Benchmarking ϵMAg-ES and BP-ϵMAg-ES on the `bbob-constrained` Testbed

Michael Hellwig
Vorarlberg University of Applied Sciences,
Josef Ressel Center for Robust Decision Making
Dornbirn, Austria
michael.hellwig@fhv.at

Hans-Georg Beyer
Vorarlberg University of Applied Sciences,
Research Center Business Informatics
Dornbirn, Austria
hans-georg.beyer@fhv.at

ABSTRACT

The ϵMAg Evolution Strategie and its bi-population version (BP-ϵMAg-ES), which performed well on the CEC 2018 and CEC 2020 benchmark sets, are evaluated using the COCO `bbob-constrained` 2.6.2 benchmark release. Furthermore, the performance of Matlab's fmincon is considered as a baseline algorithm. Unlike the results of the CEC competitions, the outcomes obtained on the COCO 2.6.2 test set do not reveal any advantage of using a bi-population approach. The reasons for these observations needs still to be investigated in future research.

CCS CONCEPTS

• **Computing methodologies**; • **Mathematics of computing** → **Continuous mathematics**;

KEYWORDS

Benchmarking, Black-box Optimization, Constrained Optimization, Evolution Strategies

ACM Reference Format:
Michael Hellwig and Hans-Georg Beyer. 2022. Benchmarking ϵMAg-ES and BP-ϵMAg-ES on the `bbob-constrained` Testbed. In *Genetic and Evolutionary Computation Conference Companion (GECCO '22 Companion), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3520304.3534010

1 INTRODUCTION

Constrained optimization is a highly relevant research field with a clear link to actual economic and technical problems. It is concerned with searching for the optimal solution of an objective function taking into account required limitations on the parameter vector components. Such limitations may arise from multiple sources, e.g. limited resources of a problem, problem-specific trade-offs, or appropriate physical boundaries. Introducing constraints into an optimization task increases the complexity of the problem. This particularly applies in the context of black-box optimization, where the analytical structure of the optimization problem must not be used by an algorithm. Due to the high complexity and long computation times, it is reasonable to perform the evaluation and comparison of solvers on specially designed benchmark environments. This allows for a determined evaluation of the strengths and weaknesses of an algorithm with respect to given problem characteristics.

Considering search heuristics for continuous optimization, only a few principled benchmark environments exist for constrained optimization tasks. The first testbeds were introduced for competitions during the IEEE Congress on Evolutionary Computation (CEC). Over the years, the competition on constrained real-parameter optimization provided specific test environments for the evaluation and comparison of stochastic search algorithms. We will refer to these by using a combination of the conference acronym and the associated year of publication. To date, four sets of test function sets (CEC2006, CEC2010, CEC2017, CEC2020) have been proposed in the context. Apart from the competitions, these test function environments turned out very popular for benchmarking Evolutionary Algorithms (EA).

The Comparing Continuous Optimizer (COCO) platform has also become a widely recognized benchmarking tool for diverse problem types. However, constrained problems were only available in a beta version so far. This has changed with the latest COCO releases. The COCO `bbob-constrained` testbed represents a state-of-the-art performance assessment framework for continuous constrained optimization problems. The BBOB-constrained benchmark examines the performance of an algorithm on 54 constrained functions. These constrained functions consist of nine objective functions with six combinations of nonlinear inequality constraints each which are considered in six different problem dimensions. The systematic design of `bbob-constrained` testbed contrasts with the CEC benchmark sets, which may appear rather unstructured. However, the constrained CEC benchmarks do contain a broader range of constrained problem characteristics, e.g. equality constraints, different types of objective functions, and even real-world problems.

The substantial differences in the various benchmark environments motivate the algorithm evaluation in this work. This paper compares two Evolution Strategies (ES) variants that have been demonstrated to be effective in the context of the constrained CEC competitions. Both algorithm variants are based on the Matrix Adaptation Evolution Strategy (MA-ES) [1]. One of these algorithms is the ϵMAg-ES, which showed very good results on the CEC2017 testbed, especially in high search space dimensions. The other is the BP-ϵMAg-ES, a follow-up to the ϵMAg-ES, which was developed for the CEC2020 competition on constrained real-world problems. The paper will investigate both variants on the BBOB-constrained benchmark and will provide an indication of their effectiveness for specific problem types.

GECCO '22 Companion, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9268-6/22/07…$15.00
https://doi.org/10.1145/3520304.3534010

The remainder of this paper is organized as follows: First, the general constrained problem considered within the COCO BBB-constrained testbed is specified in the next section. Section 3 then briefly introduces the algorithms for constrained optimization (the ϵMAg-ES and the BP-ϵMAg-ES) and their constraint handling techniques. The experimental procedure, as well as the parameter settings used for the ϵMAg-ES, are summarized in Sec. 4. We present the corresponding results in Sec. 6 and conclude the paper with a discussion of the observations in Sec. 7.

2 PROBLEM FORMULATION

According to [3], the constrained optimization problems considered in the BBOB-constrained benchmark exhibit the general form

$$\begin{aligned} &\min\ f(\mathbf{y}) \\ &s.t.\ g_i(\mathbf{y}) \leq 0, \qquad i = 1, \ldots, m, \\ &\quad \mathbf{y} \in S \subseteq \mathbb{R}^N. \end{aligned} \tag{1}$$

Without loss of generality, the optimization goal is the minimization of the real-valued objective function $f(\mathbf{y})$. Here, $\mathbf{y} \in S$ denotes the N-dimensional search space parameter vector. The set S either represents the hypercube $[-5; 5]^N$ or the whole $\mathbb{R}^N$. Additionally, the feasible region of the search space is restricted by m real-valued non-linear inequality constraint functions $g_i(\mathbf{y})$. A vector $\mathbf{y} \in S$ that satisfies all constraints simultaneously is called feasible. The set of all feasible parameter vectors is denoted

$$M := \{\mathbf{y} \in S: g_i(\mathbf{y}) \leq 0 \quad \forall i\}. \tag{2}$$

The global optimum of (1) is denoted by $\mathbf{y}^* \in M$. We will refer to the system (1) of objective function $f(\mathbf{y})$ subject to a number of constraints as a *constrained function.*

The constraint violation $\nu(\mathbf{y})$ of a vector $\mathbf{y}$ is computed as

$$\nu(\mathbf{y}) = \textstyle\sum_{i=1}^{m} G_i(\mathbf{y}), \tag{3}$$

with functions $G_i(\mathbf{y})$ defined by

$$G_i(\mathbf{y}) := \max\left(0, g_i(\mathbf{y})\right). \tag{4}$$

It is going to be used as an infeasibility metric for ranking potentially infeasible candidate solutions to problem (1). The comparison of candidate solutions always involves their evaluation on the constrained problem and thus consumes function evaluations. One function evaluation of the total budget is accounted for per evaluation of a constrained problem (i.e. objective function and all related constraint functions) regardless of whether only the constraint and/or objective function values are used by the algorithm.

3 ALGORITHM PRESENTATION

This section recaps the working principles of ϵMAg-ES and BP-ϵMAg-ES. The ϵMAg-ES [11] was proposed in the context of the CEC2018 competition on single-objective constrained real-parameter optimization using the same benchmark functions as [19]. It represents a predecessor of the BP-ϵMAg-ES proposed for the IEEE CEC2020 competition on Real-World Single-Objective Constrained Optimization [14]. Both ES variants rely on the MA-ES [1] which represents an algorithmically simplified CMA-ES. In contrast to CMA, the MA-ES replaces learning a problem-specific covariance matrix with learning a so-called adaptation matrix. To deal with constrained problems, e.g. (1), the MA-ES concept was augmented with constraint handling techniques established in the context of DE [18]. The remainder of this section will mainly recap these constraint handling techniques with a focus on the differences between both ES variants. For detailed explanations and the corresponding pseudo-codes we refer to the original papers [11], and [12], respectively.

Both algorithm variants repeatedly sample an offspring population of size λ within the specified range of parameter vector components. Yet, the magnitude of λ differs. While the ϵMAg-ES uses a fixed population size of $\lambda = 4N$, the BP-ϵMAg-ES starts with a default value of $\lambda = 4 + 3\lfloor \log(N) \rfloor$ and varies the population size in a multi-restart approach similar to the BiPop-CMA-ES in [1]. In both cases, the parental population size is determined as $\mu = \lfloor \lambda/3 \rfloor$. That is, the best μ offspring are selected to form the recombinant that is passed to the next generation. The quality of the offspring is measured by the use of a special order relation considering constraint violation and objective function value.

For dealing with problem (1), both ES variants use three very similar constraint handling techniques:

The **satisfaction of box-constraints** is ensured for every single candidate solution $y \in \mathbb{R}^N$ prior to its evaluation. Regarding the upper and lower parameter bounds of the set S in the constrained function definition, each exceeding component is reflected back into the box of valid parameter values [11].

The selection operator within ES needs to rank the generated offspring individuals of a single generation. Usually, feasible candidate solutions are always considered superior to infeasible solutions. The ϵMAg-ES uses a relaxation of the usual lexicographic ordering, the so-called **ϵ-level order relation** introduced in [17], that enables the ES to treat infeasible candidate solutions with constraint violation below a specific constraint violation level as feasible. A candidate solution y is called ϵ-feasible, if its constraint violation $\nu(\mathbf{y})$ does not exceed the $\epsilon^{(g)}$ threshold in generation g. The $\epsilon^{(g)}$-level is continuously reduced to zero with the number of generations. This way, the strategy is able to move outside the feasible region within the early phase of the search which can potentially support the convergence to the optimizer y^*.

In addition to ϵ-level order relation, the ϵMAg-ES uses **gradient-based repair** to erratically repair infeasible candidate solutions based on the gradient of the constraint functions. To this end, the Jacobian is approximated by the use of a finite differences approach. The consumed function evaluations have to be accounted for properly. If the correction is not resulting in a feasible solution, the gradient-based repair step is repeated at most $\theta_r = 3$ times. In case the strategy cannot find a feasible solution, the last infeasible candidate solution is considered as the new offspring candidate solution. A detailed description of the repair approach is provided in [11].

While the ϵMAg-ES basically employs these techniques to deal with the constraints, the BP-ϵMAg-ES comes with an additional restart mechanism that varies the population size. This approach was inspired by the BiPop-MA-ES implementation [1] for unconstrained problems, which is based on the ideas of [5] and [15].

