Comparing Natural Evolution Strategies to BIPOP-CMA-ES on Noiseless and Noisy Black-box Optimization Testbeds

Tom Schaul
Courant Institute of Mathematical Sciences, New York University
Broadway 715, New York, USA
schaul@cims.nyu.edu

ABSTRACT

Natural Evolution Strategies (NES) are a recent member of the class of real-valued optimization algorithms that are based on adapting search distributions. Exponential NES (xNES) are the most common instantiation of NES, and particularly appropriate for the BBOB 2012 benchmarks, given that many are non-separable, and their relatively small problem dimensions. Here, we augment xNES with adaptation sampling, which adapts learning rates online, and compare the resulting performance directly to the BIPOP-CMA-ES algorithm, the winner of the 2009 black-box optimization benchmarking competition (BBOB). This report provides an extensive empirical comparison, both on the noise-free and noisy BBOB testbeds.

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

Evolution Strategies, Natural Gradient, Benchmarking

1. INTRODUCTION

Evolution strategies (ES), in contrast to traditional evolutionary algorithms, aim at repeating the type of mutation that led to those good individuals. We can characterize those mutations by an explicitly parameterized *search distribution* from which new candidate samples are drawn, akin to estimation of distribution algorithms (EDA). Covariance matrix adaptation ES (CMA-ES [12]) innovated the field by introducing a parameterization that includes the full covariance matrix, allowing them to solve highly non-separable problems.

A more recent variant, *natural evolution strategies* (NES [22, 6, 20, 21]) aims at a higher level of generality, providing a procedure to update the search distribution's parameters for any type of distribution, by ascending the gradient towards higher expected fitness. Further, it has been shown [17, 15] that following the *natural gradient* to adapt the search distribution is highly beneficial, because it appropriately normalizes the update step with respect to its uncertainty and makes the algorithm scale-invariant.

Exponential NES (xNES), the most common instantiation of NES, used a search distribution parameterized by a mean vector and a full covariance matrix, and is thus most similar to CMA-ES (in fact, the precise relation is described in [4] and [5]). Given the relatively small problem dimensions of the BBOB benchmarks, and the fact that many are non-separable, it is also among the most appropriate NES variants for the task. Adaptation sampling is a technique for the online adaptation of its learning rate, which is designed to speed up convergence. This may be beneficial to algorithms like xNES, because the optimization traverses qualitatively different phases, during which different learning rates may be optimal.

In this report, we retain the original formulation of xNES (including all parameter settings, except for an added stopping criterion), but augmented with adaptation sampling. We compare this algorithm (xNES-as) to the winning entry of the 2009 BBOB competition, namely BIPOP-CMA-ES, described in detail in [7, 8]. We describe the comparative empirical performance on all 54 benchmark functions (both noise-free and noisy) of the BBOB 2012 workshop.

2. NATURAL EVOLUTION STRATEGIES

Natural evolution strategies (NES) maintain a search distribution π and adapt the distribution parameters θ by following the *natural gradient* [1] of expected fitness J, that is, maximizing

$$J(\theta) = \mathbb{E}_\theta[f(\mathbf{z})] = \int f(\mathbf{z})\ \pi(\mathbf{z} \,|\, \theta)\ d\mathbf{z}$$

Just like their close relative CMA-ES [12], NES algorithms are invariant under monotone transformations of the fitness function and linear transformations of the search space. Each iteration the algorithm produces n samples $\mathbf{z}_i \sim \pi(\mathbf{z}|\theta)$, $i \in \{1, \ldots, n\}$, i.i.d. from its search distribution, which is parameterized by θ. The gradient w.r.t. the parameters θ can be rewritten (see [22]) as

$$\nabla_\theta J(\theta) = \nabla_\theta \int f(\mathbf{z})\ \pi(\mathbf{z}\,|\,\theta)\ d\mathbf{z} = \mathbb{E}_\theta\left[f(\mathbf{z})\ \nabla_\theta \log \pi(\mathbf{z}\,|\,\theta)\right]$$

from which we obtain a Monte Carlo estimate

$$\nabla_\theta J(\theta) \approx \frac{1}{n}\sum_{i=1}^{n} f(\mathbf{z}_i)\ \nabla_\theta \log \pi(\mathbf{z}_i\,|\,\theta)$$

of the search gradient. The key step then consists in replacing this gradient by the natural gradient defined as $\mathbf{F}^{-1}\nabla_\theta J(\theta)$ where $\mathbf{F} = \mathbb{E}\left[\nabla_\theta \log \pi\left(\mathbf{z}|\theta\right) \nabla_\theta \log \pi\left(\mathbf{z}|\theta\right)^\top\right]$ is the Fisher information matrix. The search distribution is iteratively updated using natural gradient ascent

$$\theta \leftarrow \theta + \eta \mathbf{F}^{-1}\nabla_\theta J(\theta)$$

with learning rate parameter η.

