

DMS and MultiGLODS: Black-Box Optimization Benchmarking of Two Direct Search Methods on the bbob-biobj Test Suite

Dimo Brockhoff, Baptiste Plaquevent-Jourdain, Anne Auger, Nikolaus Hansen

▶ **To cite this version:**

Dimo Brockhoff, Baptiste Plaquevent-Jourdain, Anne Auger, Nikolaus Hansen. DMS and MultiGLODS: Black-Box Optimization Benchmarking of Two Direct Search Methods on the bbob-biobj Test Suite. GECCO 2021 Companion - Genetic and Evolutionary Computation Conference Companion, Jul 2021, Lille / Virtual, France. pp.8, ⟨10.1145/3449726.3463207⟩. ⟨hal-03284476⟩

HAL Id: hal-03284476

https://inria.hal.science/hal-03284476v1

Submitted on 12 Jul 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC-SA 4.0 - Attribution - Non-commercial use - ShareAlike - International License

DMS and MultiGLODS: Black-Box Optimization Benchmarking of Two Direct Search Methods on the bbob-biobj Test Suite

Dimo Brockhoff
Inria, Randopt team
CMAP, CNRS, Ecole Polytechnique, IP Paris
Palaiseau, France
firstname.lastname@inria.com

Baptiste Plaquevent-Jourdain
ENSTA Paris
Palaiseau, France
firstname.last-name@ensta-paris.fr

Anne Auger
Inria, Randopt team
CMAP, CNRS, Ecole Polytechnique, IP Paris
Palaiseau, France
firstname.lastname@inria.com

Nikolaus Hansen
Inria, Randopt team
CMAP, CNRS, Ecole Polytechnique, IP Paris
Palaiseau, France
firstname.lastname@inria.com

ABSTRACT

Direct Multisearch (DMS) and MultiGLODS are two derivative-free solvers for approximating the entire set of Pareto-optimal solutions of a multiobjective (blackbox) problem. They both follow the search/poll step approach of direct search methods, employ Pareto dominance to avoid aggregating objectives, and have theoretical limit guarantees. Although the original publications already compare the two algorithms empirically with a variety of multiobjective solvers, an analysis on their scaling behavior with dimension was missing. Here, we run the publicly available implementations on the bbob-biobj test suite of the COCO platform and by investigating their performances in more detail, observe (i) a small defect in the default initialization of DMS, (ii) for both algorithms a decrease in relative performance to other algorithms of the original studies (even matching the performance of random search for MultiGLODS in higher dimension), and (iii) consequently, an under-performance to previously untested stochastic solvers from the evolutionary computation field, especially when the dimension is higher.

CCS CONCEPTS

• **Computing methodologies** → **Continuous space search**.

KEYWORDS

Benchmarking, Black-box optimization, Bi-objective optimization

ACM Reference Format:
Dimo Brockhoff, Baptiste Plaquevent-Jourdain, Anne Auger, and Nikolaus Hansen. 2021. DMS and MultiGLODS: Black-Box Optimization Benchmarking of Two Direct Search Methods on the bbob-biobj Test Suite. In *2021 Genetic and Evolutionary Computation Conference Companion (GECCO '21 Companion), July 10–14, 2021, Lille, France.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3449726.3463207

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
GECCO '21 Companion, July 10–14, 2021, Lille, France
© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8351-6/21/07.
https://doi.org/10.1145/3449726.3463207

1 INTRODUCTION

Besides evolutionary computation approaches, derivative-free optimization methods and direct search methods in particular are another important source of algorithms which are often employed in practice to solve blackbox optimization problems. Direct search itself is a generic technique in which local poll and more global search steps are performed on the current iterate and with respect to a certain step size that shrinks as soon as no progress is observed [4].

One prominent example of a *multiobjective* direct search algorithm is Direct Multisearch (DMS, [4]). DMS employs the search and poll step of single-objective direct search methods and keeps an archive of non-dominated solutions to draw the new iterates/poll centers from. MultiGLODS is another variant of multiobjective direct search, using multistarts to address multimodal problems [3]. Besides proving theoretical convergence properties, the original publications compared the algorithms numerically with a variety of other multiobjective solvers on several artificial and real-world problems. It turned out that DMS was outperforming classical stochastic algorithms like NSGA-II as well as other multiobjective direct search algorithms like MADS [4]. In [3], it was shown empirically that MultiGLODS improves the performance of DMS on some artificial multimodal functions.

The problems in the original publications, however, had rather low problem dimension, n, and the scaling behavior of the algorithms with increasing n was not explicitly tested. This paper investigates this scaling behavior on the bbob-biobj test suite [11] of the COCO platform [7]. To this end, we use the source codes, provided by the authors, and run the default setting except for an improved initialization for DMS (see also Section 3).