Within the restarting scheme of the BP-ϵMAg-ES a modified ϵMAg-ES represents the core search driver. The modification mainly consists of the replacement of the continuous reduction of the ϵ-level towards zero during the early search phase by an ϵ-level adaptation approach that takes into account the feasibility ratio of the candidate solutions. The magnitude of constraint relaxation is decreased as long as the ratio of ϵ-feasible solutions in the parental population of size μ exceeds a minimum value. Otherwise, the current ϵ-level is increased with the intention of giving the ES more flexibility to find better points through stronger relaxation [4]. Still, after a predefined number of generations, the ϵ-level is ultimately set to zero in late search phases.

The mutation strength within the ES variants is bounded from above by the parameter $\sigma_{\max}$. While $\sigma_{\max}$ was set to a fixed value for the CEC2018 competition in [11], the modified ϵMAg-ES determines $\sigma_{\max}$ on the basis of the problem-specific box constraints.

After the computation of all λ offspring, the best offspring is compared to the best-found solution so far $\mathbf{y}_{\text{bsf}}$ and the parental recombinant $\mathbf{y}^{(g)}$ is updated. The update involves the selection of the best μ mutation vectors with respect to the ϵ-level ordering. After that, the ϵMAg-ES adapts the search path $\mathbf{p}_\sigma^{(g)}$ known from CMA-ES [9]. Its length indicates whether the mutation strength $\sigma^{(g)}$ should be decreased or increased in the next generation. It also contributes to the transformation matrix update of the ϵMAg-ES. The strategy parameters are chosen according to the recommendations in [1]. Details are provided in the following section.

These steps are repeated until the budget of function evaluations is exceeded in the ϵMAg-ES, or until a restart is performed within the BP-ϵMAg-ES. After termination, the algorithms return the best-found solution $\mathbf{y}_{\text{bsf}}$ together with its objective function value $f(\mathbf{y}_{\text{bsf}})$, and the corresponding constraint violation $\nu(\mathbf{y}_{\text{bsf}})$.

4 EXPERIMENTAL PROCEDURE

This section summarizes the experimental setup and provides the parameter settings used by the algorithms to solve the related benchmark problems.

All experiments are carried out on the constrained benchmark problems and according to the specifications provided in [2]. Consequently, each algorithm was executed on 15 instances of each of the 9 test problems and in each dimension $N = \{2, 3, 5, 10, 20, 40\}$.

Within both approaches, the ϵMAg-ES uses the standard parameters recommended for the MA-ES in [1]. The weights are

$$w_i = \frac{\ln(\mu + 0.5) - \ln i}{\sum_{j=1}^{\mu}(\ln(\mu + 0.5) - \ln j)}, \text{ for } i \in \{1, \dots, \mu\} \tag{5}$$

and the corresponding effective population size is given as $\mu_w = 1/\sum_{i=1}^{\mu} w_i^2$. The learning rates of the mutation strength, the search path, and the transformation matrix update are specified as

$$c_\sigma = \frac{\mu_w + 2}{N + \mu_w + 5} \tag{6}$$

$$c_1 = \frac{2}{(N + 1.3)^2 + \mu_w}, \text{ and} \tag{7}$$

$$c_\mu = \min\left[1 - c_1, \frac{2(\mu_w - 2 + 1/\mu_w)}{(N + 2)^2 + \mu_w}\right] \tag{8}$$

Regarding the parameters of the ϵMAg-ES, the recommendations in [11] are considered. It uses an offspring population size of $\lambda = 4N$ with corresponding parental population size of $\mu = \lfloor \lambda/3 \rfloor$. The ϵ-threshold of $\leq_\epsilon$ is gradually reduced during the first $T = 1000$ generations of the search. The gradient-based repair is applied at most $\theta_r = 3$ times with probability $\theta_p = 0.2$ in every Nth generation.

In contrast to ϵMAg-ES, BP-ϵMAg-ES uses the population size parameters $\lambda = 4 + \lfloor 3\log(N) \rfloor$ and $\mu = \lfloor \lambda/3 \rfloor$. The ϵ-threshold is gradually reduced during the first $T = 500$ generations of the search. The adaptive ϵ-level control rule is described in [12]. The gradient-based repair again is applied at most 3 times with probability $\theta_p = 0.2$ in every Nth generation.

The starting population is initialized randomly within the set S and the initial mutation strength of both ES variants is set to $\sigma^{(0)} = 1$. In the CEC context, a strong increase in mutation strength was observed on some constrained problems. Hence, the growth of $\sigma^{(g)}$ was limited to $\sigma_{\max} = 100$ for the ϵMAg-ES. The BP-ϵMAg-ES defines its upper bound on σ based on the box constraints, i.e. it uses $\sigma_{\max} = 10$ in the case of the bbob-constrained testbed.

This paper considers FMINCON as a baseline solver for the algorithm comparison. We used FMINCON of Matlab 2021b with the default settings in this study. The solver was equipped with the same budget (i.e. $10^5 \cdot N$ function evaluations) as the ES variants and applied to the bbob-constrained testbed without restarts.

5 CPU TIMING

In order to evaluate the CPU timing of the algorithm, we have run the ϵ-MAg-ES without restarts on the entire bbob-constrained test suite [2] for $10^5 \cdot N$ function evaluations according to [10]. The Matlab code was run on a HP Intel(R) Core(TM) i9-10920X CPU @ 3.5GHz with 1 processor and 12 cores.

For the ϵ-MAg-ES , the time per function evaluation for dimensions 2, 3, 5, 10, 20, and 40 equals 1.115808e-05 , 1.098617e-05, 1.341433e-05, 1.731213e-05, 3.054588e-05, and 9.474365e-05 seconds respectively. For the BP-ϵ-MAg-ES in the same setting, the time per function evaluation for dimensions 2, 3, 5, 10, 20, and 40 equals 2.241828e-05 , 1.708038e-05, 1.885893e-05, 2.426081e-05, 3.900334e-05, and 8.815745e-05 seconds respectively.

6 RESULTS

Results from experiments according to [10] and [6] on the benchmark functions given in [2] are presented in Figures 1, 2, 3, 4 and 5 as well as the Figures and Tables in the supplementary material. The experiments were performed with COCO [8], version 2.6.2, the plots were produced with version 2.6.2, and all related date are available on Zenodo [13].

The **expected runtime (ERT)**, used in the figures and tables, depends on a given target value, $\tau_t = f_{\text{opt}} + \Delta\tau$, and is computed over all relevant trials as the sum of function and constraint evaluations executed during each trial while the best value of the algorithms did not reach τ_t, summed over all trials and divided by the number of trials that actually reached τ_t [2, 7, 16]. **Statistical significance** is tested with the rank-sum test for a given target $\Delta\tau_t$ using, for each trial, either the number of needed function and constraints evaluations to reach $\Delta\tau_t$ (inverted and multiplied by −1), or, if the target was not reached, the best $\Delta\tau$-value achieved, measured only up to the smallest number of overall (function + constraints) evaluations for any unsuccessful trial under consideration.

7 DISCUSSION

Considering the expected running time (ERT) of the ES variants in Figures 1 and 2, one observes only limited performance differences between ϵ-MAg-ES and BP-ϵ-MAg-ES on the `bbob-constrained` benchmark. The illustrations indicate which algorithms were able to reach the final target of 10^{-6} with the specified budget of $10^5 \cdot N$ constrained function evaluations ($N = 2, 3, 5, 10, 20, 40$). In comparatively small dimensions, both algorithms are still able to reach the predefined final targets. Yet, in dimensions greater or equal to $N = 10$, the algorithms fail on 24 of 54 problems. Particularly, the multimodal test function group (functions 43 - 54) appears to present a great level of difficulty for the presented ES variants.

The ECDF results presented in Fig. 3 also indicate that both ES variants show a very similar performance on most instances of the bbob-constrained benchmark functions. While both algorithms hit a great proportion of targets in smaller dimensions, the ratio drops under 60% with increasing dimensionality.

The Figures 4 and 5 provide a more detailed picture of the algorithm performances on all individual constrained functions in dimension $N = 20$. For detailed results in other dimensions, we refer to the supplementary material as well as to the performance data available via the COCO platform.

Interestingly, the BP-ϵ-MAg-ES is not able to improve the ϵ-MAg-ES results on the constrained Rastrigin functions. While the bi-population approach was originally designed for multimodal test functions, it appears to be useless in the present constrained scenarios. The reasons for this can only be speculated here, and a more detailed investigation of this issue is required.

Taking into account the corresponding FMINCON results, the ES turns out not to be able to compete on the easier test functions in smaller problem dimensions.

It should be noted that the present algorithm comparison is carried out with the same strategy parameter settings that were recommended for CEC competitions. A testbed-tailored parameter tuning has not been performed for `bbob-constrained`. This is on the one hand due to the fact that we intended to stay close to the algorithm descriptions in [11, 12] and on the other hand due to the short time since the last COCO 2.6.2 release. In this respect, even the choice of the initial mutation strength might be too large on problems with a very small feasible region. This also regards the choice of the ϵ-level schedule that has not been adopted so far for the `bbob-constrained` benchmark set.

Further, it must be emphasized that the considered strategies randomly generate their starting populations within the box constraints, i.e. the set S on the `bbob-constrained` testbed. This initialization procedure was retained even though the COCO framework allows to provide the algorithms with an initial feasible solution.

The aforementioned design decisions can serve as potential explanations for the mediocre performance of the algorithms presented. However, these hypotheses need to be investigated in more detail in future work.

ACKNOWLEDGEMENT

This work was partially supported by the Austrian Science Fund FWF under grant P33702. The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association are gratefully acknowledged.

REFERENCES

[1] Hans-Georg Beyer and Bernhard Sendhoff. 2017. Simplify Your Covariance Matrix Adaptation Evolution Strategy. *IEEE Transactions on Evolutionary Computation* 21, 5 (Oct 2017), 746–759.

[2] Paul Dufossé, Asma Atamna, Philippe R. Sampaio, Dimo Brockhoff, Nikolaus Hansen, and Anne Auger. 2022. Building scalable test problems for benchmarking constrained optimizers. (2022).

