Performance of the DEMO Algorithm on the Bi-objective BBOB Test Suite

Tea Tušar
DOLPHIN Team
Inria Lille – Nord Europe
Villeneuve d'Ascq, France
tea.tusar@inria.fr

Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
tea.tusar@ijs.si

Bogdan Filipič
Department of Intelligent Systems
Jožef Stefan Institute
Ljubljana, Slovenia
bogdan.filipic@ijs.si

ABSTRACT

The paper presents the performance of the DEMO (Differential Evolution for Multiobjective Optimization) algorithm on the new bbob-biobj suite of test problems. After limited parameter tuning that comprised different environmental selection procedures, population sizes and crossover probabilities, we identify a parameter setting different from the default one that could be considered for future applications.

Keywords

Benchmarking; Black-box optimization; Bi-objective optimization; Differential evolution

1. INTRODUCTION

DEMO (Differential Evolution for Multiobjective Optimization) [7] is a simple multiobjective optimization algorithm that uses Differential Evolution (DE) [8] to explore the decision space and, originally, nondominated sorting and crowding distance (as in NSGA-II [2]) to select the best solutions for the next population. DEMO was later [10] coupled with environmental selections from SPEA2 [13] and IBEA [12]. Experiments have shown that on the majority of the 16 tested problems, the DEMO variants $\mathrm{DEMO}^{\text{NS-II}}$, $\mathrm{DEMO}^{\text{SP2}}$ and $\mathrm{DEMO}^{\text{IB}}$ performed significantly better than the corresponding algorithms NSGA-II, SPEA2 and IBEA.

Apart from the environmental selection procedures, which stem from multiple objectives and are therefore particular to DEMO, all the other parameters of DEMO are the same as for DE, namely population size, crossover probability CR and scaling factor F. The mentioned study used fixed values for all DE parameters.

In this study we wish to explore the performance of DEMO on a new suite of problems called *bbob-biobj* [9], which contains 55 bi-objective functions with different properties in 10 different instances and 6 dimensions of the decision space (2-D, 3-D, 5-D, 10-D, 20-D and 40-D[1]). In the experiments we do some (very limited) parameter tuning of DEMO parameters to see if a different parameter setting should be considered in future DEMO applications.

GECCO'16 Companion, July 20-24, 2016, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/2908961.2931708

2. EXPERIMENTAL SETUP

2.1 Performance Assessment Using the COCO Platform

All experiments were performed using the COCO (Comparing Continuous Optimizers) platform [5], which enables any-time performance assessment and provides benchmark test suites, experimentation templates [6] and tools for processing and visualizing the data generated by one or several optimizers. The main difference between the performance assessment from COCO [4] and the fixed-budget approach usually used to compare multiobjective optimizers is that COCO's performance assessment is based on the runtime (measured in the number of objective function evaluations) until a quality indicator reaches a predefined target value.

For the bi-objective bbob-biobj test suite [9], the chosen quality indicator is the hypervolume indicator computed from all solutions obtained until the given moment [1]. This is different from the prevalent approach used in evolutionary multiobjective optimization where only the last population of solutions counts. The target values are set according to the hypervolume of a predefined reference set of solutions.

As we have used the bi-objective bbob-biobj suite with 55 functions, 10 instances and 5 dimensions, each experiment was run on a total of 2750 problem instances. The experiments were performed with COCO version 1.0.1, while the plots were produced with version 1.1.

2.2 The DEMO Algorithm

DEMO, like DE, works with a population of solutions. The first population is initialized randomly from $[-5, 5]^D$, where D is the dimensionality of the decision space, which is sure to contain the extremes of the Pareto set. From the first generation on, DEMO is allowed to go spread of this space to the region of interest of the bbob-biobj suite, which equals $[-100, 100]^D$ and allegedly contains the entire Pareto set. The different handling of the first population is an adaption of DEMO to this problem suite and has not been used before. Other than this, the DEMO implementation used in these experiments is the same as in [10].

[1]Due to time limitations we skip experiments on the optional 40-D decision space.

Table 1: All parameter values used in parameter tuning of the DEMO algorithm on the bbob-biobj test suite (D is the problem dimension).

