

Comparison of Cauchy EDA and G3PCX Algorithms on the BBOB Noiseless Testbed

Petr Pošík
Czech Technical University in Prague
Faculty of Electrical Engineering, Department of Cybernetics
Technická 2, 166 27 Prague 6, Czech Republic
posik@labe.felk.cvut.cz

ABSTRACT

Estimation-of-distribution algorithm equipped with Cauchy sampling distribution is compared with the generalized generation gap algorithm with parent centric crossover. Both algorithms were already presented at the 2009 black-box optimization benchmarking workshop where they often showed similar performance. This paper compares them in more detail and adds to the understanding of their key features and differences.

Categories and Subject Descriptors

G.1.6 [Numerical Analysis]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [Analysis of Algorithms and Problem Complexity]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Benchmarking, Black-box optimization, Generalized generation gap, Parent-centric crossover, Estimation-of-distribution algorithm, Cauchy distribution

1. INTRODUCTION

Black-box optimization benchmarking (BBOB) workshop in 2009 set the stage for unified comparisons among black-box optimization algorithms. The 2010 issue of the BBOB methodology [4] added means for a detailed comparison of 2 algorithms. In this paper, two algorithms from the BBOB 2009 workshop are further compared. Data for both algorithms were taken from 2009 benchmarking, but the comparison is made using the new post-processing scripts and templates for BBOB 2010. Both algorithms fall into the class of evolutionary optimization algorithms and both use unimodal distribution to sample new candidate solutions.

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.
Copyright 2010 ACM 978-1-4503-0073-5/10/07 ...\$10.00.

Yet there are many differences between them. The two algorithms chosen for the comparison are:

- The generalized generation gap (G3) model introduced by Deb in [1] and used with the parent centric crossover operator (PCX) introduced in [2]. The performance of the G3PCX algorithm on the BBOB 2009 noiseless test suite was reported in [8].
- The estimation-of-distribution algorithm (EDA) with Cauchy sampling distribution (Cauchy EDA) [7].

Both algorithms can be described as local search techniques¹. In the final algorithm presentation at the BBOB 2009 workshop, neither of the two algorithms belonged to the top-notch algorithms; nevertheless their results were often very similar despite the differences between the two algorithms. This paper emphasizes the differences between them and their impact on the search performance.

In the next section, both algorithms are shortly described and their differences are emphasized. Sec. 3 contains all the results used to compare the algorithms and their discussion. After the presentation of the time demands of both algorithms in Sec. 4, Sec. 5 concludes the paper.

2. ALGORITHM PRESENTATION

The exact descriptions of the algorithms along with the parameter settings can be found in [8] and [7], respectively. The algorithms differ foremost in the following aspects:

- The sampling distribution in the used variant of G3PCX is Gaussian and centered around the best solution found so far, while the EDA uses Cauchy distribution centered around the mean of selected parents.
- G3PCX is a steady-state model, with small incremental changes made to the population each generation. Cauchy EDA uses generational scheme where the whole population is replaced by the offspring each generation.
- While Cauchy EDA uses relatively large set of selected individuals to estimate the shape of the distribution, G3PCX uses only 3 parents to compute the distribution parameters.

¹Note that the local-search behaviour is not generally a feature of the G3PCX algorithm. Only the particular implementation of this algorithm used in this comparison is such local-search heavy.

To fight the premature convergence, Cauchy EDA uses a constant multiplier to enlarge the variance of the distribution (as suggested in [6]). In case of G3PCX no such remedies are used. For both algorithms, the crafting effort $\text{CrE} = 0$.

3. RESULTS

Results from experiments according to [4] on the benchmark functions given in [3, 5] are presented in Figures 1, 2 and 3 and in Table 1. The **expected running time (ERT)**, used in the figures and table, depends on a given target function value, $f_t = f_{\text{opt}} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach f_t , summed over all trials and divided by the number of trials that actually reached f_t [4, 9]. **Statistical significance** is tested with the rank-sum test for a given target Δf_t (10^{-8} in Figure 1) using, for each trial, either the number of needed function evaluations to reach Δf_t (inverted and multiplied by -1), or, if the target was not reached, the best Δf -value achieved, measured only up to the smallest number of overall function evaluations for any unsuccessful trial under consideration.

