Black-Box Optimization Benchmarking: Results for the $\mathrm{BayEDA_{cG}}$ Algorithm on the Noisy Function Testbed

Marcus Gallagher
School of Information Technology and Electrical Engineering
The University of Queensland
Brisbane, 4072. Australia
marcusg@itee.uq.edu.au

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: OptimizationGlobal Optimization, Unconstrained Optimization; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

General Terms

Algorithms

Keywords

Benchmarking, Black-box optimization, Evolutionary computation

1. INTRODUCTION

This paper presents experimental results for the $\mathrm{BayEDA_{cG}}$ continuous optimization algorithm on the BBOB noisy benchmark problem suite as part of the GECCO'09 Workshop.

2. ALGORITHM PRESENTATION

$\mathrm{BayEDA_{cG}}$ is an Estimation of Distribution Algorithm that uses Bayesian Inference to learn a posterior distribution over model parameters for the probability density estimation model used [2]. In this algorithm, the distribution is a product of univariate Gaussian distributions and inference is performed over mean and variance parameters for each dimension in the search space. The algorithm is described for a one-dimensional problem in Table 1. Given the factorized probability model, the extension to the multidimensional case is straightforward.

The following description of the algorithm is from [2]:

For a univariate Gaussian (Normal) model distribution, Bayesian inference can readily be carried out: the resulting expressions given here are drawn from Gelman et al. [3]. We consider the simplest case of a noninformative (flat) prior for the model parameters, expressing no preference for any particular values for the model parameters before observing any data. In this case, inference depends only on the data (selected individuals). The standard noninformative prior is uniform on $(\mu, \log\sigma^2)$ or

$$p(\mu, \sigma^2) \propto (\sigma^2)^{-1}$$

The joint posterior can be factorised as

$$p(\mu, \sigma^2|D) = p(\mu|\sigma^2, D)p(\sigma^2|D)$$

In this case, the marginal density for σ is

$$\sigma^2|D \sim \mathrm{Inv}-\chi^2(M_{sel}-1, s^2) \quad (1)$$

where s^2 is the sample variance of the data. The conditional density for μ is

$$\mu|D, \sigma^2 \sim N(\overline{x}, \sigma^2/M_{sel}) \quad (2)$$

where $\overline{x}$ is the sample mean of the data D.

The predictive distribution for $\tilde{x}$ given the data, μ and σ is

$$\tilde{x}|D, \mu, \sigma^2 \sim N(\mu, \sigma^2) \quad (3)$$

In the $\mathrm{BayEDA_{cG}}$ algorithm, sampling from the posterior predictive distribution $p(\tilde{x}|D)$ can be easily carried out in a three-step process. Firstly, a sample $\tilde{\sigma}^2$ is drawn from (1), then this sample is used to draw a sample $\tilde{\mu}$ from (2) and finally both samples are used to draw a sample $\tilde{x}$ from (3). The process is repeated M times to produce the population for use in the next generation.

Table 1: Algorithm: $\mathrm{BayEDA_{cG}}$.

Given: population size M, selection parameter τ
BEGIN (set $t=0$) Generate M individ. uniformly in S
REPEAT for $t = 1, 2, \ldots$ until stopping criterion is met
  Select $M_{sel} = \mathrm{Round}(M\cdot\tau)$ individ. via truncation
  Calculate sample mean $\overline{x}$ and variance s^2 of D
  Sample M individuals from $p_t(\mathbf{x}|D, \theta)$:
  FOR i=1:M
    Draw sample variance $\tilde{\sigma}^2 \sim \mathrm{Inv}-\chi^2(M_{sel}-1, s^2)$
    Draw sample mean $\tilde{\mu} \sim N(\overline{x}, \tilde{\sigma}^2/(M_{sel}))$
    Draw new individual $\mathbf{x}_i \sim N(\tilde{\mu}, \tilde{\sigma}^2)$
  ENDFOR
ENDREPEAT
END

The algorithm is summarized in Table 1. Note that for implementation purposes, a random draw y from an inverse-χ^2 distribution can be obtained by firstly drawing a sample z from the χ^2 distribution and applying $y = s^2/z$. The χ^2

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.

distribution is also a special case of the gamma distribution (see [3] for details).