Environmental selection	Population size	Crossover probability CR	Scaling factor F
NS-II	100	0.3	0.5
SP2	100	0.3	0.5
IB	100	0.3	0.5
NS-II	$\lfloor 100 \ln(D) \rfloor$	0.3	0.5
NS-II	$20D$	0.3	0.5
NS-II	$\lfloor 100 \ln(D) \rfloor$	0.1	0.5
NS-II	$\lfloor 100 \ln(D) \rfloor$	0.5	0.5
NS-II	$\lfloor 100 \ln(D) \rfloor$	0.7	0.5
NS-II	$\lfloor 100 \ln(D) \rfloor$	0.9	0.5

At each step, DEMO uses the *DE/1/rand/bin* strategy to construct a new solution from three existing solutions and one parent solution. See [7] for a detailed explanation of this strategy and the replacement strategy used in DEMO. In essence, the scaling factor F (usually within $(0, 1+])$ determines how far away from the current solutions the new solution will be placed, with smaller values placing it closer to the existing solutions and larger values placing it further away. Crossover probability $CR \in [0, 1]$ defines the probability of the new solution to *not* inherit variable values from its parent. This means that small values of CR result in the new solution being very similar to its parent. The values for these parameters were fixed in [10] as follows:

- population size = 100,
- crossover probability $CR = 0.3$,
- scaling factor $F = 0.5$.

The DEMO$^{\text{IB}}$ variant follows the environmental selection from IBEA and can use the binary HD indicator as well as the binary additive epsilon indicator [12] (in fact, [10] used both). In this work, we only use the binary HD indicator.

This study explores nine different parameters settings for DEMO, which are summarized in Table 1. The logic behind these choices will be explained in Section 3. Each algorithm run was stopped after $10^5 D$ evaluations have been reached and the last population was completed. Results for the best DEMO variant are finally presented in more detail.

3. RESULTS

In the following we report on the results of the performed experiments on the first five instances of every problem (the other five instances are 'blinded' for verification purposes).

3.1 Experimenting with Environment Selection Procedures

In the first set of experiments, we used the default DEMO parameter settings and modified only the environment selection procedure. The summary results of these experiments over all problem instances for each dimension are presented in the left column of Figure 1.

As we can see, there is almost no difference in the averaged performance between DEMO$^{\text{NS-II}}$ and DEMO$^{\text{SP2}}$, while DEMO$^{\text{IB}}$ performs considerably worse. A more in-depth exploration of the results shows (figures not included here due to space limitations) that DEMO$^{\text{IB}}$ performs better than DEMO$^{\text{NS-II}}$ and DEMO$^{\text{SP2}}$ only on the three test problems with two separable objectives and its performance is especially poor on the problems that have at least one objective with low or moderate conditioning.

Table 2: Population sizes depending on the problem dimension D.

D	$\lfloor 100 \ln(D) \rfloor$	$20D$
2	69	40
3	109	60
5	160	100
10	230	200
20	299	400
40	368	800

It is also very interesting to note that although DEMO$^{\text{IB}}$ generally performs worse than the other two variants, it always produces much larger archives of solutions that were nondominated at the time of creation than the other two variants (archives by DEMO$^{\text{IB}}$ are approximately three times the size of archives by DEMO$^{\text{NS-II}}$ or DEMO$^{\text{SP2}}$).

Since the performance of DEMO$^{\text{NS-II}}$ and DEMO$^{\text{SP2}}$ is similar, we choose the 'default' environmental procedure NS-II, i.e. the original DEMO, for the rest of the experiments.

3.2 Experimenting with Population Sizes

Next, we performed some experiments with different population sizes. DEMO originally used a population size of 100 solutions regardless of the problem dimension (the dimensionality of its decision space). The nice experimental settings provided by COCO enable to explore in more detail how different performance sizes perform on different problem dimensions. We fix the other parameter values to their default values and compare the results of DEMO where its population size is set to 100, $\lfloor 100 \ln(D) \rfloor$ and $20D$, where D is the problem dimension. Table 2 shows their values for the given dimensions. The results of these experiments are presented in the middle column of Figure 1.

We can observe that larger population sizes perform worse at the beginning, but better towards the end. Therefore, if we are given a large number of total evaluations, larger populations are preferable. We set the population size to be equal to $\lfloor 100 \ln(D) \rfloor$ in the remaining experiments.

3.3 Experimenting with Crossover Probabilities

When tuned or controlled [3], DE's parameters CR and F are usually set together as they are not independent. However, due to time limitations we were unable to perform an extensive parameter tuning and have decided to leave F set to its default value 0.5 and experimented only with CR set to 0.1, 0.3, 0.5, 0.7 and 0.9. The summary results of these experiments are shown in the right column of Figure 1.