2.1 Exponential NES

While the NES formulation is applicable to arbitrary parameterizable search distributions [22, 15], the most common variant employs multinormal search distributions. For that case, two helpful techniques were introduced in [6], namely an exponential parameterization of the covariance matrix, which guarantees positive-definiteness, and a novel method for changing the coordinate system into a "natural" one, which makes the algorithm computationally efficient. The resulting algorithm, NES with a multivariate Gaussian search distribution and using both these techniques is called *xNES*, and the pseudocode is given in Algorithm 1.

Algorithm 1: Exponential NES (xNES)

```
input: f, μ_init, η_σ, η_B, u_k

initialize  μ ← μ_init
            σ ← 1
            B ← I

repeat
    for k = 1 ... n do
        draw sample s_k ~ N(0, I)
        z_k ← μ + σ B^T s_k
        evaluate the fitness f(z_k)
    end

    sort {(s_k, z_k)} with respect to f(z_k)
    and assign utilities u_k to each sample

    compute gradients
      ∇_δ J  ←  Σ_{k=1}^n u_k · s_k
      ∇_M J  ←  Σ_{k=1}^n u_k · (s_k s_k^T − I)
      ∇_σ J  ←  tr(∇_M J)/d
      ∇_B J  ←  ∇_M J − ∇_σ J · I

    update parameters
      μ  ←  μ + σB · ∇_δ J
      σ  ←  σ · exp(η_σ/2 · ∇_σ J)
      B  ←  B · exp(η_B/2 · ∇_B J)
until stopping criterion is met
```

2.2 Adaptation Sampling

First introduced in [15] (chapter 2, section 4.4), *adaptation sampling* is a new meta-learning technique [19] that can adapt hyper-parameters online, in an economical way that is grounded on a measure statistical improvement.

Here, we apply it to the learning rate of the global step-size η_σ. The idea is to consider whether a larger learning-rate $\eta'_\sigma = \frac{3}{2}\eta_\sigma$ would have been more likely to generate the good samples in the current batch. For this we determine the (hypothetical) search distribution that would have resulted from such a larger update $\pi(\cdot|\theta')$. Then we compute importance weights

$$w'_k = \frac{\pi(\mathbf{z}_k|\theta')}{\pi(\mathbf{z}_k|\theta)}$$

for each of the n samples $\mathbf{z}_k$ in our current population, generated from the actual search distribution $\pi(\cdot|\theta)$. We then conduct a *weighted* Mann-Whitney test [15] (appendix A) to determine if the set $\{\text{rank}(\mathbf{z}_k)\}$ is inferior to its reweighted counterpart $\{w'_k \cdot \text{rank}(\mathbf{z}_k)\}$ (corresponding to the larger learning rate), with statistical significance ρ. If so, we increase the learning rate by a factor of $1 + c'$, up to at most $\eta_\sigma = 1$ (where $c' = 0.1$). Otherwise it decays to its initial value:

$$\eta_\sigma \leftarrow (1 - c') \cdot \eta_\sigma + c' \cdot \eta_{\sigma,\text{init}}$$

The procedure is summarized in algorithm 2 (for details and derivations, see [15]). The combination of xNES with adaptation sampling is dubbed *xNES-as*.

One interpretation of why adaptation sampling is helpful is that half-way into the search, (after a local attractor has been found, e.g., towards the end of the valley on the Rosenbrock benchmarks f_8 or f_9), the convergence speed can be boosted by an increased learning rate. For such situations, an online adaptation of hyper-parameters is inherently well-suited.

Algorithm 2: Adaptation sampling

```
input  : η_{σ,t}, η_{σ,init}, θ_t, θ_{t−1}, {(z_k, f(z_k))}, c', ρ
output: η_{σ,t+1}
compute hypothetical θ', given θ_{t−1} and using 3/2 η_{σ,t}
for k = 1 ... n do
    w'_k = π(z_k|θ') / π(z_k|θ)
end
S ← {rank(z_k)}
S' ← {w'_k · rank(z_k)}
if weighted-Mann-Whitney(S, S') < ρ then
    return (1 − c') · η_σ + c' · η_{σ,init}
else
    return min((1 + c') · η_σ, 1)
end
```

3. EXPERIMENTAL SETTINGS

We use identical default hyper-parameter values for all benchmarks (both noisy and noise-free functions), which are taken from [6, 15]. Table 1 summarizes all the hyper-parameters used.

In addition, we make use of the provided target fitness f_{opt} to trigger *independent* algorithm restarts[1], using a simple

[1]It turns out that this use of f_{opt} is technically not permitted by the BBOB guidelines, so strictly speaking a different restart strategy should be employed, for example the one described in [15].