3. EXPERIMENTAL PROCEDURE

The $\text{BayEDA}_{\text{cG}}$ algorithm was run on the current set of BBOB noisy benchmark functions (see other document for results on noisy functions). No parameter tuning was attempted with respect to the functions. The population size (M) was set to 10 times the dimensionality of the problem. The selection threshold τ (for truncation selection) was set (rather arbitrarily) to 0.8. The total number of function evaluations was set to 2000 times the problem dimensionality (making the total number of generations for each run of the algorithm equal to 200).

The *crafting effort* for this set of experiments is zero in this case.

4. RESULTS

Results from experiments according to [4] on the benchmarks functions given in [1, 5] are presented in Figures 1 and 2 and in Tables 2 and 3.

5. CPU TIMING EXPERIMENT

For the timing experiment the $\text{BayEDA}_{\text{cG}}$ algorithm was run as suggested on f_8 until at least 30 seconds has passed. The experiments in this paper were conducted on an Intel Pentium 4 quad core 2.4Ghz processor running Linux 2.6.24-23 SMP and Matlab R2008a. The results were (in seconds per function evaluation) 2.9;3.2;3.8;5.3 and 8.3 $\times 10^{-4}$ for dimension 2;3;5;10 and 20D and 1.4 $\times 10^{-3}$ for 40D.

6. CONCLUSION

The results show a wide variety of performance across the different test functions. Some functions seem reasonably well-solved for a range of precision and dimensionality values while others show only modest performance. This is for a reasonably small number of function evaluations. Neverthess, it will be intersting to see comparative analysis at the workshop.

7. REFERENCES

[1] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noisy functions. Technical Report 2009/21, Research Center PPE, 2009.

[2] M. Gallagher, I. Wood, J. Keith, and G. Sofronov. Posterior predictive optimization with thresholding: Bayesian inference in EDAs. In *Proc. Congress on Evolutionary Computation (CEC'07)*, pages 127–133, 2007.

[3] A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, second edition, 2004.

[4] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.

[5] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noisy functions definitions. Technical Report RR-6869, INRIA, 2009.