We see that, with $F = 0.5$, increasing CR values perform better mostly on low-dimensional problems, while the opposite is true on high-dimensional problems. Based on these results we choose 0.9 as the best value for CR.

3.4 Summary

Based on this very limited parameter tuning experiment, we find that the best results were achieved by DEMO with

Environmental selections Population sizes Crossover probabilites

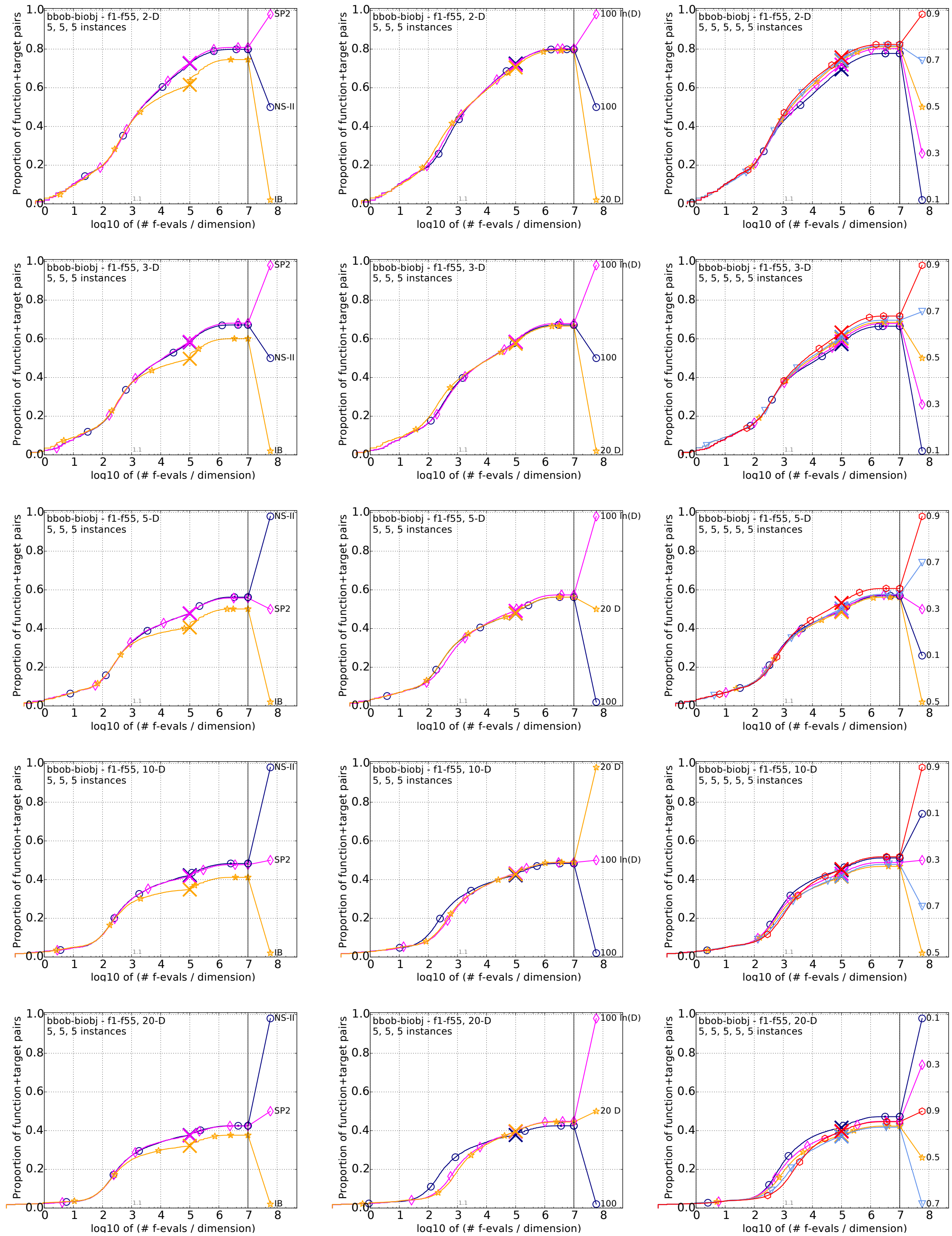

Figure 1: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/D) for the 58 targets $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ aggregated over all problem instances of the same dimension (each dimension is presented in one row). The left column compares different environmental procedures, the middle column population sizes and the right column the crossover probabilities CR.

