced targets $\{10^{-8} \dots 10^2\}$) hit in each run on

³<https://github.com/IOHprofiler/ModularCMAES>

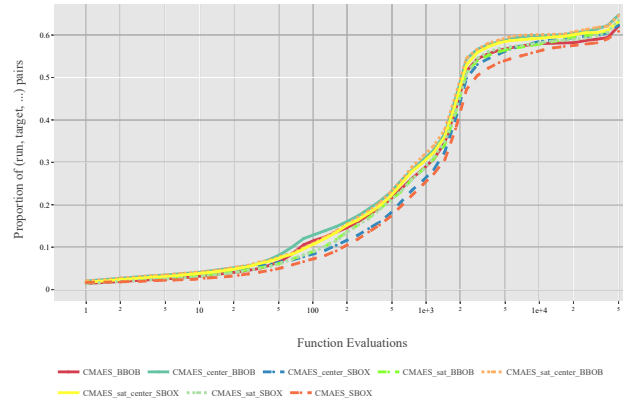


Figure 1: Empirical Cumulative Distribution Function for all considered variants within modular CMA-ES, aggregated over the 24 functions of BBOB and SBOX-COST in 5D. Targets are 51 log-spaces values between 10^2 and 10^{-8} ('bbob' default in IOHalyzer).

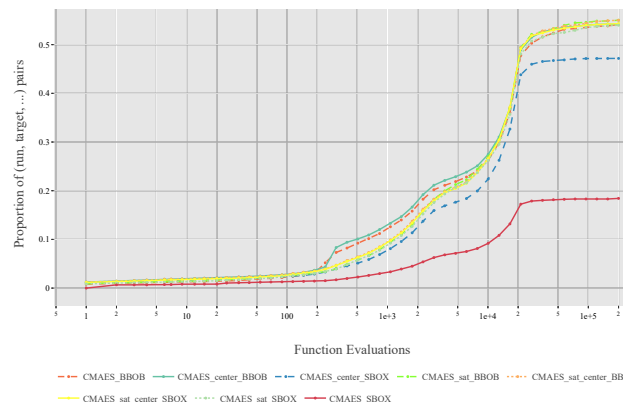


Figure 2: Empirical Cumulative Distribution Function for all considered variants within modular CMA-ES, aggregated over the 24 functions of BBOB and SBOX-COST in 20D. Targets are 51 log-spaces values between 10^2 and 10^{-8} ('bbob' default in IOHalyzer).

each function, as shown in Figure 1. From this figure, we notice that the variants of CMA-ES without boundary correction run on the SBOX-suite perform slightly worse on aggregate than the other versions.

In Figure 2, we show the ECDF curves on the 20-dimensional problems. Here, the poor performance of the CMA-ES with random initialization without boundary correction on SBOX is much more obvious. This is likely caused by the fact that in higher dimensions, a Gaussian distribution around a random point will be much more likely to generate points outside the domain, leading to many wasted function evaluations, and potentially a disrupted search process. While initializing the CMA-ES in the center of the domain alleviates this problem somewhat, it still performs worse than the versions which add a boundary handling mechanism.