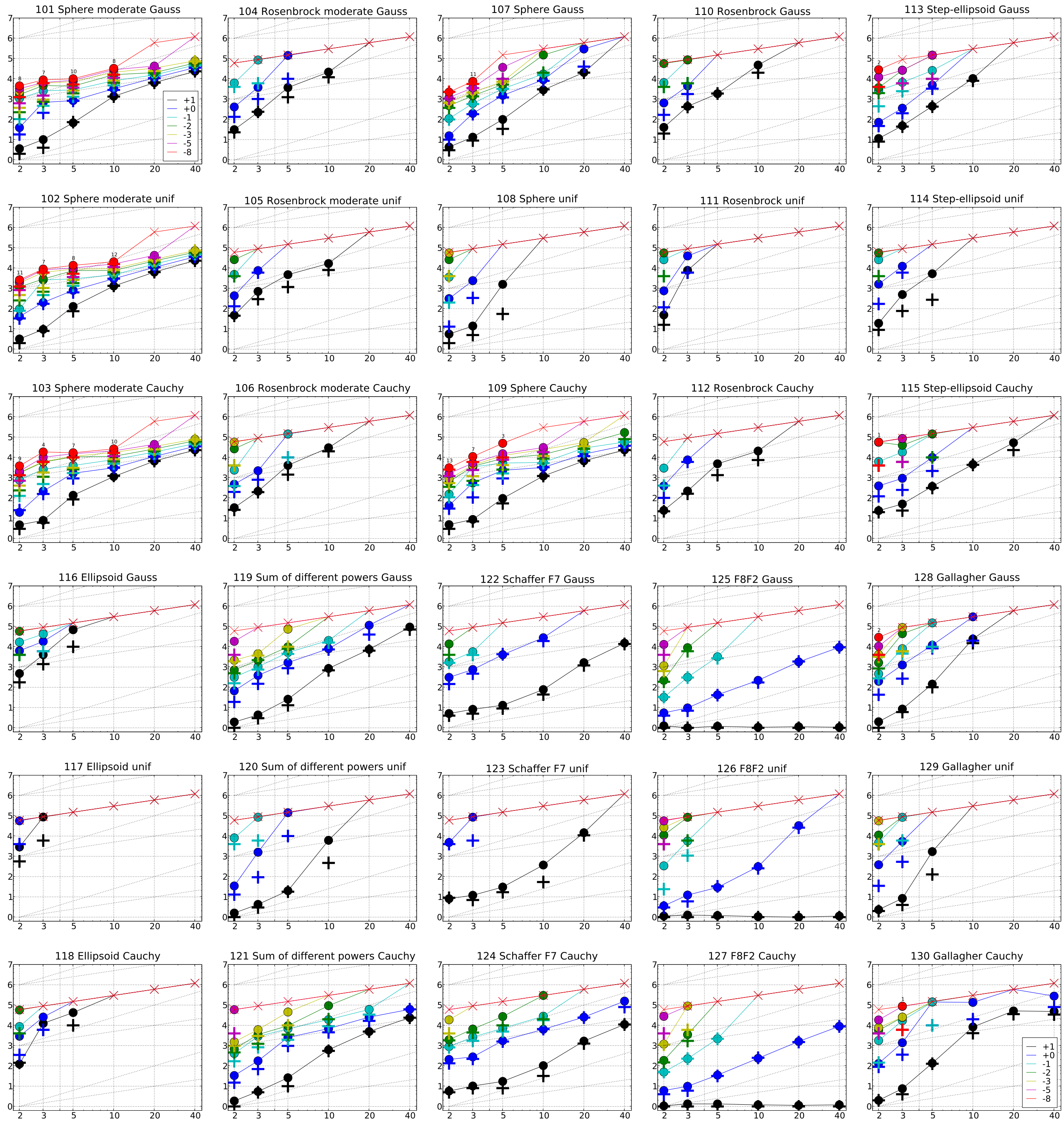

Figure 1: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_{101} and f_{130}) versus dimension in log-log presentation. The $\mathrm{ERT}(\Delta f)$ equals to $\#\mathrm{FEs}(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\mathrm{opt}} + \Delta f$ was surpassed during the trial. The $\#\mathrm{FEs}(\Delta f)$ are the total number of function evaluations while $f_{\mathrm{opt}} + \Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and f_{opt} denotes the optimal function value. Crosses (×) indicate the total number of function evaluations $\#\mathrm{FEs}(-\infty)$. Numbers above ERT-symbols indicate the number of successful trials. Annotated numbers on the ordinate are decimal logarithms. Additional grid lines show linear and quadratic scaling.

Δf	**f_{101} in 5-D**, N=15, mFE=10000					**f_{101} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	6.8e1	5.6e1	8.0e1	6.8e1	15	6.4e3	6.2e3	6.7e3	6.4e3
1	15	8.4e2	7.4e2	9.4e2	8.4e2	15	1.1e4	1.1e4	1.1e4	1.1e4
1e−1	13	2.8e3	1.9e3	4.1e3	2.0e3	15	1.5e4	1.5e4	1.6e4	1.5e4
1e−3	10	7.3e3	5.7e3	8.9e3	3.8e3	12	3.4e4	3.1e4	3.7e4	2.6e4
1e−5	10	8.5e3	7.1e3	1.0e4	4.7e3	12	4.3e4	4.1e4	4.4e4	3.4e4
1e−8	10	1.0e4	9.0e3	1.1e4	6.0e3	0	*21e−8*	*10e−8*	*49e−4*	3.5e4