NS-II environmental selection procedure, population size set to $\lfloor 100 \ln(D) \rfloor$, where D is the problem dimension, $CR = 0.9$ and $F = 0.5$. Results of this setting are presented in more detail in Figure 2, where they are aggregated over functions with similar properties, and finally in Figures 3, 4 and 5, which present results for each separate function.

When examining the plots from Figures 2 to 5 it is important to realize that the performance of DEMO is only relative to the given reference set. So, better performance of DEMO on f_{12} (the Separable ellipsoid/Attractive sector function) than on f_1 (the Sphere/Sphere function) does not necessarily mean that DEMO solves f_{12} better than f_1, but that it solves it better *relative* to the given reference sets.

3.5 CPU Timing Experiment

In order to evaluate the CPU timing of the algorithm, we have run DEMO with the settings from Section 3.4 on the entire bbob-biobj test suite for $10D$ function evaluations. The C code was run on a Windows 7 computer with Intel(R) Core(TM) i5-2410M CPU @ 2.60GHz with 1 processor and 4 cores. The time per function evaluation for dimensions 2, 3, 5, 10, 20 equals 1.82×10^{-4}, 1.21×10^{-4}, 1.45×10^{-4}, 1.09×10^{-4}, 3.64×10^{-4} seconds respectively. In total, this experiment took 9 seconds.

4. CONCLUSION

We have shown how DEMO performs on a new suite of bi-objective benchmark problems called bbob-biobj. We have performed experiments with nine different parameter settings for DEMO and found that the best results were achieved with NS-II environmental selection procedure, population size set to $\lfloor 100 \ln(D) \rfloor$, where D is the problem dimension, $CR = 0.9$ and $F = 0.5$. Although the parameter tuning was by no means complete, it gave some idea on the influence of these parameters on DEMO's performance.

Because the whole archive of solutions is used to evaluate an algorithm's performance, DEMO (as any other population-based algorithm that does not keep track of the archive) could be improved by taking the archive solutions into account when doing environmental selection (and perhaps also for candidate creation). Also, results on different population sizes have hinted that it might be beneficial to start with smaller populations and increase them later on. It might also be sensible to look at DEMO as a self-adaptive algorithm (like DEMOwSA [11]), where population size was another parameter to adapt.

Acknowledgment

The authors acknowledge support by the grant ANR-12-MONU-0009 (NumBBO) of the French National Research Agency, and by the research program P2-0209 of the Slovenian Research Agency.

5. REFERENCES

[1] D. Brockhoff, T. Tušar, D. Tušar, T. Wagner, N. Hansen, and A. Auger. Biobjective performance assessment with the COCO platform. *ArXiv e-prints*, arXiv:1605.01746, 2016.

[2] K. Deb, A. Pratap, S. Agrawal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.

[3] M. Drozdik, H. Aguirre, Y. Akimoto, and K. Tanaka. Comparison of parameter control mechanisms in multi-objective differential evolution. In C. Dhaenens, L. Jourdan, and M.-E. Marmion, editors, *Learning and Intelligent Optimization: 9th International Conference, LION 9, Lille, France, January 12-15, 2015. Revised Selected Papers*, pages 89–103. Springer, 2015.

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

[5] N. Hansen, A. Auger, O. Mersmann, T. Tušar, and D. Brockhoff. COCO: A platform for comparing continuous optimizers in a black-box setting. *ArXiv e-prints*, arXiv:1603.08785, 2016.

[6] N. Hansen, T. Tušar, O. Mersmann, A. Auger, and D. Brockhoff. COCO: The experimental procedure. *ArXiv e-prints*, arXiv:1603.08776, 2016.

[7] T. Robič and B. Filipič. DEMO: Differential evolution for multiobjective optimization. In C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, editors, *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization, EMO 2005, Guanajuato, Mexico*, volume 3410 of *Lecture Notes in Computer Science*, pages 520–533. Springer, 2005.

[8] R. Storn and K. V. Price. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

[9] T. Tušar, D. Brockhoff, N. Hansen, and A. Auger. COCO: The bi-objective black-box optimization benchmarking (bbob-biobj) test suite. *ArXiv e-prints*, arXiv:1604.00359, 2016.