Before to discuss the actual experimental settings in Section 4 and the experimental results in Section 5, we briefly discuss a defect in the default implementation of DMS in the following section. Section 6 concludes the paper.

2 ALGORITHM PRESENTATION

Direct multisearch (DMS, [4]) is an extension of direct search to multiobjective optimization. Like direct search methods, it uses a combination of search and poll steps to explore the search space.

To approximate the Pareto set, DMS keeps a list of non-dominated solutions at each iteration with a step size parameter associated to each solution. In each iteration, one solution is chosen from this list and a potential search step produces a list of new solutions from here[1]. If new non-dominated solutions are found, the set of non-dominated solutions is updated accordingly and the following poll step is skipped. Otherwise, the poll step performs evaluations at solutions $\{x_k + \alpha_k d : d \in D_k\}$ around the current solution x_k along certain poll directions d from a positive spanning set of directions D_k in a distance, given by the step size α_k associated to the solution x_k. If no new non-dominated solutions are found, the iteration is considered unsuccessful. In the case of an unsuccessful iteration, the step size is halved. The initial step size, by default, is 1.

MultiGLODS [3] is an extension of DMS to address multimodal problems. Its main differences are the addition of a flag and a distance parameter r_k to each solution in the maintained list of possible current iterates x_k as well as allowing dominated solutions to stay (inactively) in this list. The flag indicates whether a point is active or not and only active solutions can be chosen as current iterate x_k. A newly created solution (in either the search or the poll step) is added to the list as active if it is farther away than r_k to any other solution in the list—independently of whether it is non-dominated or not. If a new solution dominates an active solution from the list, it is also added: as active if no other solution dominates it or as inactive otherwise. All solutions in the list that are dominated by new solutions will become inactive. Solutions, once flagged as inactive can never be activated again.

3 DEFECT IN DEFAULT DMS INITIALIZATION

During initial experiments, we observed that the performance of DMS unexpectedly oscillated with dimension when compared over all functions, see the empirical cumulative distributions of runtime in Figure 1 (the default parameter setting is shown as "DMS-default"). While in dimensions 3 and 5, the algorithm shows good performance from the beginning of the optimization, the performance curves in dimensions 2 and 10 stay flat after the initial search point is evaluated and increase only after 200+ times dimension many function evaluations.

In personal communications with one of the authors, Ana Custódio, we identified the reason for this behavior as the default initialization: the poll set is initialized with n search points, equally distanced on the search space diagonal from the lower bound to the upper bound. For the COCO experiments, we typically use symmetric lower and upper bounds on each variable for the initialization such that only in odd dimension, the search space origin is included in the initial set. As the presence of a solution close to the origin in the initial set of DMS shows the observed positive effect for the odd dimensions, we ensured this also for the even dimensions. Whether we used the search space origin as an additional $(n+1)$th point while keeping the default positions of the other n solutions or whether we used $n+1$ equidistant points on the line between the lower and upper bound did not show a difference when aggregated over all `bbob-biobj` functions (results not shown here). In the remainder of the paper, we use the latter implementation of $n+1$ equidistant points in case the dimension is even and n equidistant points when the dimension is odd. Note in this context that MultiGLODS contains the search space origin in its initial poll set by default.

[1] In the implementation, compared here, no search step is performed

4 EXPERIMENTAL PROCEDURE

For both DMS and MultiGLODS, we use the Matlab implementation, provided by the authors. For DMS, we compare version 0.3 with the updated initialization of the poll set, as described above, to ensure the origin is included in the initial poll set also in even dimension. The parameter `x_ini` was set to the search space origin first and to a randomly chosen point in $[-5, 5]^n$ when a restart happened. The lower and upper bounds were set to -100 and 100 for each variable to create the initial poll set as indicated by COCO functions' `lower_bound` and `upper_bound`. The maximal number of function evaluations per restart is 20,000. All other parameters and algorithm components of DMS were kept as default. For MultiGLODS, we used version 0.1 and the default parameters except for the same maximal number of function evaluations per restart of 20,000 and the same upper and lower bounds of -100 and 100 for the initial poll set.