Δf	**f_{102} in 5-D**, N=15, mFE=10000					**f_{102} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.3e2	8.1e1	1.8e2	1.3e2	15	6.3e3	6.1e3	6.5e3	6.3e3
1	15	8.0e2	6.5e2	9.6e2	8.0e2	15	1.1e4	1.1e4	1.1e4	1.1e4
1e−1	13	2.9e3	1.8e3	4.2e3	2.1e3	15	1.6e4	1.5e4	1.6e4	1.6e4
1e−3	9	8.9e3	7.0e3	1.1e4	4.9e3	14	2.7e4	2.6e4	2.9e4	2.5e4
1e−5	9	1.0e4	8.4e3	1.2e4	5.7e3	12	4.3e4	4.1e4	4.4e4	3.4e4
1e−8	8	1.4e4	1.2e4	1.5e4	6.9e3	0	*26e−8*	*13e−8*	*10e−5*	3.5e4

Δf	**f_{103} in 5-D**, N=15, mFE=10000					**f_{103} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.4e2	9.2e1	1.9e2	1.4e2	15	6.6e3	6.3e3	6.8e3	6.6e3
1	14	1.7e3	9.5e2	2.4e3	1.5e3	15	1.1e4	1.1e4	1.1e4	1.1e4
1e−1	12	4.0e3	2.6e3	5.4e3	2.9e3	14	1.8e4	1.6e4	2.1e4	1.7e4
1e−3	8	1.1e4	9.0e3	1.3e4	6.3e3	12	3.5e4	3.2e4	3.7e4	2.7e4
1e−5	7	1.5e4	1.3e4	1.8e4	7.5e3	12	4.4e4	4.3e4	4.5e4	3.5e4
1e−8	7	1.7e4	1.5e4	1.8e4	8.2e3	0	*95e−8*	*35e−8*	*23e−3*	4.0e4

Δf	**f_{104} in 5-D**, N=15, mFE=10000					**f_{104} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	12	3.7e3	2.2e3	5.2e3	3.4e3	0	*26e+0*	*18e+0*	*93e+0*	3.5e4
1	1	1.4e5	1.3e5	1.5e5	1.0e4	.	.	.	.	.
1e−1	0	*33e−1*	*24e−1*	*23e+0*	8.9e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{105} in 5-D**, N=15, mFE=10000					**f_{105} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	11	4.7e3	3.1e3	6.4e3	4.3e3	0	*33e+0*	*18e+0*	*11e+1*	3.5e4
1	0	*37e−1*	*31e−1*	*30e+0*	8.9e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{106} in 5-D**, N=15, mFE=10000					**f_{106} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	12	4.1e3	2.6e3	5.5e3	2.8e3	0	*51e+0*	*18e+0*	*12e+1*	4.0e4
1	1	1.4e5	1.3e5	1.5e5	1.0e4	.	.	.	.	.
1e−1	1	1.4e5	1.3e5	1.5e5	1.0e4	.	.	.	.	.
1e−3	0	*34e−1*	*27e−1*	*13e+0*	3.2e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{107} in 5-D**, N=15, mFE=10000					**f_{107} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	9.9e1	5.5e1	1.5e2	9.9e1	15	2.1e4	1.9e4	2.2e4	2.1e4
1	15	1.5e3	1.3e3	1.8e3	1.5e3	2	3.0e5	2.9e5	3.0e5	4.0e4
1e−1	15	3.2e3	2.9e3	3.5e3	3.2e3	0	*15e−1*	*86e−2*	*30e−1*	3.5e4
1e−3	15	7.2e3	6.8e3	7.7e3	7.2e3	.	.	.	.	.
1e−5	4	3.7e4	3.6e4	3.7e4	9.7e3	.	.	.	.	.
1e−8	0	*29e−6*	*17e−7*	*38e−5*	8.9e3	.	.	.	.	.