[3] Paul Dufossé, Nikolaus Hansen, Dimo Brockhoff, Phillipe R. Sampaio, Asma Atamna, and Anne Auger. 2022. *Building scalable test problems for benchmarking constrained optimizers*. Technical Report. http://numbbo.github.io/coco-doc/bbob-constrained/ To be submitted to the SIAM Journal of Optimization.

[4] Zhun Fan, Yi Fang, Wenji Li, Yutong Yuan, Zhaojun Wang, and Xinchao Bian. 2018. LSHADE44 with an Improved ϵ Constraint-Handling Method for Solving Constrained Single-Objective Optimization Problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. DOI:http://dx.doi.org/10.1109/CEC.2018.8477943

[5] Nikolaus Hansen. 2009. Benchmarking a BI-Population CMA-ES on the BBOB-2009 Function Testbed *(GECCO '09)*. Association for Computing Machinery, New York, NY, USA, 2389–2396. DOI:http://dx.doi.org/10.1145/1570256.1570333

[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, S. Finck, and R. Ros. 2012. *Real-Parameter Black-Box Optimization Benchmarking 2012: Experimental Setup*. Technical Report. INRIA. http://numbbo.github.io/gforge/bbob2012-downloads

[8] 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

[9] N. Hansen, S. D. Müller, and P. Koumoutsakos. 2003. Reducing the Time Complexity of the Derandomized Evolution Strategy with Covariance Matrix Adaptation (CMA-ES). *Evolutionary Computation* 11, 1 (March 2003), 1–18. DOI:http://dx.doi.org/10.1162/106365603321828970

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

[11] Michael Hellwig and Hans-Georg Beyer. 2018. A Matrix Adaptation Evolution Strategy for Constrained Real-Parameter Optimization. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. DOI:http://dx.doi.org/10.1109/CEC.2018.8477950

[12] Michael Hellwig and Hans-Georg Beyer. 2020. A Modified Matrix Adaptation Evolution Strategy with Restarts for Constrained Real-World Problems. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. 1–8. DOI:http://dx.doi.org/10.1109/CEC48606.2020.9185566

[13] Michael Hellwig and Hans-Georg Beyer. 2022. COCO BBOB-constrained Benchmark Results of Two Evolution Strategy Variants. (2022). https://doi.org/10.5281/zenodo.6463075 Zenodo.

[14] Abhishek Kumar, Guohua Wu, Mostafa Z. Ali, Rammohan Mallipeddi, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. 2020. A test-suite of non-convex constrained optimization problems from the real-world and some baseline results. *Swarm and Evolutionary Computation* 56 (2020), 100693. DOI:http://dx.doi.org/https://doi.org/10.1016/j.swevo.2020.100693

[15] Ilya Loshchilov. 2013. CMA-ES with restarts for solving CEC 2013 benchmark problems. *2013 IEEE Congress on Evolutionary Computation* (2013), 369–376.

[16] Kenneth Price. 1997. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation.* IEEE, Piscataway, NJ, USA, 153–157. DOI:http://dx.doi.org/10.1109/ICEC.1997.592287

[17] Tetsuyuki Takahama and Setsuko Sakai. 2006. Constrained Optimization by the ϵ Constrained Differential Evolution with Gradient-Based Mutation and Feasible Elites. In *IEEE International Conference on Evolutionary Computation, CEC, part of WCCI 2006, Vancouver, BC, Canada, 16-21 July*. 1–8. DOI:http://dx.doi.org/10.1109/CEC.2006.1688283

[18] Tetsuyuki Takahama and Setsuko Sakai. 2010. Constrained optimization by the epsilon constrained differential evolution with an archive and gradient-based mutation. In *Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2010, Barcelona, Spain, 18-23 July*. 1–9. DOI:http://dx.doi.org/10.1109/CEC.2010.5586484

[19] G. H. Wu, R. Mallipeddi, and P. N. Suganthan. 2017. *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Single Objective Real-Parameter Optimization.* Technical Report. https://github.com/P-N-Suganthan/CEC2017 Technical Report, Nanyang Technological University, Singapore.

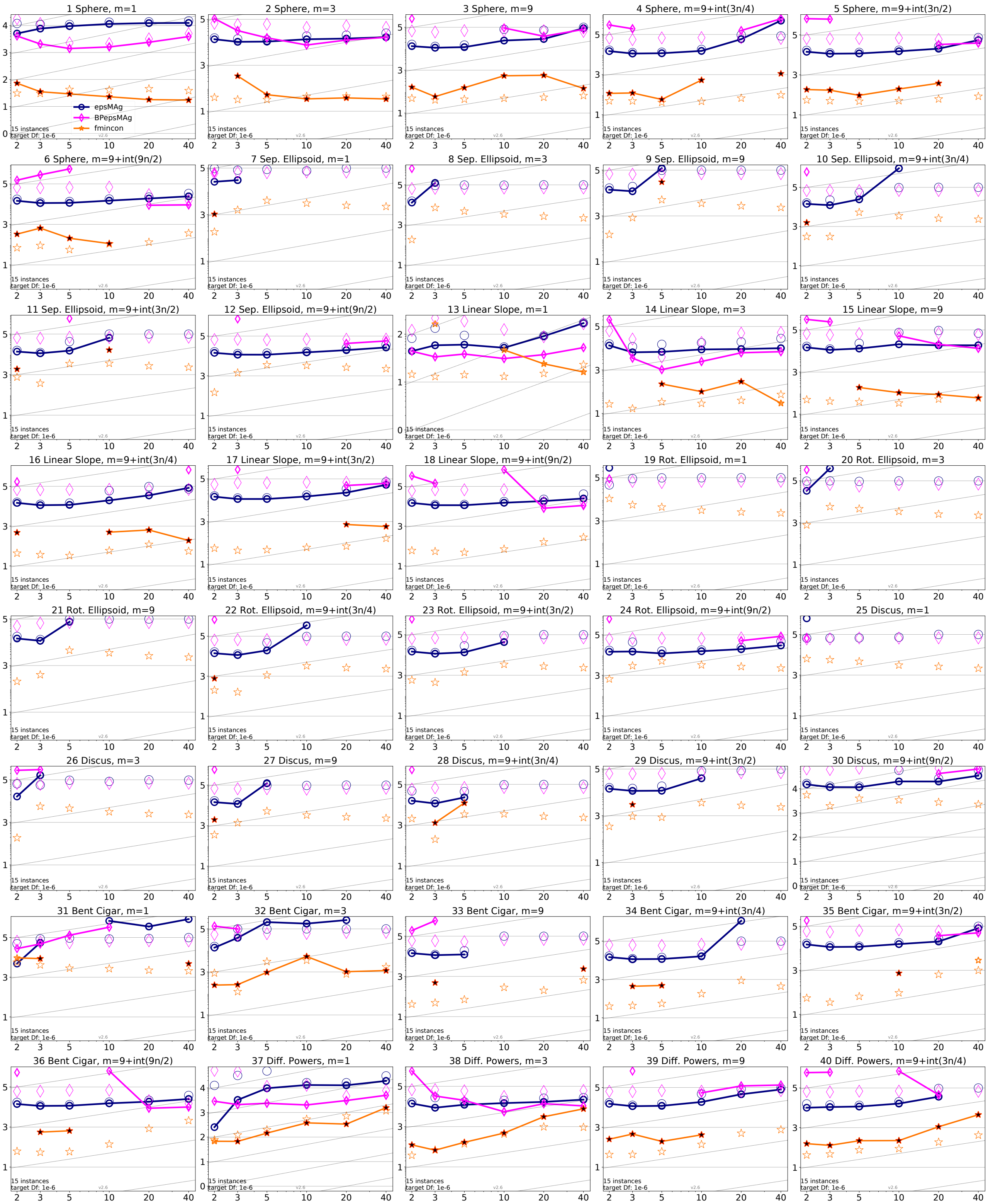

Figure 1: Expected running time (ERT in number of evaluations as $\log_{10}$ value), divided by dimension for target function value 10^{-6} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Different symbols correspond to different algorithms given in the legend of f_1. Light symbols give the maximum number of evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○: BPepsMAg, ◇: epsMAg, ⋆: fmincon.

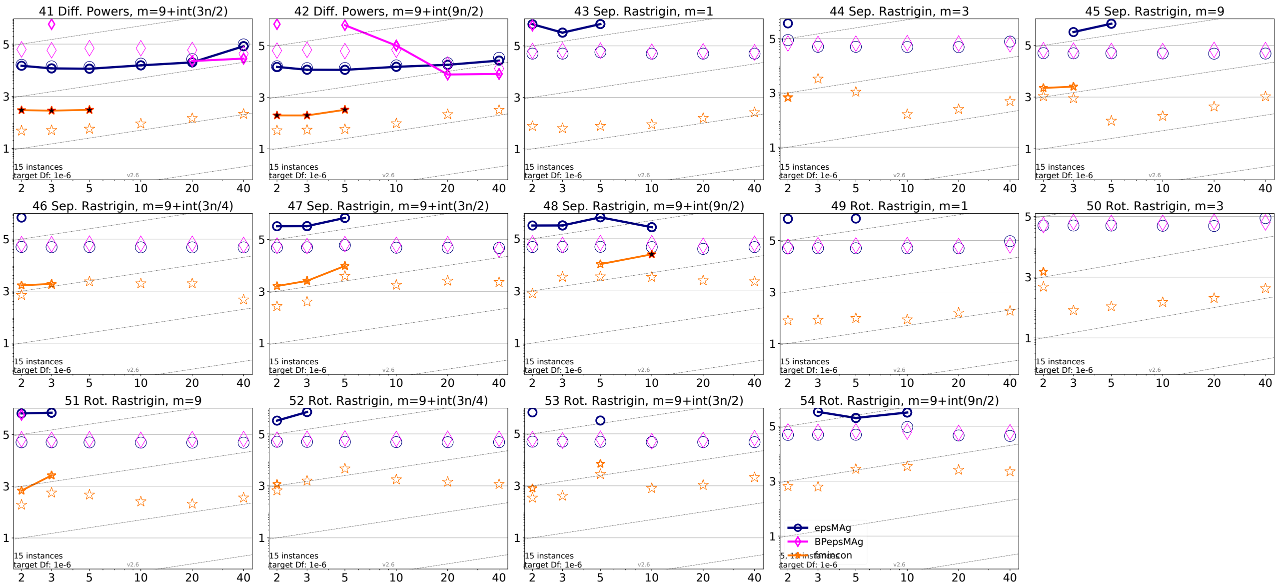

Figure 2: Expected running time (ERT in number of evaluations as $\log_{10}$ value), divided by dimension for target function value 10^{-6} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Different symbols correspond to different algorithms given in the legend of f_{54}. Light symbols give the maximum number of evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○: BPepsMAg, ◇: epsMAg, ⋆: fmincon.