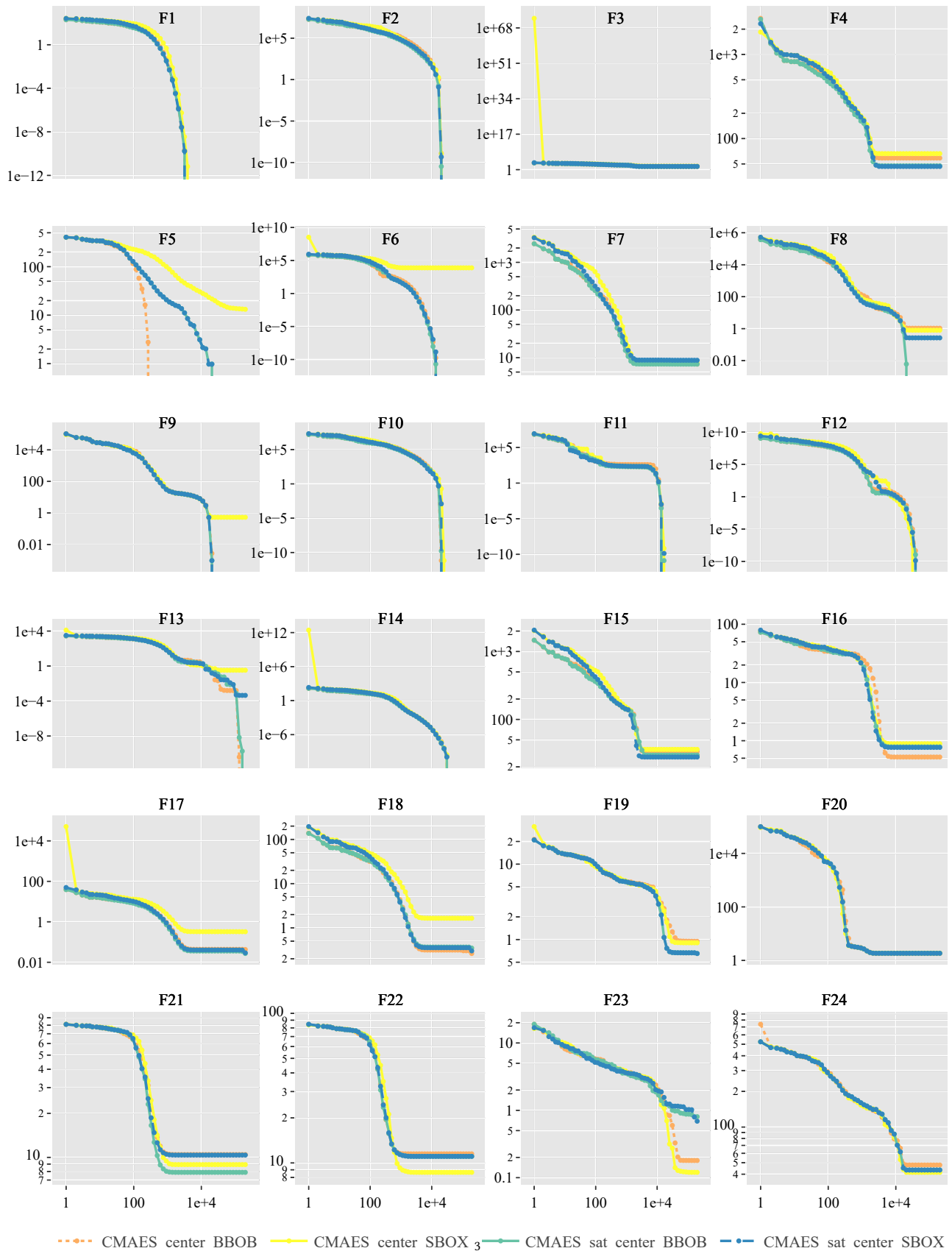


Figure 3: Mean function value over time for the CMA-ES methods with centre initialization, for 24 SBOX-COST functions of BBOB and SBOX-COST in 20D.

	DIM	funcId	ID	runtime	runs	mean	median	sd	2%	5%	10%	25%	50%	75%	90%	95%	98%	
349	1	20	5	CMAES_center_BBOB	10000	15	0	0	0	0	0	0	0	0	0	0	0	
350	2	20	5	CMAES_center_SBOX	10000	15	29.58	32.94	9.15	11.55	11.56	14.44	26.25	32.94	34.90	37.68	38.80	39.99
351	3	20	6	CMAES_center_BBOB	10000	15	0	0	0	0	0	0	0	0	0	0	0	
352	4	20	6	CMAES_center_SBOX	10000	15	7735.90	0	29961.02	0	0	0	0	0	0	34811.56	83547.75	
353	5	20	7	CMAES_center_BBOB	10000	15	7.56	7.42	3.59	1.58	2.11	3.43	5.73	7.42	8.93	11.74	13.80	14.59
354	6	20	7	CMAES_center_SBOX	10000	15	7.66	8.65	4.07	2.26	2.53	2.92	4.45	8.65	10.19	12.02	13.64	15.07
355	7	20	14	CMAES_center_BBOB	10000	15	0	0	0	0	0	0	0	0	0	0	0	0
356	8	20	14	CMAES_center_SBOX	10000	15	0	0	0	0	0	0	0	0	0	0	0	0
357	9	20	15	CMAES_center_BBOB	10000	15	33.70	32.83	11.75	14.61	17.11	20.10	25.87	32.83	42.78	47.16	50.54	53.05
358	10	20	15	CMAES_center_SBOX	10000	15	36.17	33.83	11.41	22.17	22.59	24.48	27.36	33.83	43.28	53.47	54.49	55.22
359	11	20	16	CMAES_center_BBOB	10000	15	0.52	0.39	0.44	0.10	0.11	0.12	0.17	0.39	0.73	1.10	1.38	1.51
360	12	20	16	CMAES_center_SBOX	10000	15	0.89	0.84	0.60	0.09	0.10	0.16	0.32	0.84	1.39	1.57	1.73	1.87
361	13	20	17	CMAES_center_BBOB	10000	15	0.04	0.03	0.04	0	0	0	0.02	0.03	0.06	0.09	0.12	0.15
362	14	20	17	CMAES_center_SBOX	10000	15	0.33	0.08	0.67	0	0	0.01	0.02	0.08	0.25	0.76	1.46	2.12
363	15	20	18	CMAES_center_BBOB	10000	15	0.31	0.25	0.23	0.07	0.07	0.09	0.17	0.25	0.36	0.70	0.73	0.75
364	16	20	18	CMAES_center_SBOX	10000	15	1.65	1.58	1.18	0.33	0.34	0.43	0.73	1.58	2.34	2.84	3.38	4.11
365	17	20	21	CMAES_center_BBOB	10000	15	10.45	7.59	8.87	0	0	0.63	5.15	7.59	16.40	21.19	25.97	28.86
366	18	20	21	CMAES_center_SBOX	10000	15	8.92	7.59	6.71	1.06	1.61	2.02	3.21	7.59	12.34	17.11	19.15	22.01
367	19	20	22	CMAES_center_BBOB	10000	15	11.50	7.27	14.07	1.05	1.58	1.95	3.51	7.27	14.58	20.38	32.28	46.99
368	20	20	22	CMAES_center_SBOX	10000	15	8.65	5.06	7.93	1.05	1.58	1.95	2.59	5.06	12.86	20.38	22.89	24.44
369	21	20	23	CMAES_center_BBOB	10000	15	2.24	2.62	1.14	0.15	0.19	0.38	1.71	2.62	3.05	3.36	3.39	3.41
370	22	20	23	CMAES_center_SBOX	10000	15	1.82	2.49	1.41	0.07	0.07	0.09	0.12	2.49	2.99	3.22	3.32	3.44