Δf	**f_{108} in 5-D**, N=15, mFE=10000					**f_{108} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	13	1.6e3	6.9e1	3.1e3	1.6e3	0	*72e+0*	*55e+0*	*87e+0*	3.2e3
1	0	*34e−1*	*16e−1*	*11e+0*	2.5e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{109} in 5-D**, N=15, mFE=10000					**f_{109} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	9.5e1	6.4e1	1.3e2	9.5e1	15	6.7e3	6.5e3	7.0e3	6.7e3
1	13	2.3e3	1.4e3	3.7e3	1.5e3	14	1.5e4	1.2e4	1.7e4	1.4e4
1e−1	11	5.3e3	3.6e3	6.9e3	3.2e3	12	2.9e4	2.5e4	3.3e4	2.4e4
1e−3	8	1.2e4	1.0e4	1.5e4	6.1e3	10	5.4e4	5.3e4	5.6e4	3.6e4
1e−5	8	1.5e4	1.4e4	1.6e4	7.8e3	0	*26e−5*	*87e−6*	*20e−2*	3.5e4
1e−8	3	5.0e4	5.0e4	5.0e4	1.0e4	.	.	.	.	.

Δf	**f_{110} in 5-D**, N=15, mFE=10000					**f_{110} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.8e3	1.6e3	2.0e3	1.8e3	0	*70e+0*	*42e+0*	*16e+1*	3.5e4
1	0	*30e−1*	*26e−1*	*33e−1*	6.3e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{111} in 5-D**, N=15, mFE=10000					**f_{111} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	0	*67e+0*	*15e+0*	*10e+1*	5.0e3	0	*22e+2*	*11e+2*	*60e+2*	3.5e4
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{112} in 5-D**, N=15, mFE=10000					**f_{112} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	11	4.9e3	3.2e3	6.5e3	3.6e3	0	*48e+0*	*21e+0*	*12e+1*	3.5e4
1	0	*35e−1*	*29e−1*	*20e+0*	2.5e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{113} in 5-D**, N=15, mFE=10000					**f_{113} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	4.5e2	3.6e2	5.4e2	4.5e2	0	*19e+0*	*12e+0*	*34e+0*	3.5e4
1	13	4.9e3	3.9e3	6.0e3	3.7e3	.	.	.	.	.
1e−1	5	2.6e4	2.3e4	2.8e4	8.9e3	.	.	.	.	.
1e−3	1	1.5e5	1.4e5	1.5e5	1.0e4	.	.	.	.	.
1e−5	1	1.5e5	1.4e5	1.5e5	1.0e4	.	.	.	.	.
1e−8	0	*20e−2*	*42e−3*	*13e−1*	5.0e3	.	.	.	.	.