4.1 Compared Algorithms

DMS and MultiGLODS have been compared with the following five algorithms from the available COCO data sets. NSGA-II, as implemented in the platypus platform [1] is one of the most commonly used stochastic multiobjective algorithms and serves as a baseline, especially because it has been included in the original DMS paper already (but not in the platypus implementation). UP-MO-CMA-ES [9] is a multiobjective CMA-ES variant with unbounded population size, HMO-MO-CMA-ES [10] a hybrid variant of it which runs multiple different algorithms (including BOBYQA on weighted sums in the very beginning) in parallel. The portfolio algorithm HMO-CMA-ES is considered the best performing data set among all 32 currently available `bbob-biobj` data sets of COCO. It follows mostly the performance of the artificial reference algorithm "best 2016" that has been generated by extracting the best runtime for each function and target precision out of 16 algorithms from the BBOB 2016 workshop, see [2] for details. Finally, COMO-100 is a recent multiobjective CMA-ES variant [5] which aims at finding a set of 100 solutions that maximize the hypervolume indicator (among all sets of 100 solutions). A simple random search that uniformly samples the variables between -5 and 5 serves as most basic baseline.

4.2 CPU Timing

In order to evaluate the CPU timing of the algorithms, we have run both DMS and MultiGLODS on the first 10 instances of the entire `bbob-biobj` test suite [11] for $1000n$ function evaluations according to [8]. The Matlab code was run with Matlab, version R2019a on an Intel(R) Xeon(R) CPU E5-2683 v4 2.10GHz with 64 cores. The time per function evaluation of DMS for dimensions 2, 3, 5, 10, and 20 equals 0.427, 0.334, 0.274, 0.258, and 0.285 milliseconds respectively. For MultiGLODS, the corresponding numbers are 0.263, 0.213, 0.166, 0.133, and 0.170 milliseconds per evaluation. The numbers indicate that for both algorithms, the internal costs are rather low and the cost per function evaluation does not grow significantly with

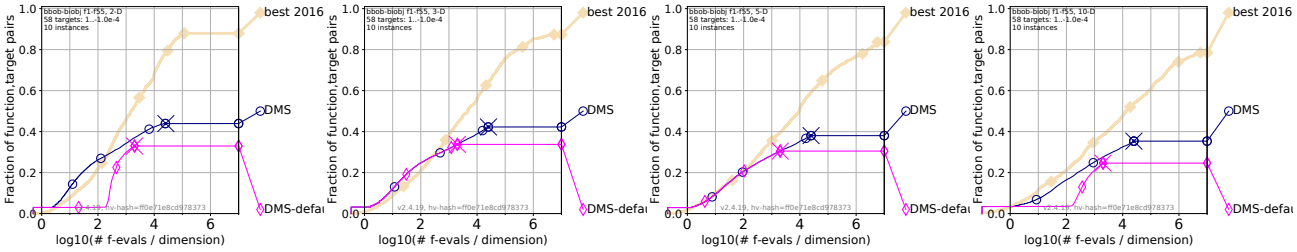


Figure 1: Performance of default DMS initialization ("*DMS-default*") compared to improved initialization with search space origin always contained in the initial solution set of DMS (with $n+1$ equidistant points in even and n equidistant points in odd dimension in the initial poll set, denoted "*DMS*") in different dimensions, aggregated over all 55 bbob-biobj functions.

dimension in our experiments. However, the cost per function evaluation grows with the budget of the algorithms. For DMS, for example, the time per function evaluation in the above experiment is less than half in 20-D with a budget that is 10 times smaller.

5 RESULTS

Results from experiments according to [8], [6] and [2] on the benchmark functions given in [11] are presented in Figures 3, 4, 5 and 6. The experiments were performed with COCO [7], version 2.4, and also the plots were produced with COCO, version 2.4.

Overall Observations. Although both algorithms are direct search variants, their performance on the bbob-biobj test suite is quite different. While DMS shows stable performance over a large variety of functions, even outperforming the best-2016 reference algorithm for small budgets in low dimensions, MultiGLODS is hardly better than random search even in moderate dimensions (Figure 2). For both algorithms, we see a decline in performance with dimension, for MultiGLODS clearly stronger than for the other algorithms in our comparison.

Performance of DMS. Overall the performance of DMS is quite good over a large variety of functions and highly competitive in particular in low dimensions and for small budgets. In 2-D and 3-D and over all functions, DMS outperforms the best-2016 reference for budgets of up to about 200 times dimension, in 5-D, DMS at least matches the best-2016 performance in this low-budget range. In higher dimension and with larger budget, algorithms like NSGA-II and for even higher budgets the CMA-ES variants take the lead. The portfolio algorithm HMO-MO-CMA-ES is better for all budgets in larger dimension.

Exceptional good performance, here for 10-D (Figure 4) can be observed in particular for problems involving the Schwefel (F_{39}, F_{44}, F_{48}) and Rastrigin function (F_{49}) as well as for the double different powers function (F_{41}).