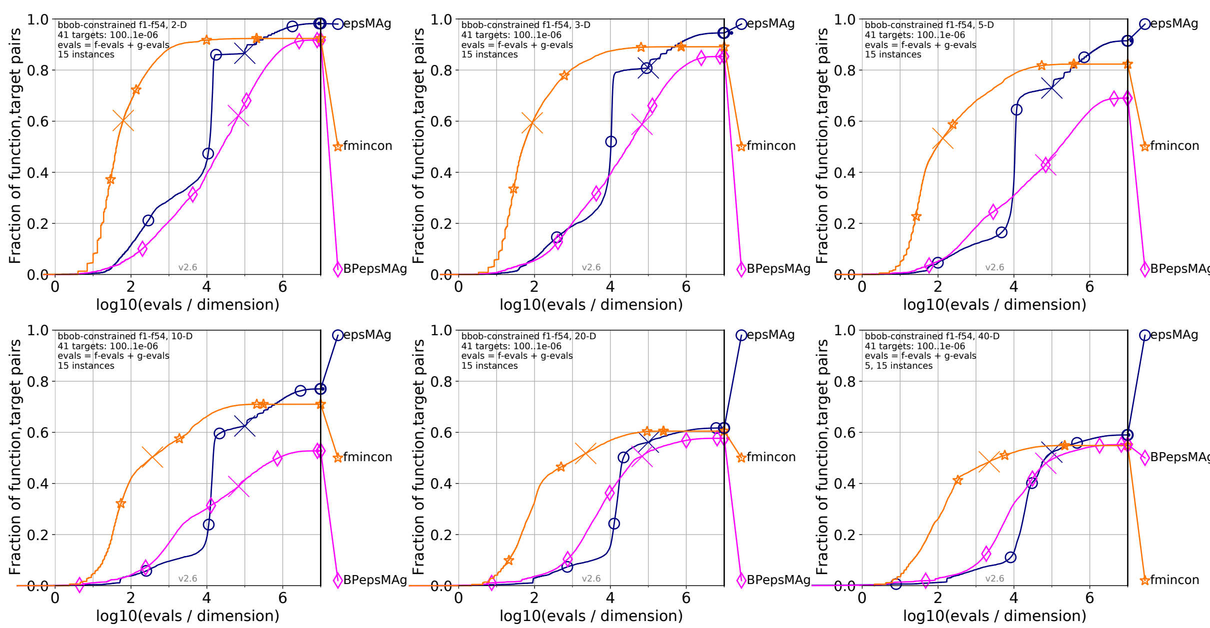

Figure 3: Runtime distributions (ECDFs) over all targets: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of evaluations, divided by dimension (evals/DIM) for the 41 targets $10^{[-6..2]}$ in dimension 2 to 40 and over all constrained test functions..

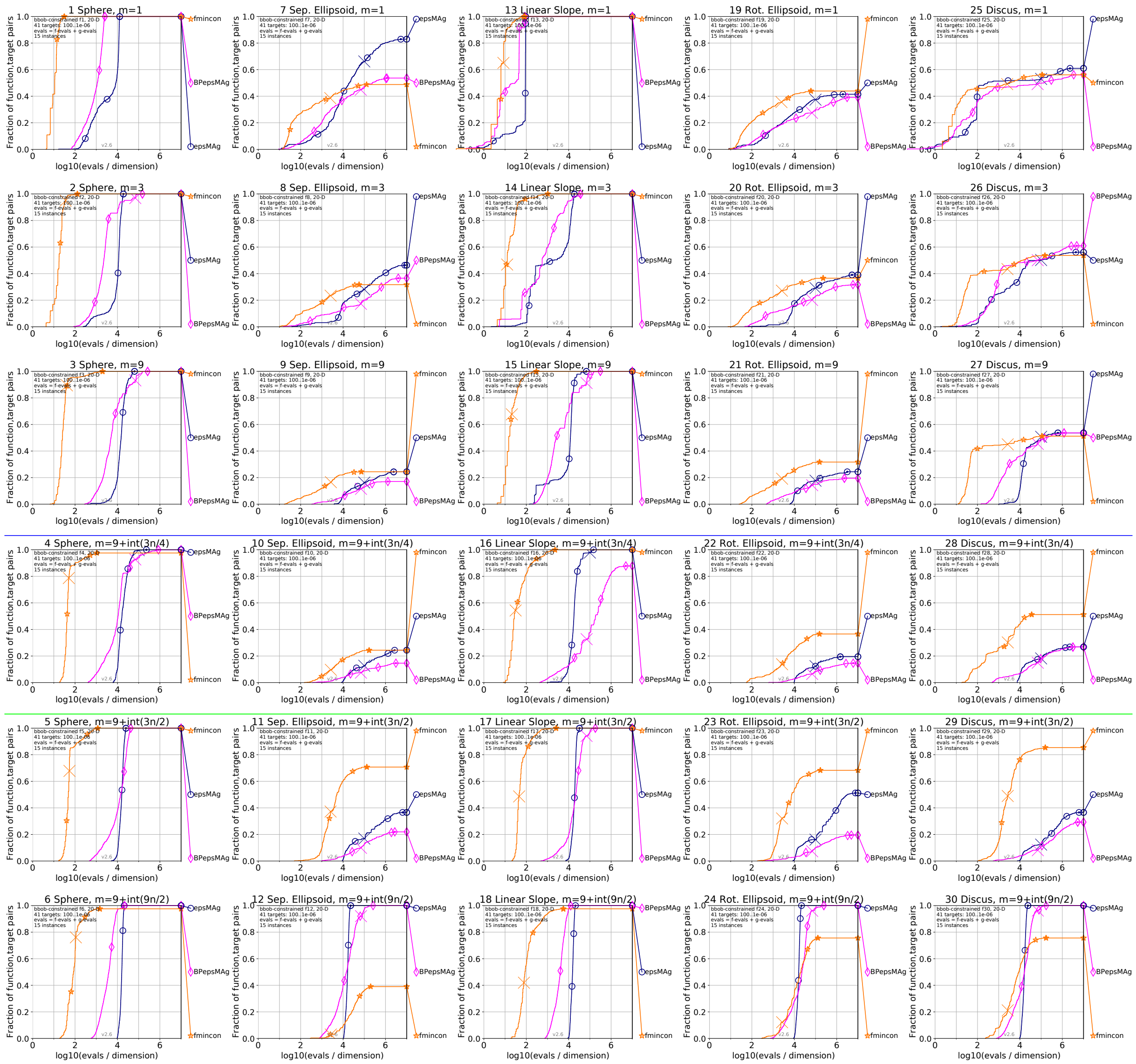

Figure 4: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of evaluations, divided by dimension (evals/DIM) for the 41 targets $10^{[-6..2]}$ in dimension 20 and problem IDs 1 to 30. Above the blue (rsp. green) line are problems where there are less overall (rsp. active) constraints than dimension.

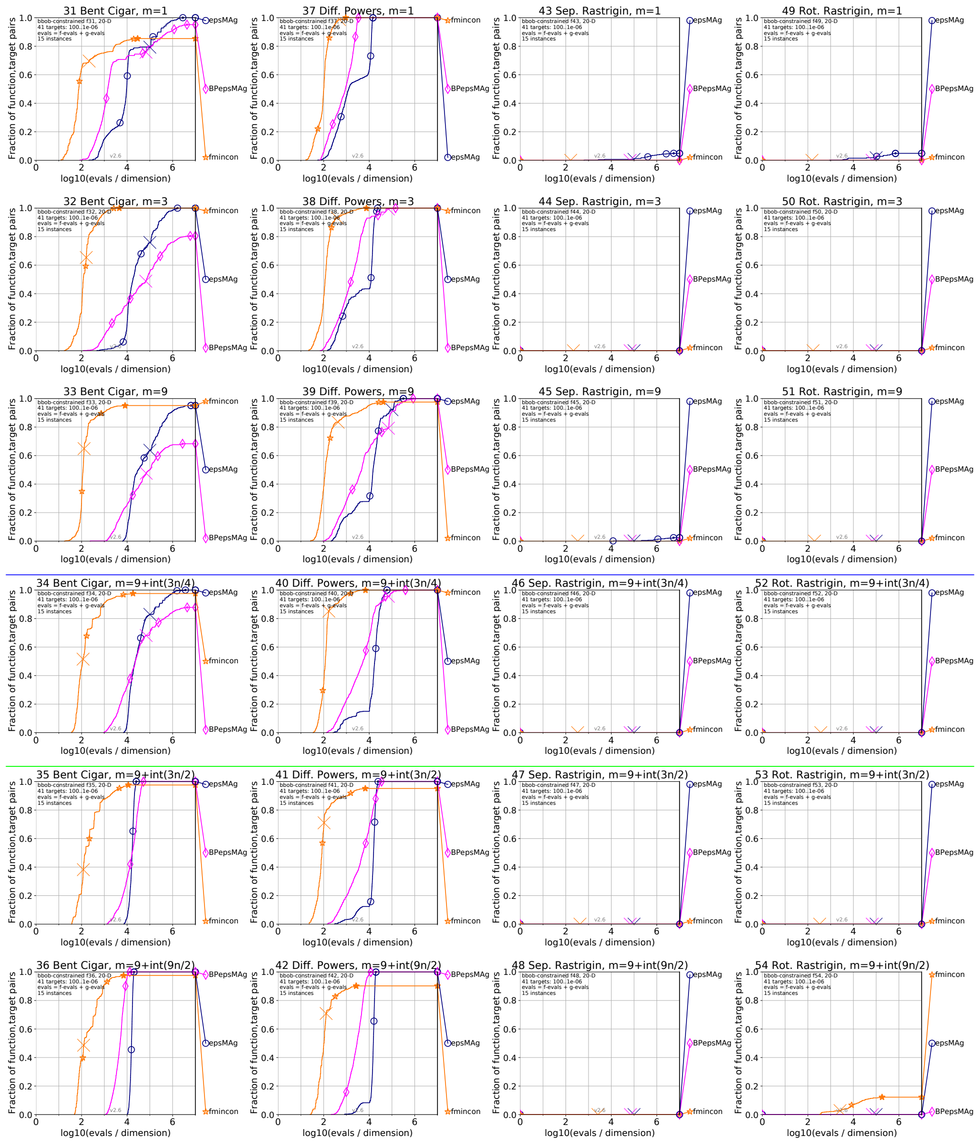

Figure 5: Empirical cumulative distribution of simulated (bootstrapped) runtimes, measured in number of evaluations, divided by dimension (evals/DIM) for the 41 targets $10^{[-6..2]}$ in dimension 20 and problem IDs 31 to 54. Above the blue (rsp. green) line are problems where there are less overall (rsp. active) constraints than dimension.