Δf	**f_{114} in 5-D**, N=15, mFE=10000					**f_{114} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	10	5.3e3	3.3e3	8.1e3	4.1e3	0	*32e+1*	*22e+1*	*43e+1*	2.5e3
1	0	*73e−1*	*46e−1*	*27e+0*	2.2e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{115} in 5-D**, N=15, mFE=10000					**f_{115} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	3.7e2	2.8e2	4.6e2	3.7e2	8	5.3e4	4.6e4	6.0e4	2.8e4
1	8	1.0e4	6.9e3	1.2e4	4.5e3	0	*90e−1*	*55e−1*	*19e+0*	2.5e4
1e−1	1	1.4e5	1.3e5	1.5e5	1.0e4	.	.	.	.	.
1e−3	0	*89e−2*	*15e−2*	*23e−1*	2.2e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{116} in 5-D**, N=15, mFE=10000					**f_{116} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	2	6.9e4	6.3e4	7.5e4	6.2e3	0	*24e+2*	*14e+2*	*56e+2*	3.5e4
1	0	*37e+0*	*65e−1*	*30e+1*	4.5e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{117} in 5-D**, N=15, mFE=10000					**f_{117} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	0	*21e+1*	*15e+1*	*57e+1*	4.5e2	0	*13e+3*	*11e+3*	*18e+3*	2.2e4
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{118} in 5-D**, N=15, mFE=10000					**f_{118} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	3	4.3e4	3.7e4	4.7e4	7.7e3	0	*10e+2*	*47e+1*	*18e+2*	3.5e4
1	0	*90e+0*	*63e−1*	*31e+1*	2.0e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{119} in 5-D**, N=15, mFE=10000					**f_{119} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.6e1	1.7e1	3.5e1	2.6e1	15	7.4e3	6.2e3	8.7e3	7.4e3
1	15	1.6e3	1.0e3	2.4e3	1.6e3	5	1.1e5	1.1e5	1.2e5	3.7e4
1e−1	13	5.2e3	4.3e3	6.2e3	4.3e3	0	*15e−1*	*72e−2*	*52e−1*	3.5e4
1e−3	2	7.4e4	7.3e4	7.5e4	9.2e3	.	.	.	.	.
1e−5	0	*68e−4*	*92e−5*	*11e−2*	7.9e3	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{120} in 5-D**, N=15, mFE=10000					**f_{120} in 20-D**, N=15, mFE=40000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	2.0e1	1.4e1	2.6e1	2.0e1	0	*20e+0*	*13e+0*	*27e+0*	4.0e3
1	1	1.4e5	1.3e5	1.5e5	1.8e3	.	.	.	.	.
1e−1	0	*17e−1*	*11e−1*	*30e−1*	3.5e2	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 2: Shown are, for functions f_{101}-f_{120} and for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\rm opt} + \Delta f$ (ERT, see Figure 1); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value (RT_{succ}). If $f_{\rm opt} + \Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 1 for the names of functions.