[10] T. Tušar and B. Filipič. Differential evolution versus genetic algorithms in multiobjective optimization. In S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, editors, *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007, Matsushima, Japan*, volume 4403 of *Lecture Notes in Computer Science*, pages 257–271. Springer, 2007.

[11] A. Zamuda, J. Brest, B. Bosković, and V. Žumer. Differential evolution for multiobjective optimization with self adaptation. In *2007 IEEE Congress on Evolutionary Computation*, pages 3617–3624, 2007.

[12] E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In X. Yao, E. K. Burke, J. A. Lozano, J. Smith, J. J. M. Guervós, J. A. Bullinaria, J. E. Rowe, P. Tino, A. Kabán, and H.-P. Schwefel, editors, *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature, PPSN VIII, Birmingham, UK*, volume 3242 of *Lecture Notes in Computer Science*, pages 832–842. Springer, 2004.

[13] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. In K. C. Giannakoglou, editor, *Proceedings of Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, EUROGEN 2001, Athens, Greece*, pages 95–100. CIMNE, 2002.

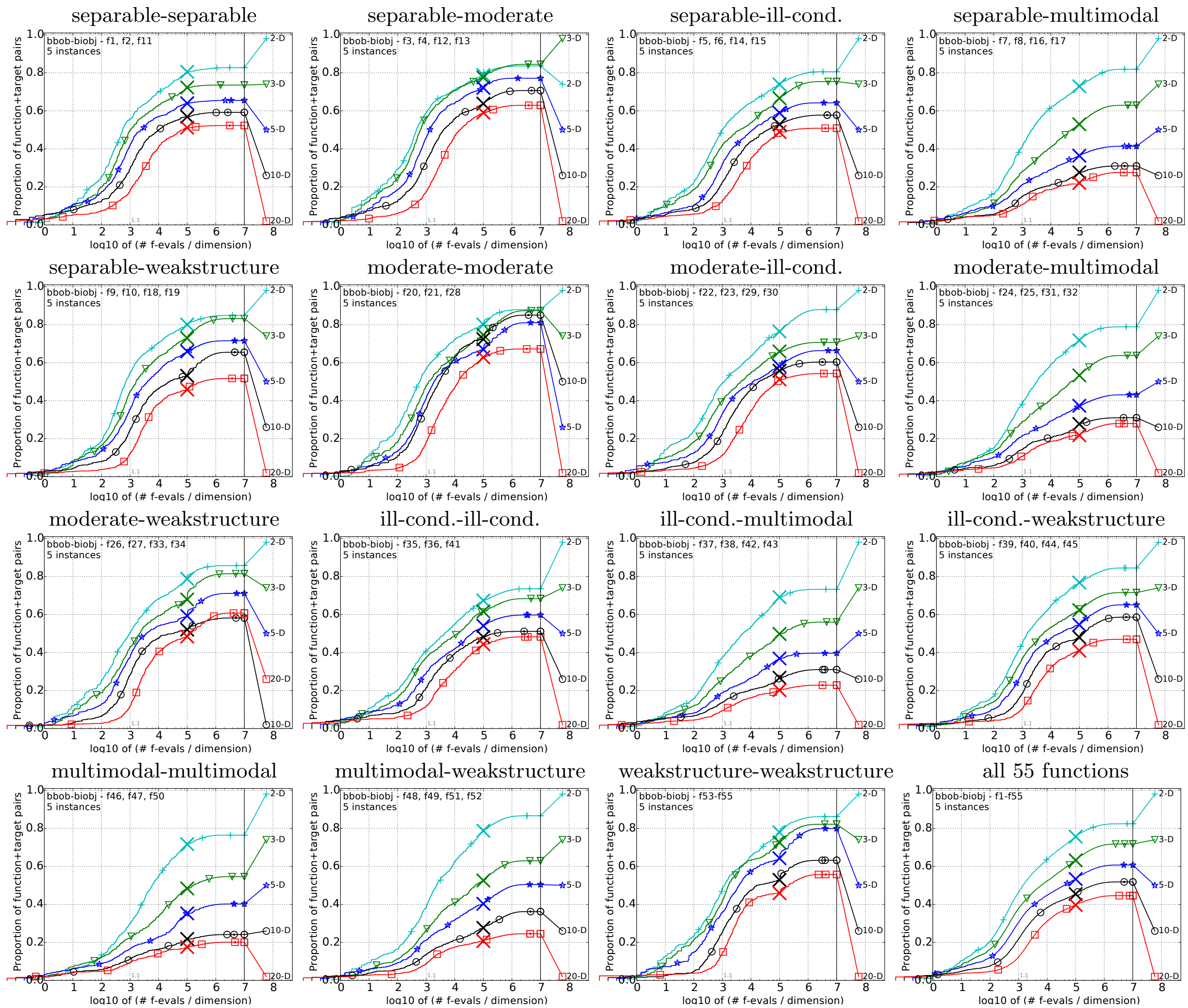

Figure 2: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/D) for the 58 targets $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ for all function groups and all dimensions. The aggregation over all 55 functions is shown in the last plot.

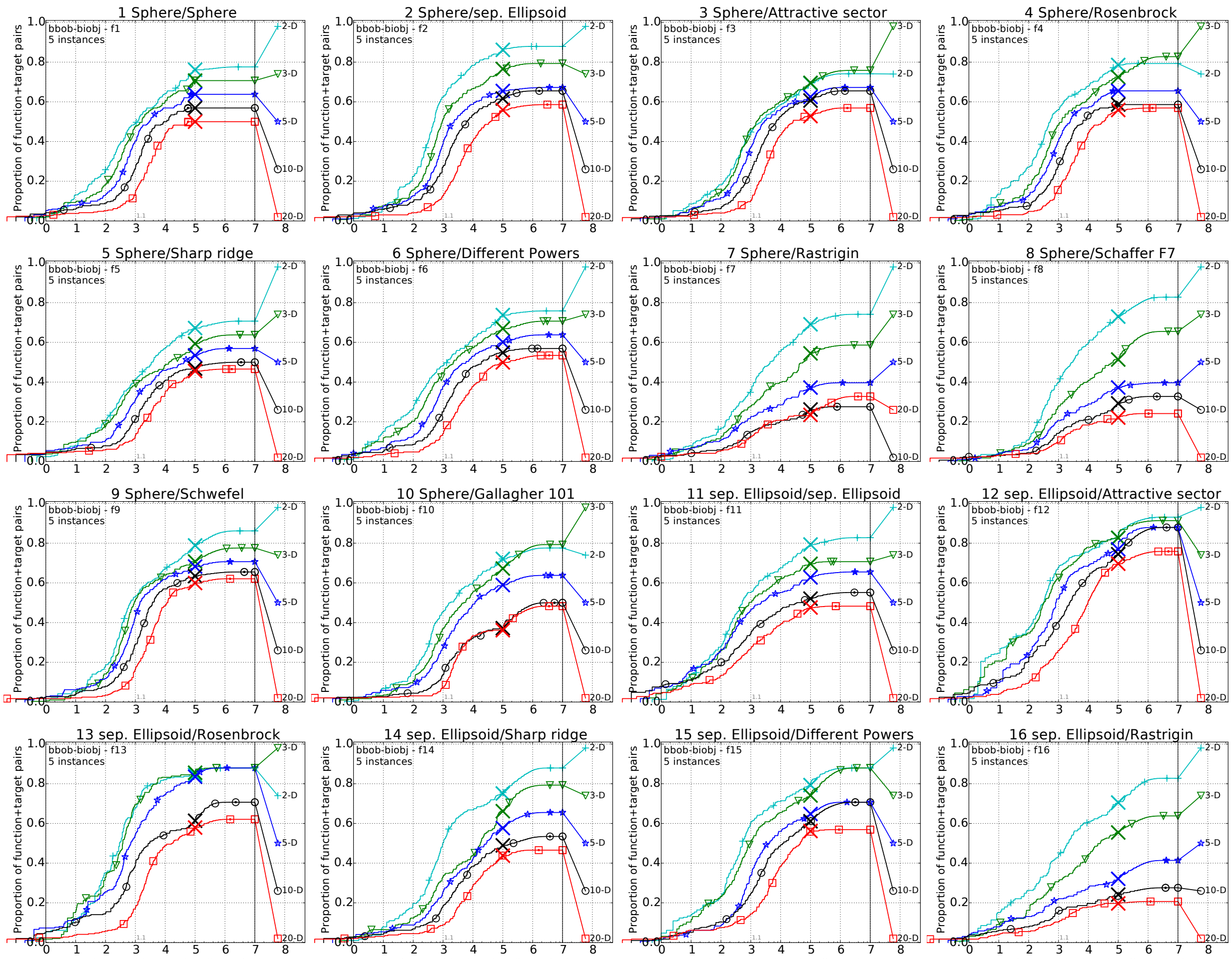

Figure 3: Empirical cumulative distribution of simulated (bootstrapped) runtimes in number of objective function evaluations divided by dimension (FEvals/D) for the 58 targets $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ for functions f_1 to f_{16} and all dimensions.