A APPENDIX

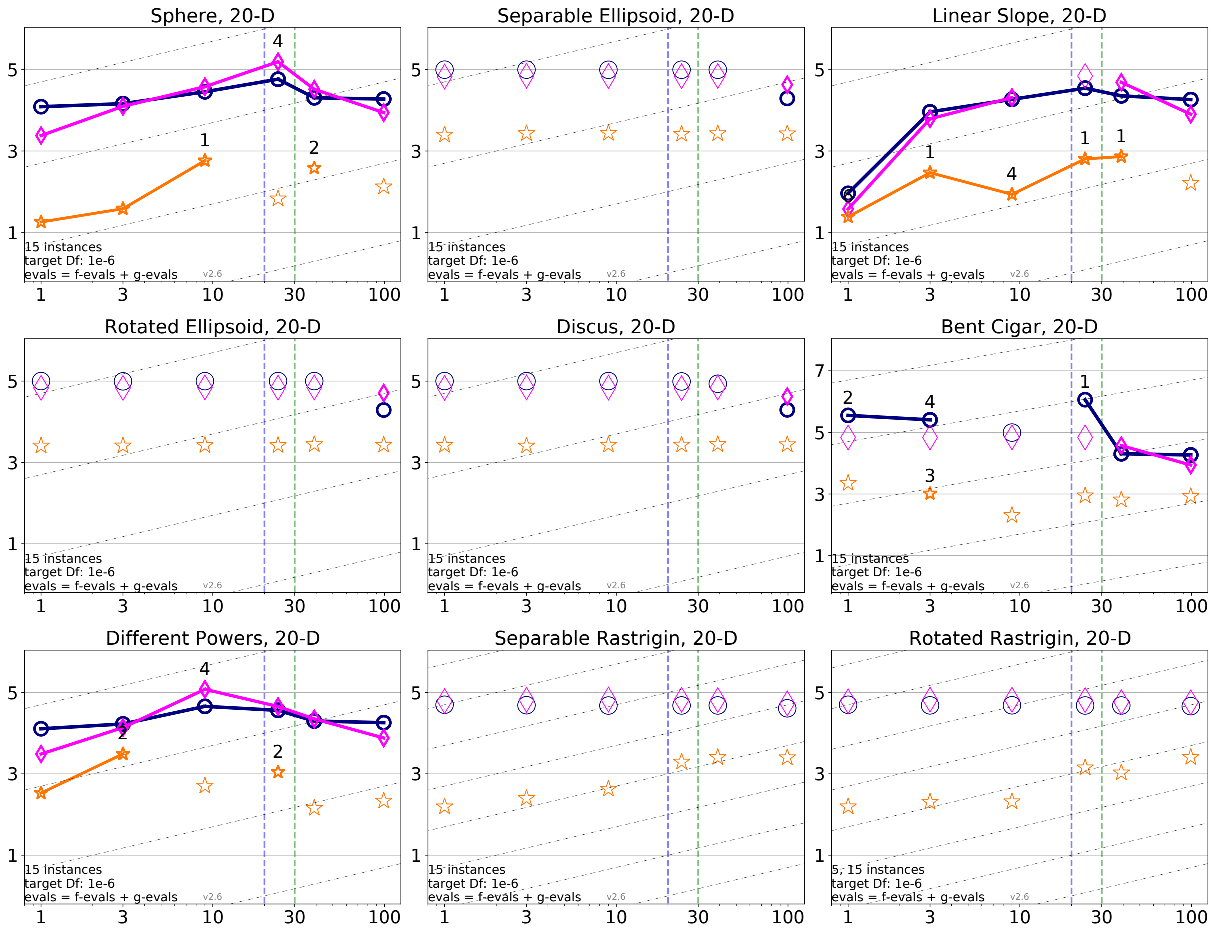

Figure 6: Scaling of ERT divided by dimension with the number of constraints for all nine raw functions in the `bbob-constrained` suite, in dimension 20 for a single target. The vertical dashed lines depict the number of active (blue) and overall number (green) of constraints.

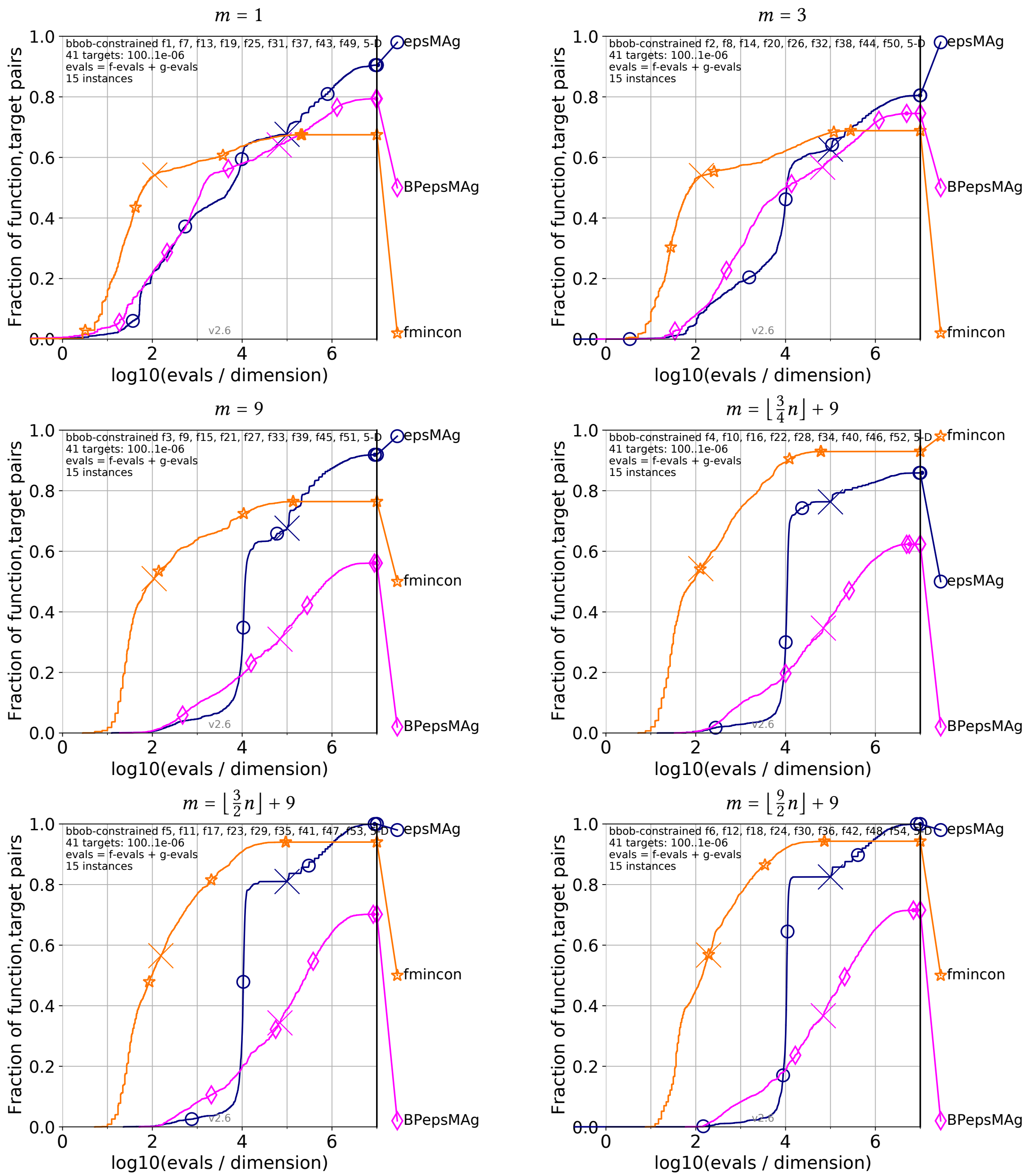

Figure 7: Bootstrapped empirical cumulative distribution of the number of evaluations divided by dimension (evals/DIM) for 41 targets with target precision in $10^{[-6..2]}$ for all functions and subgroups in 5-D.