DMS is most challenged by the multimodal functions: in 20-D for example, the multimodal function groups are the only ones where the performance stalls at a very low level (Figures 5 and 6) whereas for the other function groups, not involving an objective function from the multimodal function group, the performance continues to increase. The handling of multimodality has been therefore correctly identified as a crucial improvement in the literature.

Performance of MultiGLODS. Suprisingly, MultiGLODS does not show improvements compared to DMS, especially not on the multimodal functions it is designed for. But also on many other functions, the performance of MultiGLODS is similar or for 17 functions in 10-D even worse than the random search baseline. Exceptions in 10-D are the double sphere (F_1), sphere/separable ellipsoid (F_2), sphere/Rosenbrock (F_4), sphere/Schwefel (F_9), double Rosenbrock (F_{28}), Rosenbrock/Schwefel (F_{33}), and the double Schwefel (F_{53}) functions.

The most difficult function for MultiGLODS is clearly the attractive sector function: whenever it appears as one of the objective functions, the performance is not improving after the first few function evaluations anymore in dimensions 10 and 20. In the lower dimensions (not shown here) the performance gets better over time on these functions, but only in dimensions 2 and 3, the performance of MultiGLODS gets better than random search eventually.

Overall, we can conclude that MultiGLODS is highly affected by dimensionality and competitive only in dimensions 2 and 3 while comparable to random search in the higher dimensions.

6 FINAL RECOMMENDATIONS

The conducted comparison of the two direct search methods DMS and MultiGLODS on the bbob-biobj test suite shows a strong decline in performance with increasing dimension and a performance of MultiGLODS that is comparable with random search. We therefore recommend other algorithms in larger dimension. In low dimension and in particular with a small budget of less than 200 times dimension, DMS is a viable alternative to the more complicated portfolio algorithm HMO-MO-CMA-ES and highly competitive, even outperforming the best-2016 reference in dimensions 2 and 3. If a larger budget is available, algorithms from the evolutionary computation field such as CMA-ES variants take the lead.

ACKNOWLEDGMENTS

We would like to thank Ana Custódio for providing the source code of both algorithms and for her discussions regarding the initialization of DMS.

REFERENCES

[1] Dimo Brockhoff and Tea Tušar. 2019. Benchmarking algorithms from the platypus framework on the biobjective bbob-biobj testbed. In *Genetic and Evolutionary Computation Conference (Companion)*. 1905–1911.

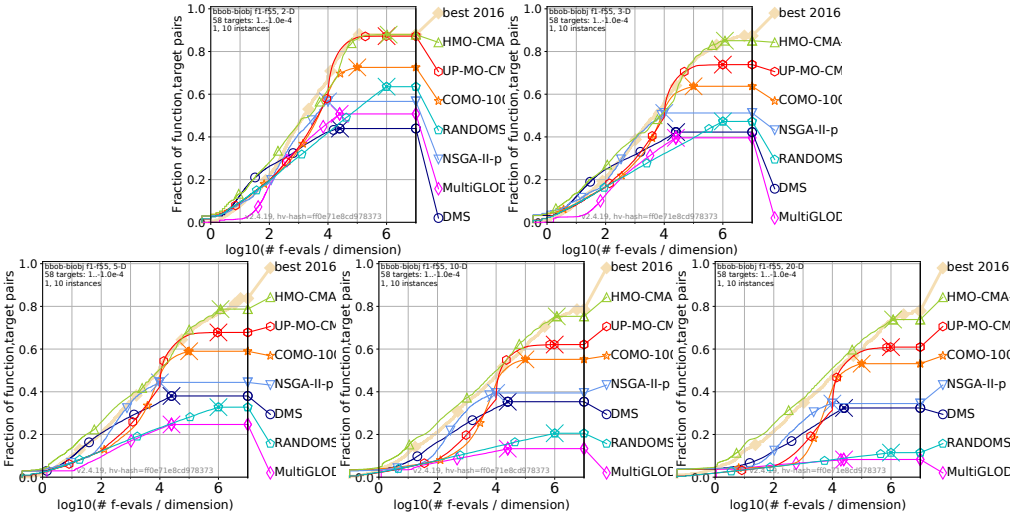


Figure 2: Empirical runtime distributions aggregated over all 55 bbob-biobj functions in dimensions 2, 3, 5, 10, and 20.

[2] D. Brockhoff, T. Tušar, D. Tušar, T. Wagner, N. Hansen, and A. Auger. 2016. Biobjective Performance Assessment with the COCO Platform. *ArXiv e-prints* arXiv:1605.01746 (2016).