G3PCX outperforms Cauchy EDA on functions 1, 5, 6, 8, 9, 12, 16, 21, 22, 23, while Cauchy EDA beats G3PCX on functions 2, 7, 10, 13, 17, 18, i.e. in this small competition the G3PCX wins 10:6. (The results on the other functions are mixed, or neither algorithm solved the problem successfully.)

In Fig. 1, we can often see a peak at the beginning of the ERT ratio lines (functions 1, 5, 7, 11, 13, 14). The peak means that Cauchy EDA is much less efficient in the beginning of the search than the G3PCX algorithm. This is probably due to the fact that the probabilistic model used by Cauchy EDA needs some time to adapt to the fitness landscape, so that while G3PCX improves the best-so-far solution rather quickly right from the beginning, Cauchy EDA blunders. After finding the right model, the Cauchy EDA is sometimes able to close the gap and take the lead (e.g. for both ellipsoid functions 2 and 10 and for discus function 11 for dimensions lower than 20, or for the sharp ridge function 13).

G3PCX failed on functions 7 (Step-ellipsoid) and on both Schaffer’s functions 17 and 18. It seems that for these functions the global point of view represented by a unimodal Cauchy distribution is a better approach. Also for function 13 (sharp ridge problem), the global probabilistic model seems better, at least for tight target levels.

Interesting results may be found for function 14. G3PCX algorithm is orders of magnitude faster than Cauchy EDA for a broad range of target levels. But for target levels at about 10^{-5} and tighter, Cauchy EDA takes over and its results are much better. It seems that G3PCX is not even able to find some of the tighter target levels. One explanation for that might be that the stopping criterion for G3PCX is not set properly and actually prevents the algorithm from finding these target levels. Another explanation may be that for these tight target levels the population of G3PCX often converges to certain subspace of the search space and the algorithm stagnates.

Another note can be made on the variance enlargement constant used by Cauchy EDA. It was set to be approximately optimal for the Rosenbrock’s function. However,

Table 2: The average time demands per function evaluation (in microseconds) of the two compared algorithms.

Dim	2	3	5	10	20	40
G3PCX	470	443	420	411	540	750
CauchyEDA	51	17	9	9	11	NA

such setting may be too large for other functions. The ERT ratio for sphere function shows that with increasing problem dimensionality the gap between the algorithms gets larger. Also the results for function 21 and 22 (and possibly for functions 16 and 23) suggest, that the slow convergence of Cauchy EDA prevents it to be restarted more often which is the key to solve these problems; on the contrary, G3PCX converges probably much faster and is thus restarted more often which gives it a chance to have higher success rate.

Looking at Fig. 3, it can be stated that G3PCX beats CauchyEDA mainly on the separable functions (where for 20D, we can expect the G3PCX to be faster than Cauchy EDA at least 80 % of time regardless of the target level), on moderate functions (where G3PCX would be winner about 75 % of time), and on weak-structure functions (where Cauchy EDA almost does not work at all). On the other hand, Cauchy EDA has higher success rates on ill-conditioned and multi-modal functions, but compared to G3PCX and other algorithms it is orders of magnitude slower.

4. CPU TIMING EXPERIMENTS

The time requirements of both algorithms are taken from the respective articles, [8] and [7]. The multistart algorithm was run with the maximal number of evaluations set to 10^5 , the basic algorithm was restarted for at least 30 seconds. The experiment was conducted on Intel Core 2 CPU, T5600, 1.83 GHz, 1 GB RAM with Windows XP SP3 in MATLAB R2007b. The comparison of the average time demands per function evaluation are shown in Table 2.