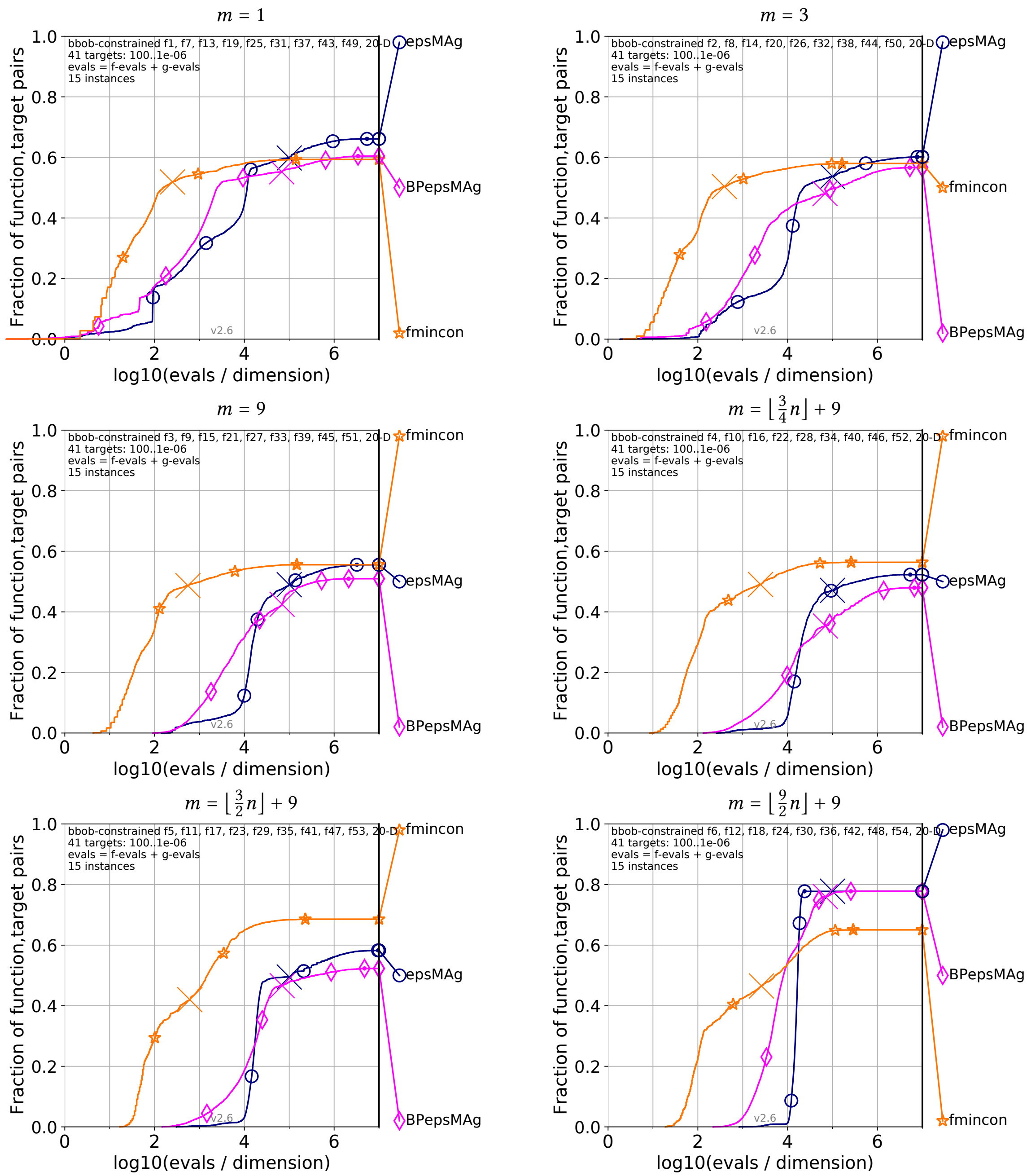

Figure 8: Bootstrapped empirical cumulative distribution of the number of evaluations divided by dimension (evals/DIM) for 41 targets with target precision in $10^{[-6..2]}$ for all functions and subgroups in 20-D.

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-6	#succ
f1								
epsMAg	8420(2365)	2.4e4(2e4)	8.5e4(6e4)	1.5e5(8e4)	2.0e5(2e4)	2.4e5(6186)	2.5e5(9092)	15/15
BPepsMA	4269(837)	9235(1168)	1.5e4(1352)	2.3e4(2330)	2.9e4(2146)	4.2e4(1776)	4.8e4(2798)	15/15
fmincon	**124**(42)$^{\star 3}$	**141**(42)$^{\star 3}$	**181**(71)$^{\star 3}$	**227**(71)$^{\star 3}$	**241**(29)$^{\star 3}$	**315**(69)$^{\star 3}$	**360**(110)$^{\star 3}$	15/15
f2								
epsMAg	1.0e5(1e5)	1.6e5(1e5)	2.1e5(7e4)	2.3e5(6e4)	2.5e5(5e4)	2.8e5(5e4)	2.9e5(5e4)	15/15
BPepsMA	1.9e4(2e4)	2.7e4(2e4)	6.7e4(9e4)	9.0e4(9e4)	1.4e5(1e5)	2.3e5(5e5)	2.5e5(5e5)	13/15
fmincon	**229**(130)$^{\star 3}$	**264**(129)$^{\star 3}$	**330**(90)$^{\star 3}$	**365**(110)$^{\star 3}$	**386**(151)$^{\star 3}$	**512**(218)$^{\star 3}$	**761**(364)$^{\star 3}$	12/15
f3								
epsMAg	1.6e5(9e4)	2.1e5(6e4)	2.9e5(1e5)	3.4e5(1e5)	3.8e5(2e5)	5.0e5(3e5)	5.7e5(3e5)	15/15
BPepsMA	3.3e4(2e4)	7.7e4(5e4)	1.4e5(1e5)	1.8e5(2e5)	2.5e5(3e5)	4.9e5(6e5)	7.6e5(1e6)	11/15
fmincon	**348**(106)$^{\star 3}$	**402**(127)$^{\star 3}$	**473**(129)$^{\star 3}$	**510**(90)$^{\star 3}$	**630**(146)$^{\star 3}$	**752**(221)$^{\star 3}$	**1.2e4**(1e4)$^{\star 3}$	1/15
f4								
epsMAg	2.1e5(4e4)	2.4e5(4e4)	2.7e5(5e4)	3.4e5(1e5)	4.3e5(1e5)	6.3e5(3e5)	**1.2e6**(8e5)	13/15
BPepsMA	3.8e4(2e4)	8.9e4(4e4)	1.6e5(5e4)	2.2e5(5e4)	2.9e5(5e4)	6.7e5(4e5)	3.1e6(3e6)	4/15
fmincon	**656**(168)$^{\star 3}$	**734**(162)$^{\star 3}$	**796**(141)$^{\star 3}$	**844**(126)$^{\star 3}$	**900**(162)$^{\star 3}$	**1130**(564)$^{\star 3}$	∞ *972*	0/15
f5								
epsMAg	2.2e5(3e4)	2.6e5(4e4)	3.0e5(5e4)	3.3e5(6e4)	3.6e5(7e4)	3.9e5(7e4)	4.1e5(7e4)	15/15
BPepsMA	5.2e4(2e4)	1.3e5(4e4)	2.1e5(5e4)	3.1e5(9e4)	3.9e5(1e5)	5.8e5(2e5)	6.6e5(1e5)	15/15
fmincon	**792**(252)$^{\star 3}$	**850**(252)$^{\star 3}$	**887**(210)$^{\star 3}$	**906**(231)$^{\star 3}$	**965**(231)$^{\star 3}$	**3800**(5584)$^{\star 3}$	**7622**(1e4)$^{\star 3}$	2/15
f6								
epsMAg	2.5e5(3e4)	2.9e5(4e4)	3.1e5(3e4)	3.2e5(3e4)	3.4e5(3e4)	3.6e5(3e4)	3.8e5(3e4)	15/15
BPepsMA	2.9e4(8203)	4.5e4(1e4)	6.4e4(2e4)	8.6e4(2e4)	1.0e5(2e4)	1.5e5(3e4)	**1.8e5**(5e4)	15/15
fmincon	**1360**(714)$^{\star 3}$	**1483**(693)$^{\star 3}$	**1522**(672)$^{\star 3}$	**1576**(693)$^{\star 3}$	**1576**(693)$^{\star 3}$	**4038**(5898)	∞ *1598*	0/15
f7								
epsMAg	2.9e4(3e4)	7.5e4(3e4)	2.4e5(1e5)	**6.1e5**(8e5)	**1.5e6**(1e6)	∞	∞ *2e6*	0/15
BPepsMA	8708(5218)	3.5e4(2e4)	3.7e5(3e5)	3.9e6(5e6)	∞	∞	∞ *7e5*	0/15
fmincon	**968**(493)$^{\star 3}$	**4414**(7001)$^{\star 3}$	**3.9e4**(3e4)*	∞	∞	∞	∞ *4e4*	0/15
f8								
epsMAg	2.1e5(2e5)	1.7e6(2e6)	**2.0e7**(2e7)	∞	∞	∞	∞ *1e6*	0/15
BPepsMA	6.7e5(1e6)	6.7e6(9e6)	∞	∞	∞	∞	∞ *9e5*	0/15
fmincon	**1.3e4**(2e4)*	**1.4e5**(2e5)$^{\star 3}$	∞	∞	∞	∞	∞ *4e4*	0/15
f9								
epsMAg	1.1e6(1e6)	∞	∞	∞	∞	∞	∞ *1e6*	0/15
BPepsMA	2.2e6(2e6)	∞	∞	∞	∞	∞	∞ *1e6*	0/15
fmincon	**3.4e4**(8e4)*	∞	∞	∞	∞	∞	∞ *5e4*	0/15
f10								
epsMAg	3.2e6(3e6)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BPepsMA	1.3e7(1e7)	∞	∞	∞	∞	∞	∞ *9e5*	0/15
fmincon	**1.1e5**(1e5)	∞	∞	∞	∞	∞	∞ *5e4*	0/15
f11								
epsMAg	1.7e6(2e6)	1.0e7(1e7)	∞	∞	∞	∞	∞ *2e6*	0/15
BPepsMA	2.8e6(3e6)	∞	∞	∞	∞	∞	∞ *1e6*	0/15
fmincon	**4.2e4**(3e4)	**5.1e4**(5e4)	**6.1e4**(3e4)	**8.2e4**(5e4)	**2.0e5**(9e4)	∞	∞ *4e4*	0/15
f12								
epsMAg	2.6e5(3e4)	2.8e5(4e4)	**3.1e5**(4e4)	**3.3e5**(3e4)	**3.4e5**(3e4)	**3.8e5**(3e4)	**3.9e5**(4e4)	15/15