Table 1: Summary statistics of function value reached after 10 000 evaluations for the CMA-ES with center initialization and no boundary correction on a selected set of 20-dimensional problems.

To get a more detailed view of the performance on individual functions, we plot the convergence trajectories (mean function value over time) in Figure 3. To ease readability, we only show the CMA-ES variants which are initialized in the center of the domain. From this figure, we can see that for several functions, the performance differences between the algorithms run on SBOX and on BBOB are rather small. In particular, this is the case for the functions {4, 8, 9, 19, 20, 24}, where the instance generation procedure was not modified from the original BBOB suite. However, the differences on function 5 clearly show the impact of the hard box-constraint, with the CMA-ES on BBOB easily solving the problem (achieved by moving outside the domain).

Finally, for a few selected functions, we show some detailed summary statistics in Table 1, which confirms the observations from Figure 3: for many problems, the CMA-ES can overcome the addition of hard box-constraints and changed function initialization, although there are some problems where the changes lead to a noticeable deterioration in performance.⁴

5 CONCLUSIONS

In this paper we have benchmarked four variants of CMA-ES on SBOX-COST, which is a variant of the BBOB suite that enforces box-constraints. Our results clearly show that enforcing box-constraints has a negative effect on the performance of classical CMA-ES, specially with higher dimensionality. Although not handling box-constraints sometimes performs worse than the constraint handling method evaluated here (saturation), over all BBOB functions, handling box-constraints is clearly better than ignoring them when the box-constraints are strict.

REPRODUCIBILITY

To ensure the reproducibility of results presented in this paper, the complete code used for experiments and runs data in the IOH format have been uploaded to a Zenodo repository.⁵

⁴Note that the discussion in this paper is purely for illustration purposes. In a submitted paper, this should be extended.

⁵<https://doi.org/10.5281/zenodo.7649077>

REFERENCES

- Jacob de Nobel, Diederick Vermetten, Hao Wang, Carola Doerr, and Thomas Bäck. 2021. Tuning as a means of assessing the benefits of new ideas in interplay with existing algorithmic modules. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1375–1384.
- Jacob de Nobel, Furong Ye, Diederick Vermetten, Hao Wang, Carola Doerr, and Thomas Bäck. 2021. IOHexperimenter: Benchmarking Platform for Iterative Optimization Heuristics. *arXiv preprint arXiv:2111.04077* (2021).
- Carola Doerr, Hao Wang, Furong Ye, Sander van Rijn, and Thomas Bäck. 2018. IOHprofiler: A benchmarking and profiling tool for iterative optimization heuristics. *arXiv preprint arXiv:1810.05281* (2018).
- Nikolaus Hansen, Anne Auger, Raymond Ros, Olaf Mersmann, Tea Tušar, and Dimo Brockhoff. 2020. COCO: A platform for comparing continuous optimizers in a black-box setting. *Optimization Methods and Software* (2020), 1–31.
- Nikolaus Hansen and Andreas Ostermeier. 1996. Adapting Arbitrary Normal Mutation Distributions in Evolution Strategies: The Covariance Matrix Adaptation. In *Proceedings of the IEEE International Conference on Evolutionary Computation*. 312–317. <https://doi.org/10.1109/ICEC.1996.542381>
- Sander van Rijn, Hao Wang, Bas van Stein, and Thomas Bäck. 2017. Algorithm configuration data mining for cma evolution strategies. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 737–744.
- Diederick Vermetten, Fabio Caraffini, Bas van Stein, and Anna V. Kononova. 2022. Using Structural Bias to Analyse the Behaviour of Modular CMA-ES. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion* (Boston, Massachusetts) (*GECCO '22*). Association for Computing Machinery, New York, NY, USA, 1674–1682. <https://doi.org/10.1145/3520304.3534035>
- Hao Wang, Diederick Vermetten, Furong Ye, Carola Doerr, and Thomas Bäck. 2022. IOHalyzer: Detailed Performance Analyses for Iterative Optimization Heuristics. *ACM Trans. Evol. Learn. Optim.* 2, 1 (2022), 3:1–3:29. <https://doi.org/10.1145/3510426>