[3] A. L. Custódio and J. F. A. Madeira. 2018. MultiGLODS: global and local multiobjective optimization using direct search. *Journal of Global Optimization* 72, 2 (2018), 323–345.

[4] A. L. Custódio, J. F. A. Madeira, A. I. F. Vaz, and L. N. Vicente. 2011. Direct multisearch for multiobjective optimization. *SIAM Journal on Optimization* 21 (2011), 1109–1140.

[5] P. Dufossé and C. Touré. 2019. Benchmarking MO-CMA-ES and COMO-CMA-ES on the Bi-objective bbob-biobj Testbed. In *Genetic and Evolutionary Computation Conference (Companion)*. 1920–1927.

[6] N. Hansen, A Auger, D. Brockhoff, D. Tušar, and T. Tušar. 2016. COCO: Performance Assessment. *ArXiv e-prints* arXiv:1605.03560 (2016).

[7] N. Hansen, A. Auger, R. Ros, O. Mersmann, T. Tušar, and D. Brockhoff. 2021. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *Optimization Methods and Software* 36 (2021), 114–144. Issue 1. DOI: http://dx.doi.org/https://doi.org/10.1080/10556788.2020.1808977

[8] N. Hansen, T. Tušar, O. Mersmann, A. Auger, and D. Brockhoff. 2016. COCO: The Experimental Procedure. *ArXiv e-prints* arXiv:1603.08776 (2016).

[9] O. Krause, T. Glasmachers, N. Hansen, and C. Igel. 2016. Unbounded Population MO-CMA-ES for the Bi-Objective BBOB Test Suite. In *Genetic and Evolutionary Computation Conference (Companion) (GECCO 2016 Companion)*. ACM, New York, NY, USA, 1177–1184. DOI: http://dx.doi.org/10.1145/2908961.2931699

[10] I. Loshchilov and T. Glasmachers. 2016. Anytime Bi-Objective Optimization with a Hybrid Multi-Objective CMA-ES (HMO-CMA-ES). In *Genetic and Evolutionary Computation Conference (Companion) (GECCO 2016 Companion)*. ACM, New York, NY, USA, 1169–1176. DOI: http://dx.doi.org/10.1145/2908961.2931698

[11] T. Tušar, D. Brockhoff, N. Hansen, and A. Auger. 2016. COCO: The Bi-objective Black-Box Optimization Benchmarking (bbob-biobj) Test Suite. *ArXiv e-prints* arXiv:1604.00359 (2016).