The differences in the average time needed for function evaluation are probably caused by the different offspring population sizes. While G3PCX generates 2 offspring each generation, Cauchy EDA generates tens or hundreds. This means that the evaluation routine is called less often in Cauchy EDA than in G3PCX and can take advantage of the MATLAB matrix processing capabilities to a larger extent.

5. CONCLUSIONS

Two restarted local-search algorithms, G3PCX and Cauchy EDA, were compared using the BBOB 2010 methodology. Neither of the algorithms is successful in solving all the benchmark functions. The results are mixed, but the G3PCX gives better results for a larger subset of the benchmark functions. This is probably caused by the fact that it converges faster and can be restarted more often, when needed.

Acknowledgements

The author is supported by the Grant Agency of the Czech Republic with the grant no. 102/08/P094 entitled “Machine learning methods for solution construction in evolutionary algorithms”.

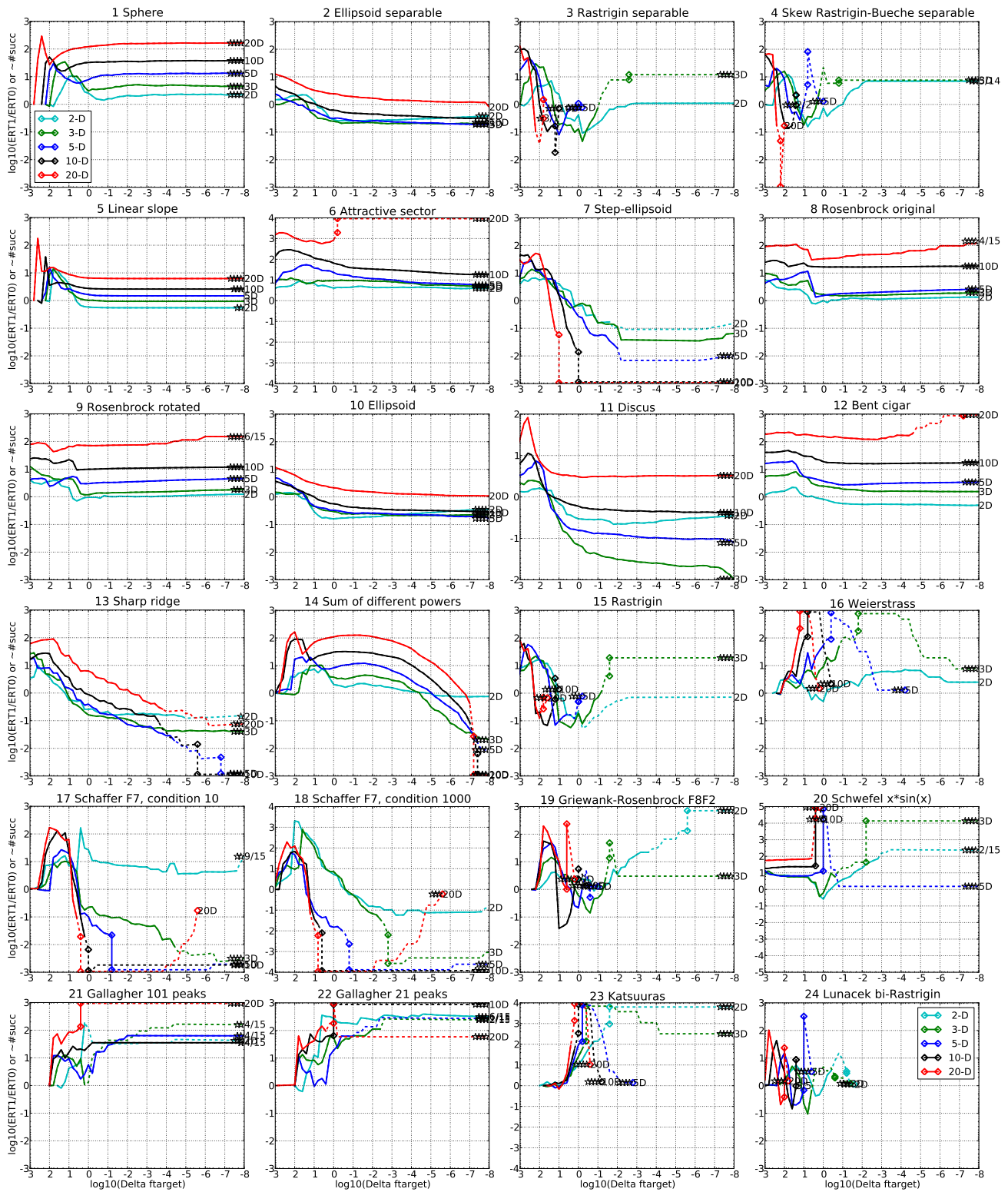


Figure 1: ERT ratio of CauchyEDA divided by G3PCX versus $\log_{10}(\Delta f)$ for f_1 – f_{24} in 2, 3, 5, 10, 20, 40-D. Ratios $< 10^0$ indicate an advantage of CauchyEDA, smaller values are always better. The line gets dashed when for any algorithm the ERT exceeds thrice the median of the trial-wise overall number of f -evaluations for the same algorithm on this function. Symbols indicate the best achieved Δf -value of one algorithm (ERT gets undefined to the right). The dashed line continues as the fraction of successful trials of the other algorithm, where 0 means 0% and the y-axis limits mean 100%, values below zero for CauchyEDA. The line ends when no algorithm reaches Δf anymore. The number of successful trials is given, only if it was in $\{1 \dots 9\}$ for CauchyEDA (1st number) and non-zero for G3PCX (2nd number). Results are significant with $p = 0.05$ for one star and $p = 10^{-\#\ast}$ otherwise, with Bonferroni correction within each figure.