BPepsMA	**1.3e5**(2e5)	**2.2e5**(3e5)	3.3e5(4e5)	5.2e5(9e5)	6.1e5(7e5)	7.8e5(9e5)	8.5e5(7e5)	13/15
fmincon	6.2e5(5e5)	6.3e5(4e5)	6.4e5(5e5)	∞	∞	∞	∞ *4e4*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-6	#succ
f13								
epsMAg	613(494)	1747(232)	1792(31)	1822(25)	1822(25)	1822(25)	1822(25)	15/15
BPepsMA	143(152)	394(422)	680(426)	748(787)	748(787)	748(787)	748(787)	15/15
fmincon	**72**(42)	**86**(42)	**95**(42)	**117**(42)	**151**(21)	**242**(148)	**481**(744)	6/15
f14								
epsMAg	3.7e4(6e4)	7.5e4(9e4)	1.0e5(1e5)	1.3e5(1e5)	1.6e5(2e5)	1.8e5(2e5)	1.8e5(2e5)	15/15
BPepsMA	5641(4709)	9488(8088)	1.6e4(1e4)	4.6e4(8e4)	5.7e4(8e4)	1.2e5(2e5)	1.2e5(2e5)	15/15
fmincon	**245**(239)$^{\star 2}$	**263**(218)$^{\star 2}$	**316**(239)$^{\star 2}$	**319**(239)$^{\star 2}$	**341**(239)$^{\star 2}$	**503**(268)*	**5860**(5416)*	1/15
f15								
epsMAg	1.2e5(1e5)	1.8e5(1e5)	2.2e5(1e5)	2.4e5(2e5)	2.6e5(2e5)	3.0e5(2e5)	3.7e5(4e5)	15/15
BPepsMA	3.5e4(6e4)	1.5e5(3e5)	1.8e5(4e5)	2.3e5(4e5)	3.3e5(5e5)	4.1e5(4e5)	4.1e5(5e5)	12/15
fmincon	**255**(181)$^{\star 3}$	**296**(202)$^{\star 3}$	**346**(267)$^{\star 3}$	**366**(287)$^{\star 3}$	**405**(287)$^{\star 3}$	**971**(694)$^{\star 3}$	**1719**(1494)$^{\star 3}$	4/15
f16								
epsMAg	2.1e5(8e4)	3.2e5(1e5)	4.0e5(2e5)	4.2e5(2e5)	4.4e5(2e5)	5.9e5(3e5)	7.0e5(4e5)	14/15
BPepsMA	2.4e5(3e5)	2.5e6(4e6)	4.4e6(4e6)	4.4e6(7e6)	4.4e6(6e6)	1.4e7(2e7)	∞ *9e5*	0/15
fmincon	**454**(273)$^{\star 3}$	**653**(580)$^{\star 3}$	**723**(601)$^{\star 3}$	**776**(643)$^{\star 3}$	**807**(643)$^{\star 3}$	**2549**(2840)$^{\star 2}$	**1.3e4**(1e4)$^{\star 2}$	1/15
f17								
epsMAg	2.8e5(4e4)	3.5e5(6e4)	3.7e5(7e4)	3.9e5(7e4)	4.1e5(7e4)	4.4e5(7e4)	4.5e5(7e4)	15/15
BPepsMA	1.5e5(8e4)	3.2e5(2e5)	4.6e5(2e5)	5.4e5(3e5)	6.2e5(3e5)	8.8e5(8e5)	9.8e5(9e5)	13/15
fmincon	**789**(189)$^{\star 3}$	**867**(233)$^{\star 3}$	**895**(233)$^{\star 3}$	**915**(212)$^{\star 3}$	**1175**(464)$^{\star 3}$	**3636**(2558)$^{\star 3}$	**1.5e4**(1e4)$^{\star 3}$	1/15
f18								
epsMAg	2.4e5(2e4)	2.6e5(3e4)	2.9e5(4e4)	3.1e5(4e4)	3.2e5(4e4)	3.5e5(3e4)	3.7e5(3e4)	15/15
BPepsMA	3.8e4(2e4)	5.3e4(2e4)	7.0e4(3e4)	8.6e4(3e4)	1.0e5(4e4)	1.4e5(6e4)	**1.6e5**(7e4)	15/15
fmincon	**1441**(966)$^{\star 3}$	**1570**(882)$^{\star 3}$	**1595**(882)$^{\star 3}$	**1621**(903)$^{\star 3}$	**1744**(903)$^{\star 3}$	**8245**(6894)	∞ *1537*	0/15
f19								
epsMAg	2.3e4(2e4)	1.8e5(2e5)	3.0e6(3e6)	∞	∞	∞	∞ *2e6*	0/15
BPepsMA	1.9e4(2e4)	1.0e6(3e6)	1.3e7(7e6)	∞	∞	∞	∞ *8e5*	0/15
fmincon	**1032**(848)$^{\star 3}$	**1.1e4**(2e4)$^{\star 2}$	**6.9e4**(9e4)$^{\star 2}$	∞	∞	∞	∞ *4e4*	0/15
f20								
epsMAg	1.7e5(9e4)	8.8e5(7e5)	**2.1e7**(2e7)	∞	∞	∞	∞ *1e6*	0/15
BPepsMA	1.9e5(2e5)	3.8e6(3e6)	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	**2818**(1951)$^{\star 3}$	**6.3e4**(8e4)$^{\star 3}$	∞	∞	∞	∞	∞ *5e4*	0/15
f21								
epsMAg	6.2e5(4e5)	∞	∞	∞	∞	∞	∞ *1e6*	0/15
BPepsMA	1.4e6(1e6)	∞	∞	∞	∞	∞	∞ *9e5*	0/15
fmincon	**2.1e4**(3e4)$^{\star 2}$	**3.0e5**(3e5)$^{\star 2}$	∞	∞	∞	∞	∞ *4e4*	0/15
f22								
epsMAg	2.9e6(5e6)	∞	∞	∞	∞	∞	∞ *2e6*	0/15
BPepsMA	1.4e7(3e7)	∞	∞	∞	∞	∞	∞ *9e5*	0/15
fmincon	**8.1e4**(9e4)$^{\star 2}$	**3.1e5**(2e5)$^{\star 2}$	∞	∞	∞	∞	∞ *5e4*	0/15
f23								
epsMAg	1.9e6(2e6)	9.9e6(9e6)	1.0e7(9e6)	2.1e7(3e7)	∞	∞	∞ *1e6*	0/15
BPepsMA	7.6e6(1e7)	∞	∞	∞	∞	∞	∞ *1e6*	0/15
fmincon	**4.3e4**(2e4)*	**7.2e4**(1e5)*	**1.1e5**(7e4)*	**1.1e5**(1e5)*	**6.9e5**(5e5)*	∞	∞ *5e4*	0/15
f24								
epsMAg	2.5e5(3e4)	2.8e5(4e4)	3.1e5(3e4)	3.2e5(3e4)	3.4e5(3e4)	**3.7e5**(3e4)	**3.9e5**(3e4)	15/15
BPepsMA	**1.9e5**(2e5)	3.0e5(3e5)	3.9e5(3e5)	4.8e5(3e5)	6.3e5(4e5)	9.3e5(8e5)	1.0e6(6e5)	13/15
fmincon	2.8e5(4e5)	**2.8e5**(2e5)	**2.9e5**(2e5)	**2.9e5**(4e5)	**2.9e5**(3e5)	∞	∞ *4e4*	0/15