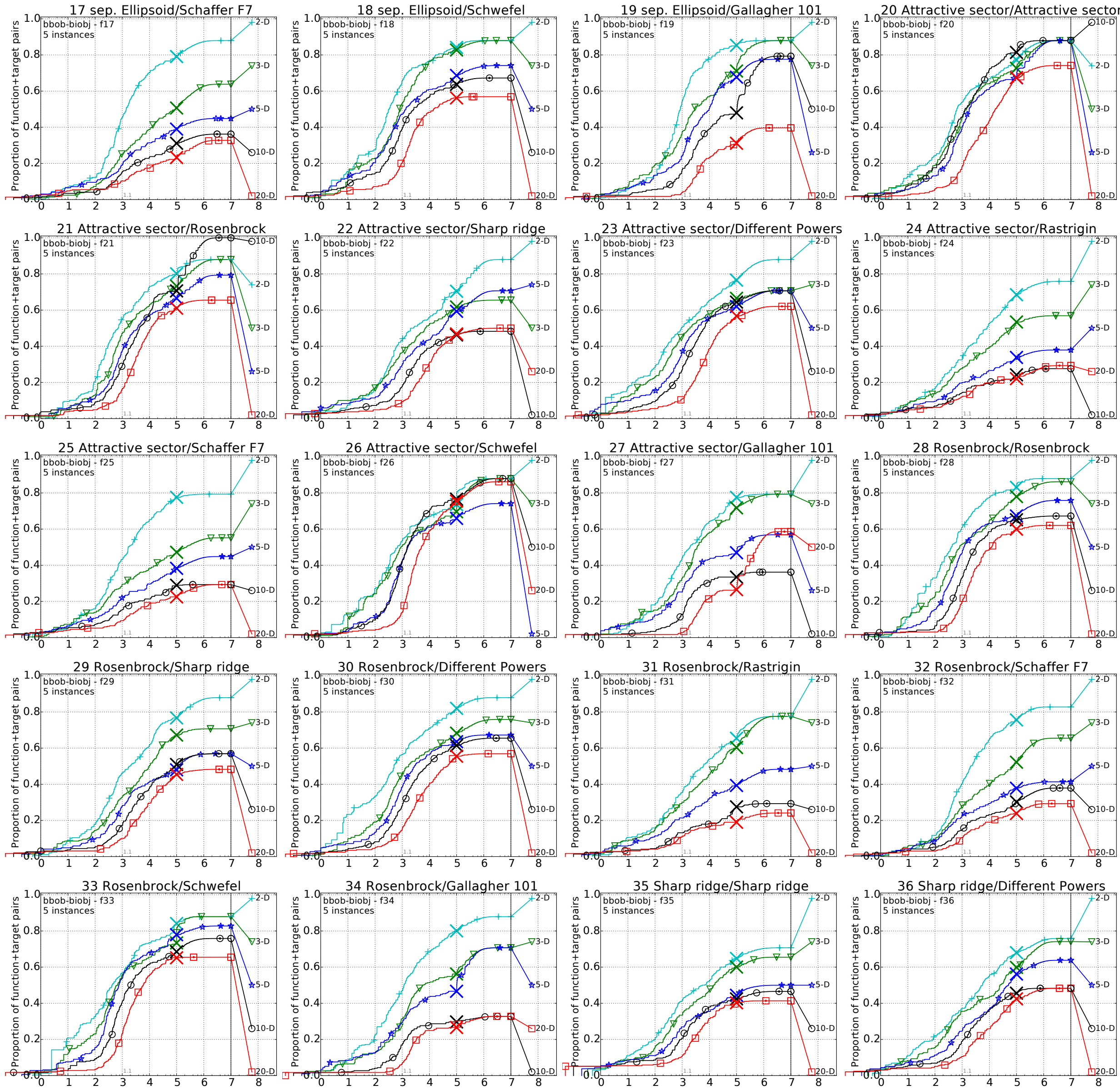

Figure 4: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/DIM) for the targets as given in Fig. 3 for functions f_{17} to f_{36} and all dimensions.