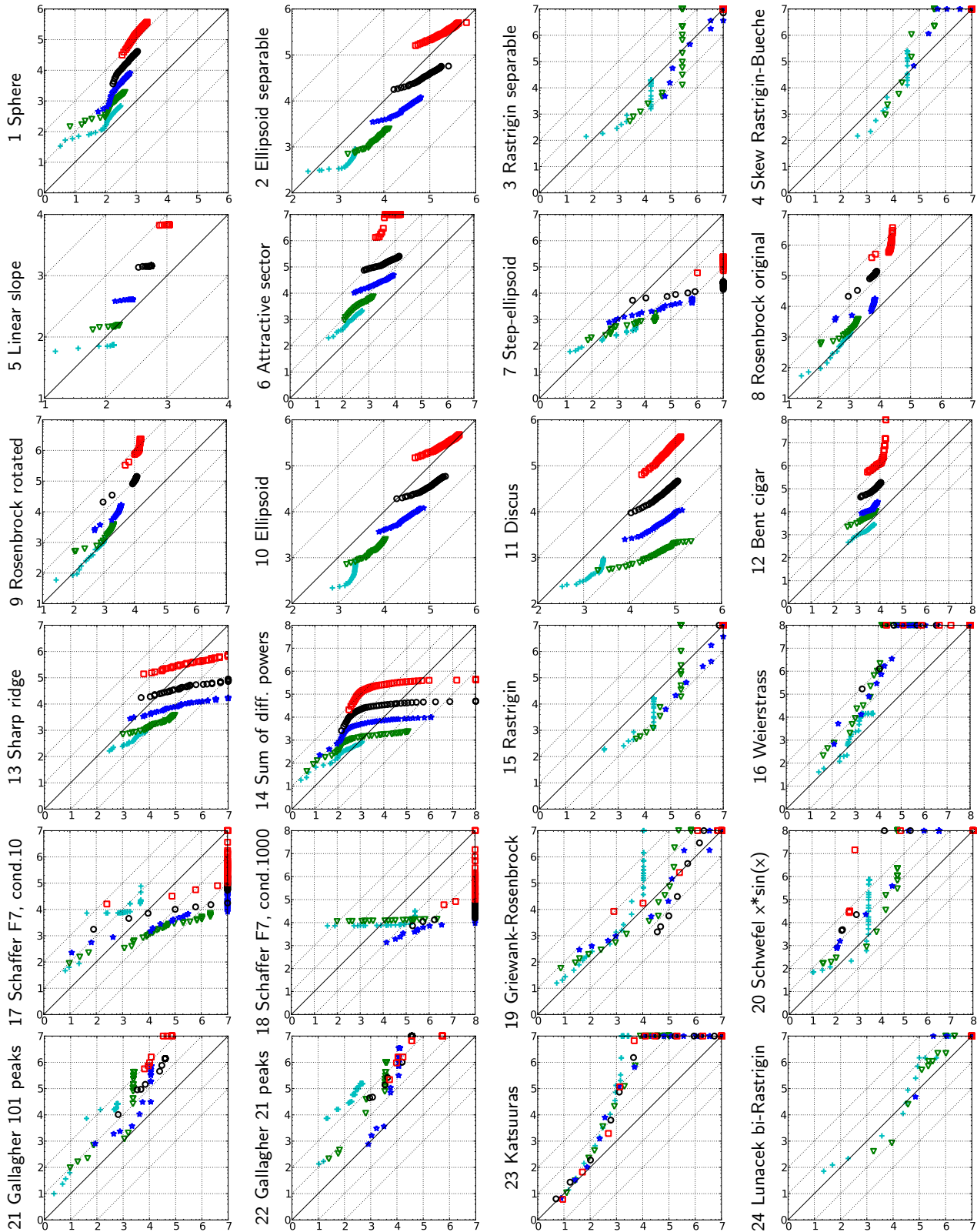


Figure 2: Expected running time (ERT in log10 of number of function evaluations) of CauchyEDA versus G3PCX for 46 target values $\Delta f \in [10^{-8}, 10]$ in each dimension for functions f_1 – f_{24} . Markers on the upper or right edge indicate that the target value was never reached by CauchyEDA or G3PCX respectively. Markers represent dimension: 2: +, 3: ∇ , 5: *, 10: \circ , 20: \square , 40: \diamond .