Table 1: Expected runtime (ERT) to reach given targets, measured in number of evaluations, in dimension 20. For each function, the ERT and, in braces as dispersion measure, the half difference between 10 and 90%-tile of (bootstrapped) runtimes is shown for the different target precision values as shown in the top row. #succ is the number of trials that reached the last target $f_{\text{opt}} + 10^{-6}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of functions (54). A ↓ indicates the same tested against . Best results are printed in bold. Data produced with COCO v2.6

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-6	#succ
f25								
epsMAg	548(468)	1672(481)	1892(72)	7.8e5(1e6)	∞	∞	∞ *9e5*	0/15
BPepsMA	314(408)	1065(876)	4117(4852)	4.3e6(3e6)	∞	∞	∞ *8e5*	0/15
fmincon	**158**(172)	**206**(230)*	**246**(209)$^{\star 3}$	**1.2e5**(2e5)	∞	∞	∞ *4e4*	0/15
f26								
epsMAg	5.6e4(9e4)	1.0e5(1e5)	1.5e5(2e5)	2.0e6(2e6)	∞	∞	∞ *8e5*	0/15
BPepsMA	7640(7217)	1.3e4(9133)	3.2e4(6e4)	2.5e6(2e6)	∞	∞	∞ *9e5*	0/15
fmincon	**543**(312)$^{\star 3}$	**635**(316)$^{\star 3}$	**7333**(1e4)*	**2.4e5**(3e5)	∞	∞	∞ *4e4*	0/15
f27								
epsMAg	2.9e5(2e5)	4.7e5(6e5)	5.4e5(9e5)	1.5e6(2e6)	∞	∞	∞ *1e6*	0/15
BPepsMA	1.0e5(2e5)	2.2e5(3e5)	4.3e5(7e5)	1.7e6(1e6)	∞	∞	∞ *6e5*	0/15
fmincon	**831**(292)$^{\star 3}$	**930**(326)$^{\star 3}$	**1025**(325)$^{\star 3}$	**4.8e5**(7e5)	∞	∞	∞ *3e4*	0/15
f28								
epsMAg	9.0e5(1e6)	1.1e7(1e7)	∞	∞	∞	∞	∞ *2e6*	0/15
BPepsMA	1.2e6(1e6)	1.6e7(2e7)	∞	∞	∞	∞	∞ *1e6*	0/15
fmincon	**1.7e4**(2e4)$^{\star 2}$	**4.8e4**(5e4)$^{\star 2}$	**8.3e4**(7e4)$^{\star 2}$	**2.8e5**(2e5)$^{\star 2}$	∞	∞	∞ *4e4*	0/15
f29								
epsMAg	3.1e6(2e6)	8.9e6(7e6)	∞	∞	∞	∞	∞ *1e6*	0/15
BPepsMA	4.4e6(6e6)	1.5e7(1e7)	∞	∞	∞	∞	∞ *1e6*	0/15
fmincon	**2.4e4**(2e4)*	**4.5e4**(5e4)*	**6.5e4**(5e4)*	**6.8e4**(5e4)*	**7.9e4**(6e4)*	∞	∞ *4e4*	0/15
f30								
epsMAg	2.6e5(5e4)	2.9e5(5e4)	3.1e5(5e4)	3.2e5(4e4)	3.4e5(4e4)	**3.8e5**(5e4)	**3.9e5**(5e4)	15/15
BPepsMA	1.1e5(1e5)	1.8e5(2e5)	2.7e5(2e5)	4.4e5(2e5)	5.4e5(3e5)	7.3e5(3e5)	8.4e5(4e5)	14/15
fmincon	**9.0e4**(7e4)	**1.4e5**(1e5)	**1.4e5**(1e5)	**1.4e5**(1e5)	**1.9e5**(2e5)	∞	∞ *5e4*	0/15
f31								
epsMAg	2.2e4(1e4)	1.0e5(5e4)	1.5e5(5e4)	1.9e5(3e4)	3.1e5(2e4)	**3.1e6**(4e6)	**7.1e6**(1e7)	2/15
BPepsMA	8084(1883)	1.5e4(3236)	2.1e4(4185)	2.9e4(3244)	2.3e5(5e5)	5.1e6(6e6)	∞ *6e5*	0/15
fmincon	**651**(321)$^{\star 3}$	**1060**(368)$^{\star 3}$	**1314**(427)$^{\star 3}$	**1547**(353)$^{\star 3}$	**1.2e4**(2e4)$^{\star 3}$	∞	∞ *2555*	0/15
f32								
epsMAg	1.9e5(8e4)	2.6e5(1e5)	3.7e5(2e5)	4.7e5(4e5)	1.1e6(7e5)	3.5e6(4e6)	5.1e6(6e6)	4/15
BPepsMA	5.3e4(5e4)	2.1e5(3e5)	5.9e5(6e5)	1.3e6(1e6)	3.7e6(5e6)	∞	∞ *7e5*	0/15
fmincon	**1271**(550)$^{\star 3}$	**2107**(962)$^{\star 3}$	**2680**(864)$^{\star 3}$	**3135**(889)$^{\star 3}$	**3282**(1008)$^{\star 3}$	**8762**(6870)$^{\star 3}$	**2.1e4**(3e4)$^{\star 3}$	3/15
f33								
epsMAg	2.7e5(1e5)	3.5e5(1e5)	5.7e5(8e5)	9.1e5(7e5)	2.3e6(2e6)	6.7e6(7e6)	∞ *1e6*	0/15
BPepsMA	1.8e5(3e5)	3.9e5(7e5)	7.0e5(8e5)	2.3e6(2e6)	4.5e6(3e6)	∞	∞ *9e5*	0/15
fmincon	**1871**(823)$^{\star 3}$	**2199**(591)$^{\star 3}$	**2374**(888)$^{\star 3}$	**2455**(883)$^{\star 3}$	**2534**(811)$^{\star 3}$	**7813**(6514)$^{\star 3}$	∞ *2503*	0/15
f34								
epsMAg	2.3e5(3e4)	2.7e5(5e4)	3.3e5(9e4)	4.8e5(1e5)	7.3e5(3e5)	7.2e6(5e6)	**2.3e7**(2e7)	1/15
BPepsMA	7.5e4(3e4)	1.6e5(1e5)	2.7e5(2e5)	5.6e5(3e5)	1.3e6(2e6)	1.5e7(2e7)	∞ *1e6*	0/15
fmincon	**2717**(2406)$^{\star 3}$	**2810**(2385)$^{\star 3}$	**2918**(2364)$^{\star 3}$	**3081**(2314)$^{\star 3}$	**3929**(6145)$^{\star 3}$	**9012**(9027)*	∞ *2290*	0/15
f35								
epsMAg	2.4e5(5e4)	2.9e5(6e4)	3.2e5(7e4)	3.4e5(8e4)	3.6e5(7e4)	3.9e5(7e4)	**4.1e5**(7e4)	15/15
BPepsMA	1.1e5(5e4)	1.7e5(8e4)	2.8e5(2e5)	3.7e5(2e5)	4.6e5(2e5)	6.6e5(2e5)	7.4e5(2e5)	15/15
fmincon	**3583**(3119)$^{\star 3}$	**3646**(3119)$^{\star 3}$	**3692**(3140)$^{\star 3}$	**3720**(3119)$^{\star 3}$	**3749**(3140)$^{\star 3}$	**5.7e4**(5e4)	∞ *2491*	0/15
f36								
epsMAg	2.7e5(3e4)	2.8e5(3e4)	3.0e5(2e4)	3.1e5(2e4)	3.3e5(2e4)	3.6e5(2e4)	3.7e5(2e4)	15/15
BPepsMA	4.8e4(1e4)	6.7e4(2e4)	8.6e4(2e4)	1.0e5(3e4)	1.2e5(4e4)	1.6e5(4e4)	**1.7e5**(4e4)	15/15
fmincon	**3992**(4068)$^{\star 3}$	**4085**(4110)$^{\star 3}$	**4096**(4131)$^{\star 3}$	**4438**(4110)*	**4446**(5330)*	**1.3e4**(2e4)	∞ *2396*	0/15
f37								
epsMAg	5317(1354)	9612(1687)	1.6e4(3146)	5.2e4(8e4)	1.9e5(1e5)	2.4e5(3e4)	2.6e5(3e4)	15/15
BPepsMA	2831(238)	5151(801)	1.1e4(1612)	1.8e4(3960)	2.7e4(4590)	4.9e4(5780)	6.1e4(4752)	15/15
fmincon	**755**(218)$^{\star 3}$	**1327**(353)$^{\star 3}$	**1806**(213)$^{\star 3}$	**2025**(190)$^{\star 3}$	**2221**(216)$^{\star 3}$	**3586**(1690)$^{\star 3}$	**6747**(5326)$^{\star 3}$	13/15
f38								
epsMAg	8082(1059)	1.8e4(1e4)	8.4e4(1e5)	2.3e5(2e5)	2.8e5(6e4)	3.2e5(6e4)	3.4e5(6e4)	15/15
BPepsMA	5340(2156)	1.5e4(3592)	2.8e4(4e4)	4.4e4(4e4)	1.2e5(2e5)	2.2e5(5e5)	2.7e5(5e5)	14/15
fmincon	**1132**(364)$^{\star 3}$	**1695**(257)$^{\star 3}$	**2041**(247)$^{\star 3}$	**2281**(290)$^{\star 3}$	**2668**(381)$^{\star 3}$	**5944**(2874)$^{\star 3}$	**6.2e4**(5e4)$^{\star 3}$	2/15
f39								
epsMAg	2.8e4(2e4)	1.1e5(1e5)	3.1e5(7e4)	4.5e5(5e5)	6.1e5(7e5)	8.8e5(1e6)	**9.1e5**(9e5)	12/15
BPepsMA	8312(2480)	2.9e4(3e4)	1.1e5(2e5)	3.0e5(4e5)	4.3e5(6e5)	1.0e6(1e6)	2.4e6(3e6)	4/15
fmincon	**1267**(241)$^{\star 3}$	**1638**(245)$^{\star 3}$	**2173**(397)$^{\star 3}$	**2484**(582)$^{\star 3}$	**2962**(784)$^{\star 3}$	**9975**(8577)$^{\star 3}$	∞ *6089*	0/15

Δf_{opt}	1e1	1e0	1e-1	1e-2	1e-3	1e-5	1e-6	#succ
f40								
epsMAg	1.5e5(1e5)	2.2e5(1e5)	3.0e5(5e4)	3.8e5(1e5)	4.3e5(1e5)	6.0e5(3e5)	7.3e5(3e5)	15/15
BPepsMA	1.1e4(4789)	2.8e4(2e4)	5.5e4(2e4)	1.1e5(3e4)	2.1e5(1e5)	5.9e5(4e5)	9.1e5(7e5)	10/15
fmincon	**1336**(358)$^{\star 3}$	**1761**(398)$^{\star 3}$	**2115**(443)$^{\star 3}$	**2535**(529)$^{\star 3}$	**2658**(466)$^{\star 3}$	**5497**(4322)$^{\star 3}$	**2.2e4**(4e4)$^{\star 3}$	2/15
f41								
epsMAg	1.6e5(1e5)	2.7e5(3e4)	2.9e5(3e4)	3.2e5(5e4)	3.4e5(5e4)	3.8e5(7e4)	**4.0e5**(6e4)	15/15
BPepsMA	1.3e4(4711)	3.2e4(9626)	7.3e4(4e4)	1.3e5(5e4)	1.9e5(1e5)	3.6e5(2e5)	4.5e5(2e5)	15/15
fmincon	**966**(378)$^{\star 3}$	**1253**(356)$^{\star 3}$	**1515**(338)$^{\star 3}$	**1708**(334)$^{\star 3}$	**1814**(321)$^{\star 3}$	**4137**(2869)$^{\star 3}$	∞ *1907*	0/15
f42								
epsMAg	2.4e5(3e4)	2.7e5(3e4)	2.9e5(3e4)	3.0e5(3e4)	3.2e5(3e4)	3.5e5(3e4)	3.6e5(3e4)	15/15
BPepsMA	1.8e4(6220)	2.9e4(9130)	4.6e4(2e4)	6.2e4(3e4)	8.0e4(3e4)	1.2e5(4e4)	**1.5e5**(5e4)	15/15
fmincon	**1564**(777)$^{\star 3}$	**1800**(756)$^{\star 3}$	**1861**(798)$^{\star 3}$	**1990**(799)$^{\star 3}$	**2074**(778)$^{\star 3}$	**1.6e4**(3e4)$^{\star 2}$	∞ *2259*	0/15
f43								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *1e6*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *1643*	0/15
f44								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *2406*	0/15
f45								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *3718*	0/15
f46								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *3998*	0/15
f47								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *8e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *4817*	0/15
f48								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *6e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *3e4*	0/15
f49								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *1517*	0/15
f50								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *1842*	0/15
f51								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *2690*	0/15
f52								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *8e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *5504*	0/15
f53								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *9e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *3948*	0/15
f54								
epsMAg	∞	∞	∞	∞	∞	∞	∞ *7e5*	0/15
BPepsMA	∞	∞	∞	∞	∞	∞	∞ *6e5*	0/15
fmincon	∞	∞	∞	∞	∞	∞	∞ *4e4*	0/15

Table 2: Expected runtime (ERT) to reach given targets, measured in number of evaluations, in dimension 20. For each function, the ERT and, in braces as dispersion measure, the half difference between 10 and 90%-tile of (bootstrapped) runtimes is shown for the different target precision values as shown in the top row. #succ is the number of trials that reached the last target $f_{opt} + 10^{-6}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of functions (54). A ↓ indicates the same tested against . Best results are printed in bold. Data produced with COCO v2.6