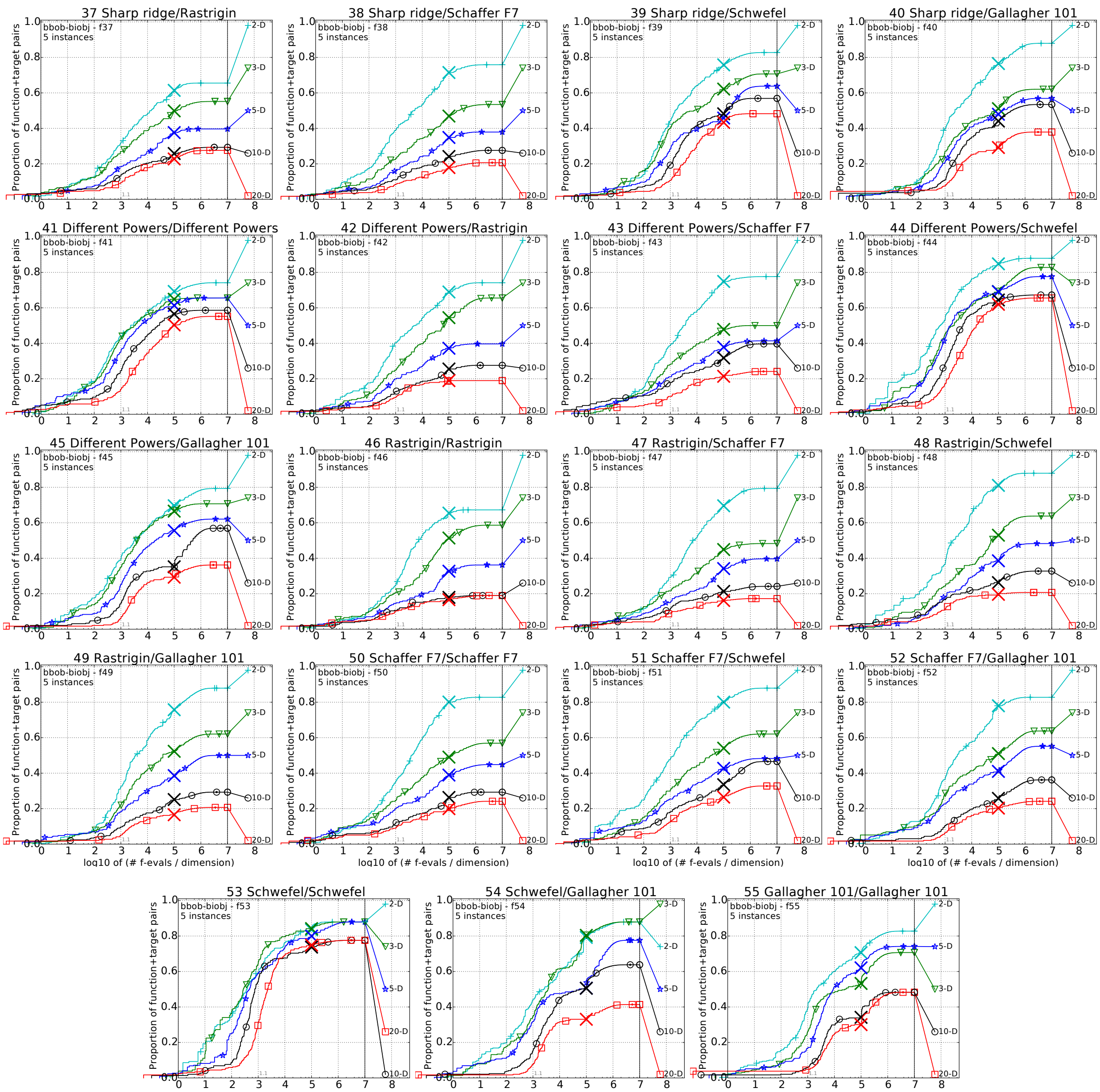

Figure 5: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of objective function evaluations, divided by dimension (FEvals/DIM) for the targets as given in Fig. 3 for functions f_{37} to f_{55} and all dimensions.