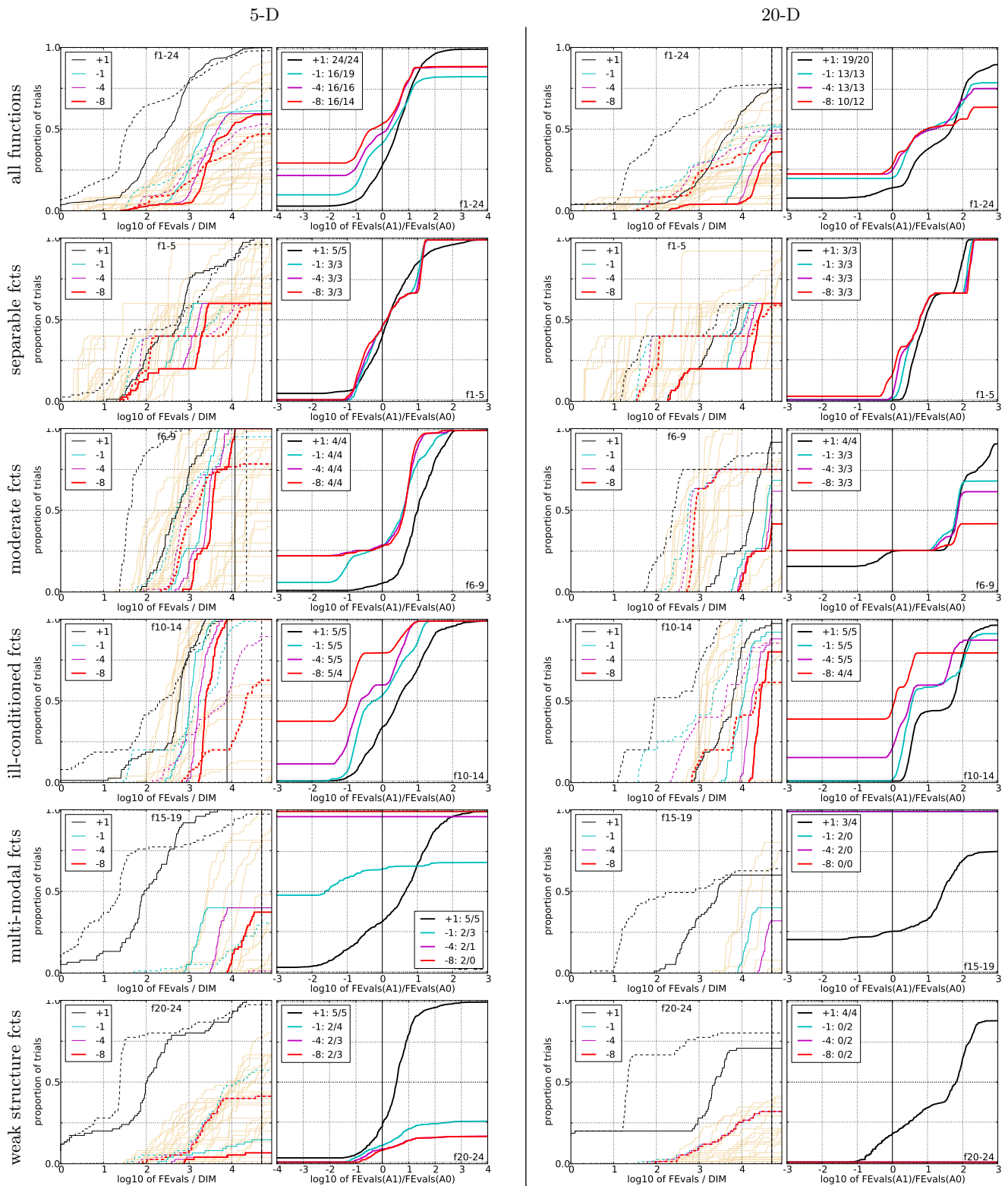


Figure 3: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/ D) to reach a target value $f_{\text{opt}} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for CauchyEDA (solid) and G3PCX (dashed). Light beige lines show the ECDF of FEvals for target value $\Delta f = 10^{-8}$ of algorithms benchmarked during BBOB-2009. Right sub-columns: ECDF of FEval ratios of CauchyEDA divided by G3PCX, all trial pairs for each function. Pairs where both trials failed are disregarded, pairs where one trial failed are visible in the limits being > 0 or < 1 . The legends indicate the number of functions that were solved in at least one trial (CauchyEDA first).

6. REFERENCES

- [1] K. Deb. A population-based algorithm-generator for real-parameter optimization. *Soft Computing*, 9(4):236–253, April 2005.
- [2] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. Technical report, Indian Institute of Technology, April 2002.
- [3] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.
- [4] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.
- [5] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. Updated February 2010.
- [6] P. Pošík. Preventing premature convergence in a simple EDA via global step size setting. In G. Rudolph, editor, *Parallel Problem Solving from Nature – PPSN X*, volume 5199 of *Lecture Notes in Computer Science*, pages 549–558. Springer, 2008.
- [7] P. Pošík. BBOB-benchmarking a simple estimation-of-distribution algorithm with Cauchy distribution. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 2309–2314, New York, NY, USA, 2009. ACM.
- [8] P. Pošík. BBOB-benchmarking the generalized generation gap model with parent centric crossover. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 2321–2328, New York, NY, USA, 2009. ACM.
- [9] K. Price. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*, pages 153–157, 1997.