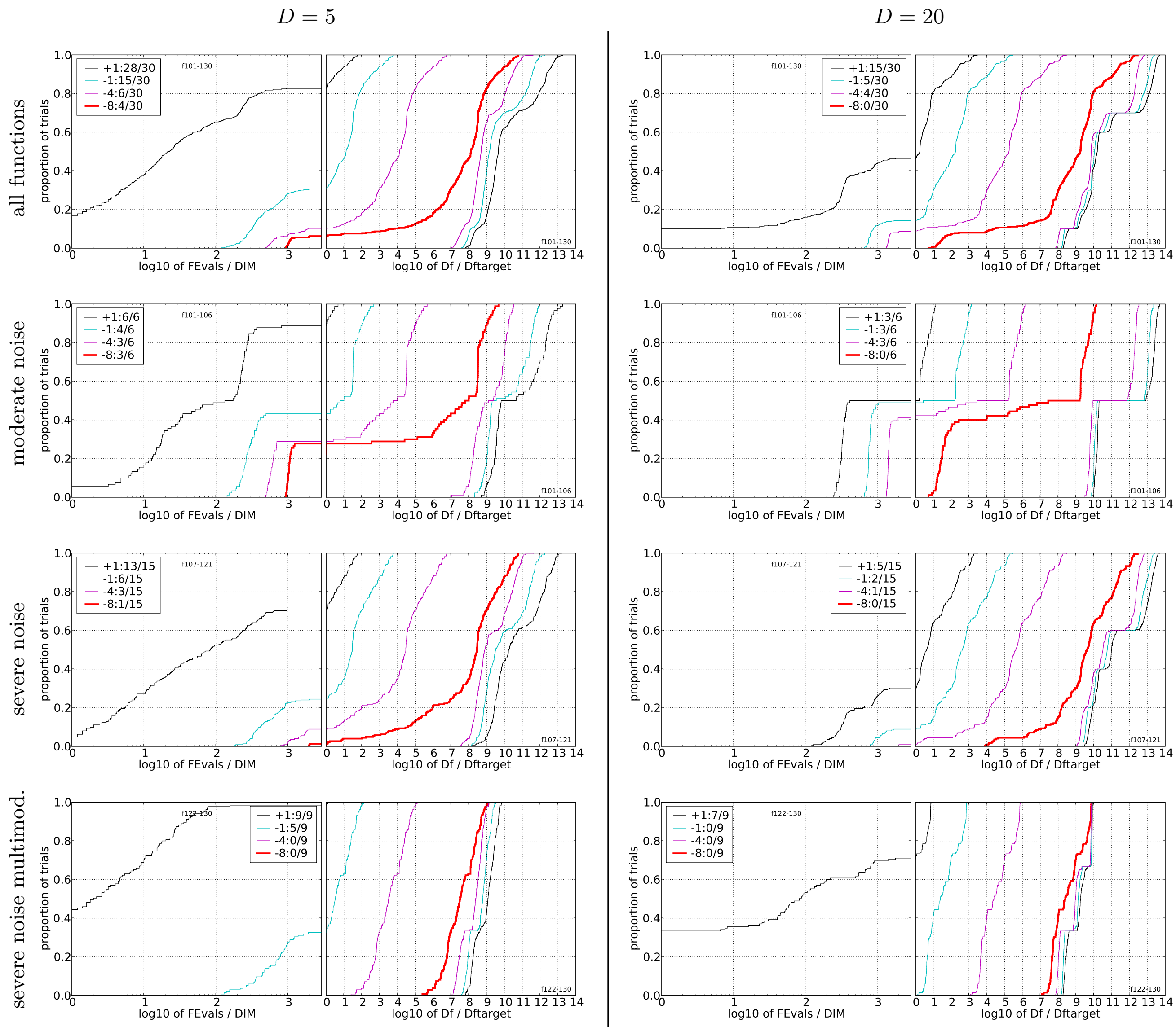

Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. Right subplots: ECDF of the best achieved Δf divided by 10^k (upper left lines in continuation of the left subplot), and best achieved Δf divided by 10^{-8} for running times of $D, 10\,D, 100\,D \ldots$ function evaluations (from right to left cycling black-cyan-magenta). Top row: all results from all functions; second row: moderate noise functions; third row: severe noise functions; fourth row: severe noise and highly-multimodal functions. The legends indicate the number of functions that were solved in at least one trial. FEvals denotes number of function evaluations, D and DIM denote search space dimension, and Δf and Df denote the difference to the optimal function value.