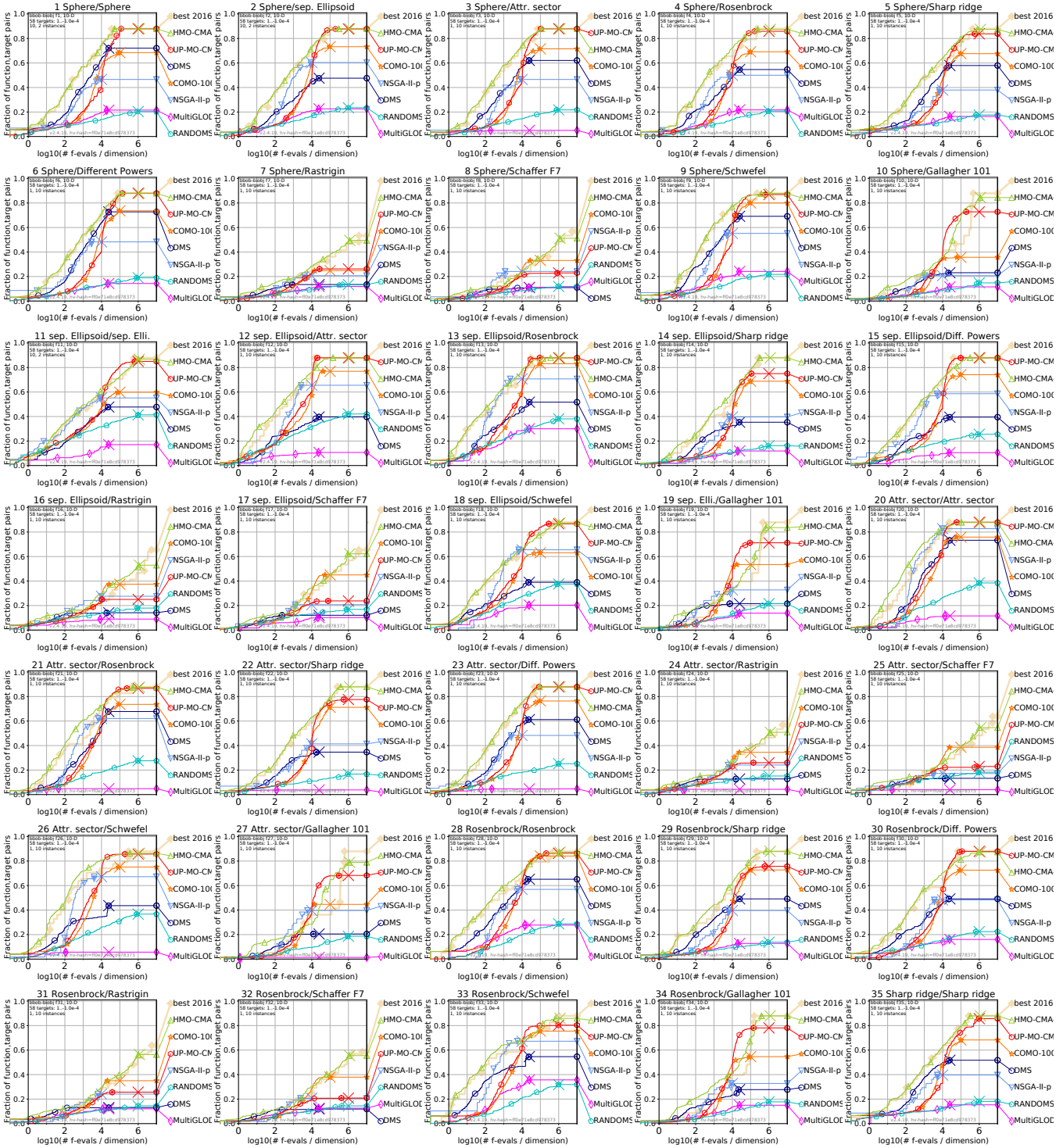


Figure 3: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/DIM) for the 58 targets $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ in dimension 10.

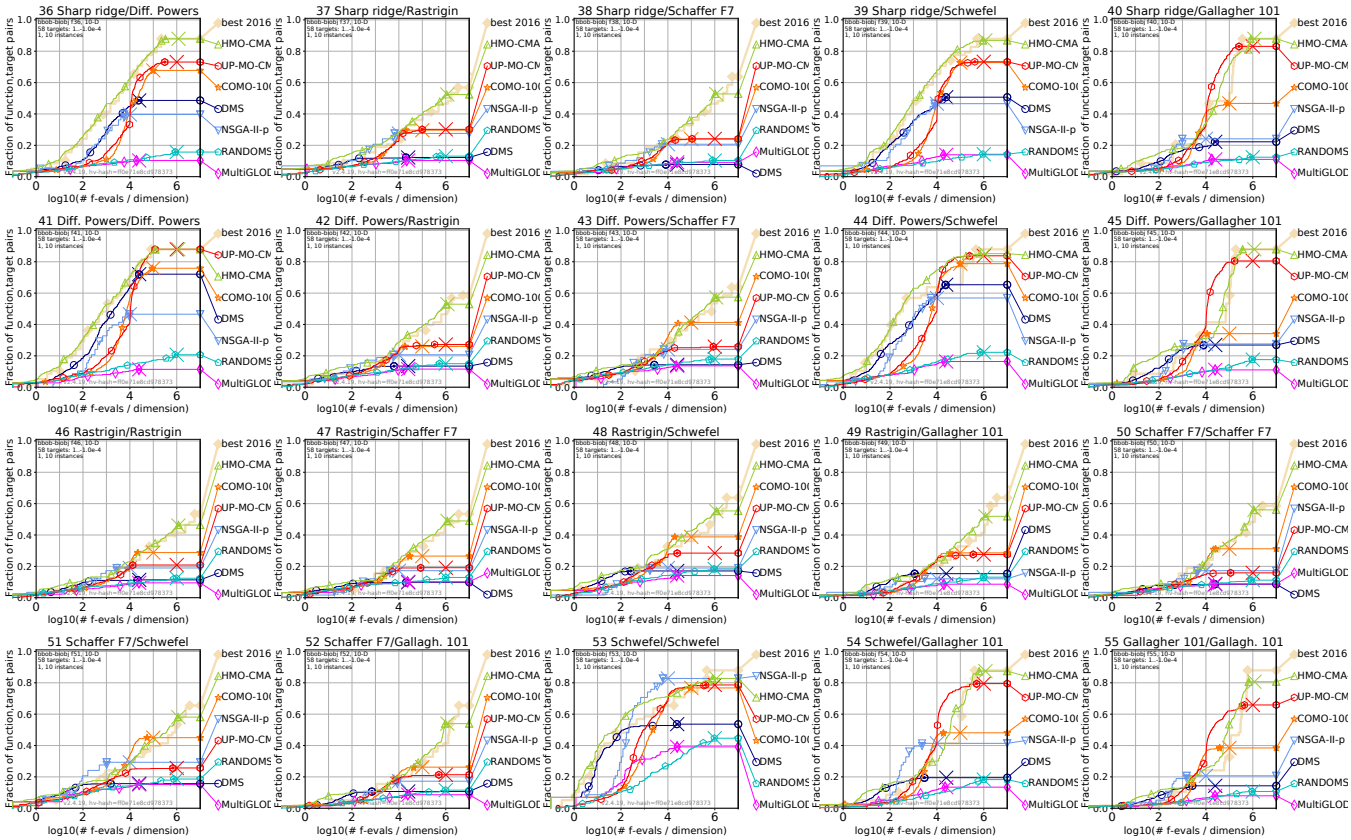


Figure 4: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) as in Fig. 3 but for functions f_{36} to f_{55} in 10-D.

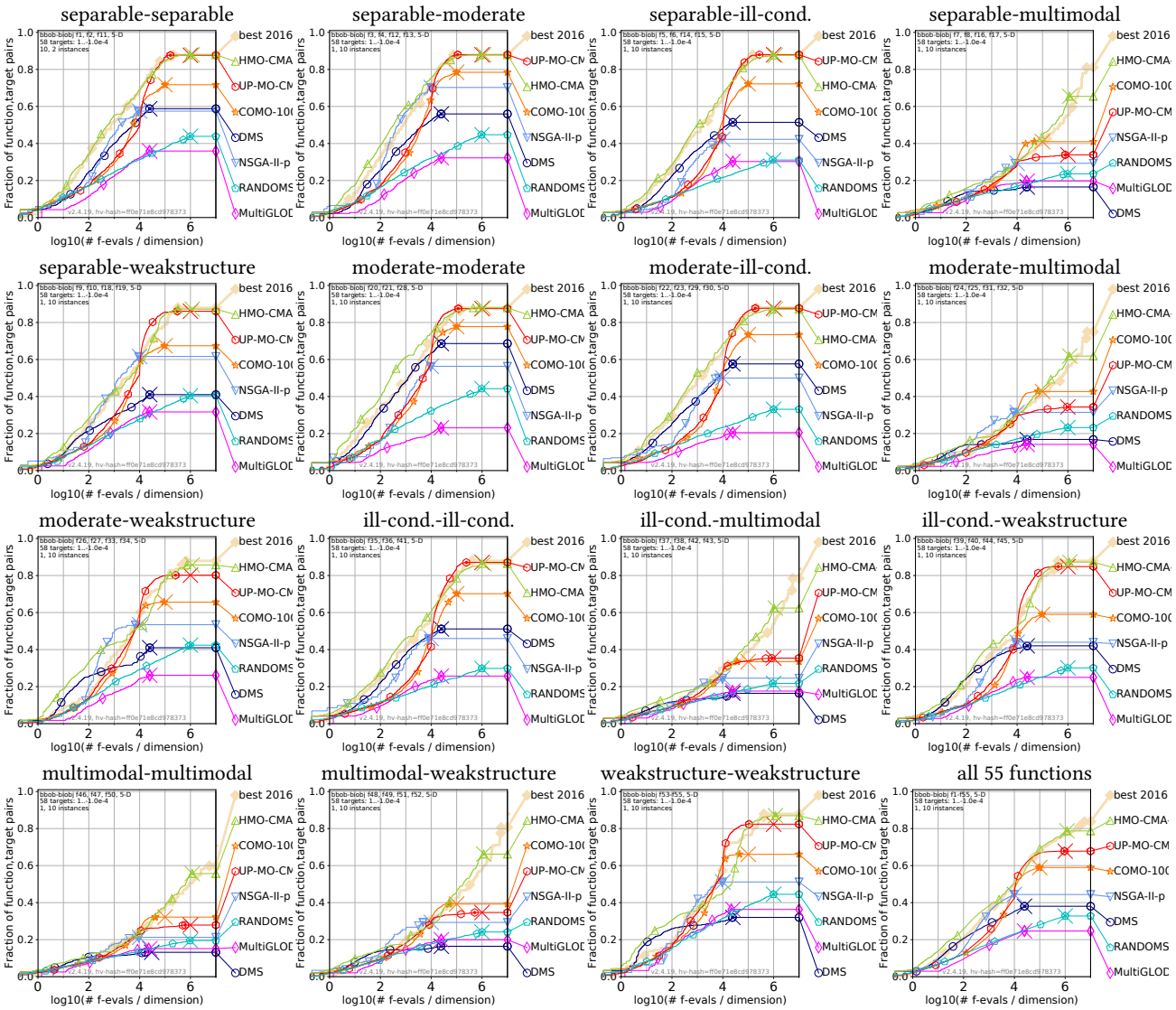


Figure 5: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ for all functions and subgroups in 5-D. As reference algorithm, the best algorithm from BBOB 2016 is shown as light thick line with diamond markers.