	f_{121} in 5-D, N=15, mFE=10000					**f_{121} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.6e1	1.3e1	4.0e1	2.6e1	15	5.0e3	4.5e3	5.4e3	5.0e3
1	13	2.5e3	1.5e3	3.9e3	1.7e3	13	2.5e4	2.1e4	3.0e4	2.1e4
1e−1	10	6.7e3	4.8e3	8.5e3	5.0e3	8	6.1e4	5.6e4	6.5e4	3.3e4
1e−3	3	4.6e4	4.3e4	4.9e4	8.5e3	0	*58e−3*	*18e−3*	*13e−1*	3.5e4
1e−5	0	*41e−4*	*69e−5*	*12e−1*	6.3e3	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{122} in 5-D, N=15, mFE=10000					**f_{122} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e1	8.7e0	1.8e1	1.3e1	15	1.6e3	1.3e3	2.0e3	1.6e3
1	15	4.0e3	3.6e3	4.5e3	4.0e3	0	*34e−1*	*20e−1*	*46e−1*	3.5e4
1e−1	0	*30e−2*	*14e−2*	*54e−2*	8.9e3	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{123} in 5-D, N=15, mFE=10000					**f_{123} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.0e1	2.0e1	4.3e1	3.0e1	13	1.5e4	9.4e3	2.0e4	1.3e4
1	0	*25e−1*	*16e−1*	*50e−1*	6.3e2	0	*83e−1*	*71e−1*	*10e+0*	1.1e4
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{124} in 5-D, N=15, mFE=10000					**f_{124} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.7e1	1.1e1	2.5e1	1.7e1	15	1.7e3	1.3e3	2.0e3	1.7e3
1	15	1.8e3	1.5e3	2.0e3	1.8e3	15	2.5e4	2.4e4	2.7e4	2.5e4
1e−1	12	7.2e3	6.2e3	8.1e3	5.7e3	0	*37e−2*	*15e−2*	*78e−2*	3.5e4
1e−3	0	*13e−3*	*45e−4*	*21e−2*	8.9e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{125} in 5-D, N=15, mFE=10000					**f_{125} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.2e0	1.0e0	1.4e0	1.2e0	15	1.1e0	1.0e0	1.3e0	1.1e0
1	15	4.2e1	3.2e1	5.3e1	4.2e1	15	1.8e3	1.7e3	1.9e3	1.8e3
1e−1	15	3.2e3	2.6e3	3.9e3	3.2e3	0	*50e−2*	*43e−2*	*53e−2*	2.0e4
1e−3	0	*65e−3*	*36e−3*	*83e−3*	6.3e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{126} in 5-D, N=15, mFE=10000					**f_{126} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.2e0	1.0e0	1.4e0	1.2e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	15	2.9e1	2.4e1	3.5e1	2.9e1	11	3.3e4	2.6e4	3.9e4	2.3e4
1e−1	0	*27e−2*	*13e−2*	*43e−2*	4.0e2	0	*94e−2*	*81e−2*	*11e−1*	2.0e4
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{127} in 5-D, N=15, mFE=10000					**f_{127} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e0	1.1e0	1.6e0	1.3e0	15	1.1e0	1.1e0	1.3e0	1.1e0
1	15	3.5e1	2.6e1	4.4e1	3.5e1	15	1.6e3	1.4e3	1.8e3	1.6e3
1e−1	15	2.2e3	1.8e3	2.6e3	2.2e3	0	*42e−2*	*36e−2*	*45e−2*	2.2e4
1e−3	0	*41e−3*	*20e−3*	*75e−3*	5.0e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{128} in 5-D, N=15, mFE=10000					**f_{128} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.4e2	1.1e2	1.8e2	1.4e2	0	*45e+0*	*36e+0*	*58e+0*	1.8e4
1	9	1.2e4	1.0e4	1.4e4	7.4e3	.	.	.	.	.
1e−1	1	1.5e5	1.5e5	1.5e5	1.0e4	.	.	.	.	.
1e−3	0	*60e−2*	*11e−2*	*23e−1*	7.1e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{129} in 5-D, N=15, mFE=10000					**f_{129} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	13	1.7e3	2.6e2	3.2e3	1.6e3	0	*69e+0*	*66e+0*	*72e+0*	2.5e3
1	0	*62e−1*	*21e−1*	*10e+0*	3.2e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

	f_{130} in 5-D, N=15, mFE=10000					**f_{130} in 20-D**, N=15, mFE=40000				
Δf	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.4e2	1.1e2	1.6e2	1.4e2	8	5.1e4	4.3e4	6.0e4	2.4e4
1	1	1.4e5	1.3e5	1.5e5	1.0e4	0	*99e−1*	*21e−1*	*25e+0*	3.5e4
1e−1	1	1.4e5	1.3e5	1.5e5	1.0e4	.	.	.	.	.
1e−3	0	*19e−1*	*12e−1*	*50e−1*	5.6e3	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 3: Shown are, for functions f_{121}-f_{130} and for a given target difference to the optimal function value Δf: the number of successful trials (#); the expected running time to surpass $f_{\rm opt} + \Delta f$ (ERT, see Figure 1); the 10%-tile and 90%-tile of the bootstrap distribution of ERT; the average number of function evaluations in successful trials or, if none was successful, as last entry the median number of function evaluations to reach the best function value ($\mathbf{RT_{succ}}$). If $f_{\rm opt} + \Delta f$ was never reached, figures in *italics* denote the best achieved Δf-value of the median trial and the 10% and 90%-tile trial. Furthermore, N denotes the number of trials, and mFE denotes the maximum of number of function evaluations executed in one trial. See Figure 1 for the names of functions.