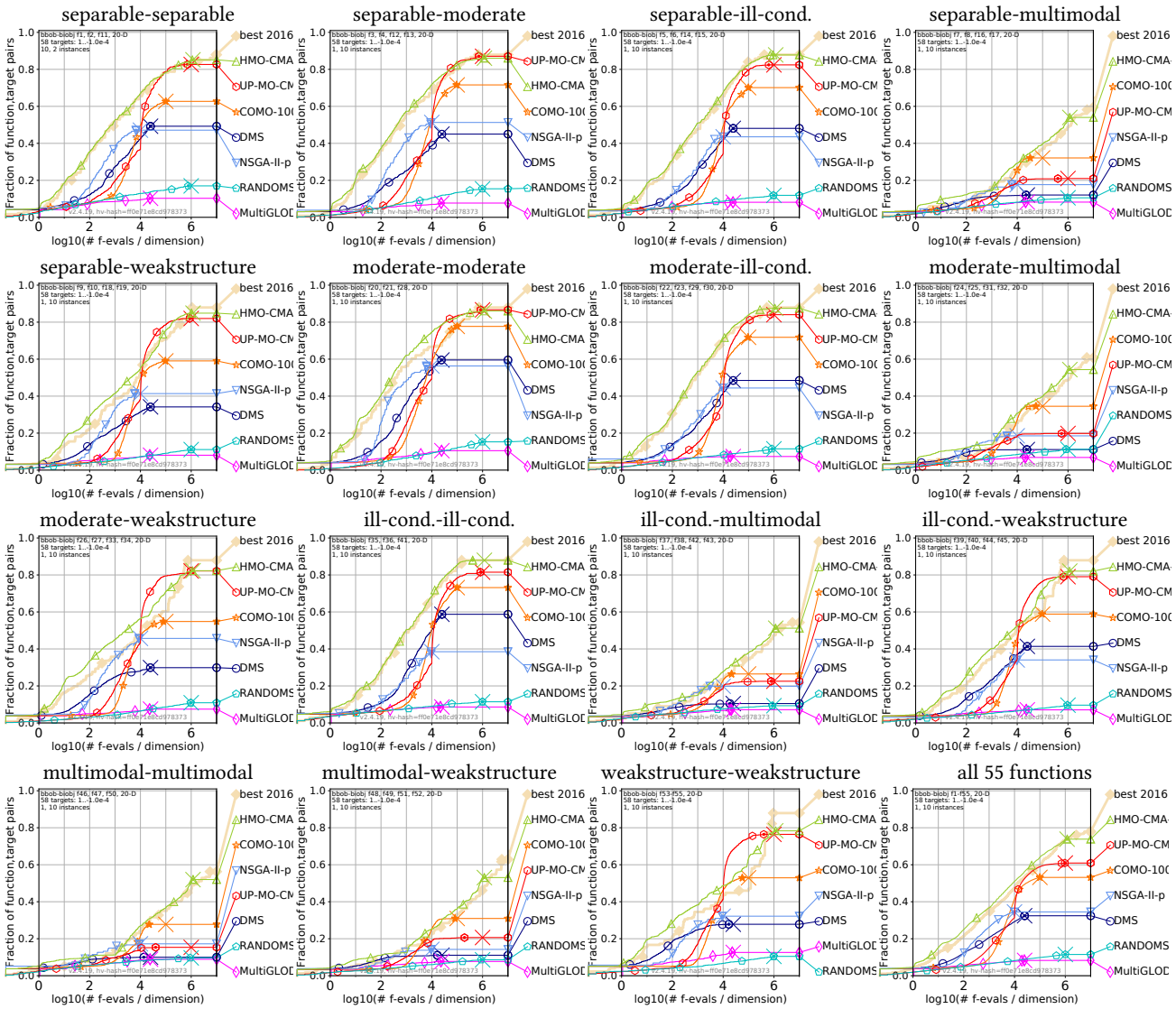


Figure 6: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ for all functions and subgroups in 20-D. As reference algorithm, the best algorithm from BBOB 2016 is shown as light thick line with diamond markers.