Table 1: Default parameter values for xNES (including the utility function and adaptation sampling) as a function of problem dimension d.

parameter	default value
n	$4 + \lfloor 3\log(d) \rfloor$
$\eta_\sigma = \eta_\mathbf{B}$	$\frac{3(3+\log(d))}{5d\sqrt{d}}$
u_k	$\frac{\max\left(0, \log(\frac{n}{2}+1) - \log(k)\right)}{\sum_{j=1}^{n} \max\left(0, \log(\frac{n}{2}+1) - \log(j)\right)} - \frac{1}{n}$
ρ	$\frac{1}{2} - \frac{1}{3(d+1)}$
c'	$\frac{1}{10}$

ad-hoc procedure: If the log-progress during the past $1000d$ evaluations is too small, i.e., if

$$\log_{10}\left|\frac{f_{\rm opt} - f_t}{f_{\rm opt} - f_{t-1000d}}\right| < (r+2)^2 \cdot m^{3/2} \cdot [\log_{10}|f_{\rm opt} - f_t| + 8]$$

where m is the remaining budget of evaluations divided by $1000d$, f_t is the best fitness encountered until evaluation t and r is the number of restarts so far. The total budget is $10^5 d^{3/2}$ evaluations.

Implementations of this and other NES algorithm variants are available in Python through the PyBrain machine learning library [18], as well as in other languages at `www.idsia.ch/~tom/nes.html`.

4. RESULTS

Results from experiments according to [9] on the benchmark functions given in [2, 10, 3, 11] are presented in Figures 1, 2 and 3 and in Tables 2, 3 and 4. The **expected running time (ERT)**, used in the figures and table, depends on a given target function value, $f_{\rm t} = f_{\rm opt} + \Delta f$, 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 $f_{\rm t}$, summed over all trials and divided by the number of trials that actually reached $f_{\rm t}$ [9, 13]. **Statistical significance** is tested with the rank-sum test for a given target $\Delta f_{\rm t}$ (10^{-8} as in Figure 1) using, for each trial, either the number of needed function evaluations to reach $\Delta f_{\rm t}$ (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.

Some of the result plots (like performance scaling with dimension on the noisy benchmarks), as well as the CPU-timing results were omitted here but are available in a standalone benchmarking report [16].

5. DISCUSSION

Figure 2 gives a good overview picture, showing that across all benchmarks taken together, both BIPOP-CMA-ES and xNES-as performs better than most of the BBOB 2009 contestants.

According to Tables 2, 3 and 4, BIPOP-CMA-ES is consistently outperforming xNES-as (in dimensions 5 and 20) on functions 1, 5, 6, 15, 20, 23, 101, 102, 103, 105, 107, 108, 109, 114, 120, 122, 123 and 127, and it additionally does so in dimension 20 on functions 8, 9, 12, 16, 24 104, 113 and 116.

On the other hand, xNES-as is consistently outperforming BIPOP-CMA-ES (in dimensions 5 and 20) on functions 4, 10, 11, 18, 115 and 118, and additionally does so on dimension 20 on function 2.

In conclusion, we find that xNES-as is close in performance to BIPOP-CMA-ES, across a large fraction of the benchmark functions; but there is some diversity as well, with xNES-as being significantly better on 6 of the functions and significantly worse on 18 of them. Clearly, xNES-as underperforms on multi-modal functions, a weakness that could be addressed through larger population sizes, or better even, *adaptive* population sizes – possibly using a similar scheme than the one presented here for making learning rates adaptive.

Acknowlegements

The author wants to thank the organizers of the BBOB workshop for providing such a well-designed benchmark setup, and especially such high-quality post-processing utilities.

This work was funded in part through AFR postdoc grant number 2915104, of the National Research Fund Luxembourg.

6. REFERENCES

[1] S. I. Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10:251–276, 1998.

[2] 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.

[3] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Presentation of the noisy functions. Technical Report 2009/21, Research Center PPE, 2010.

[4] N. Fukushima, Y. Nagata, S. Kobayashi, and I. Ono. Proposal of distance-weighted exponential natural evolution strategies. In *2011 IEEE Congress of Evolutionary Computation*, pages 164–171. IEEE, 2011.

[5] T. Glasmachers, T. Schaul, and J. Schmidhuber. A Natural Evolution Strategy for Multi-Objective Optimization. In *Parallel Problem Solving from Nature (PPSN)*, 2010.

[6] T. Glasmachers, T. Schaul, Y. Sun, D. Wierstra, and J. Schmidhuber. Exponential Natural Evolution Strategies. In *Genetic and Evolutionary Computation Conference (GECCO)*, Portland, OR, 2010.

[7] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In Rothlauf [14], pages 2389–2396.

[8] N. Hansen. Benchmarking a BI-population CMA-ES on the BBOB-2009 noisy testbed. In Rothlauf [14], pages 2397–2402.

[9] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2012: Experimental setup. Technical report, INRIA, 2012.

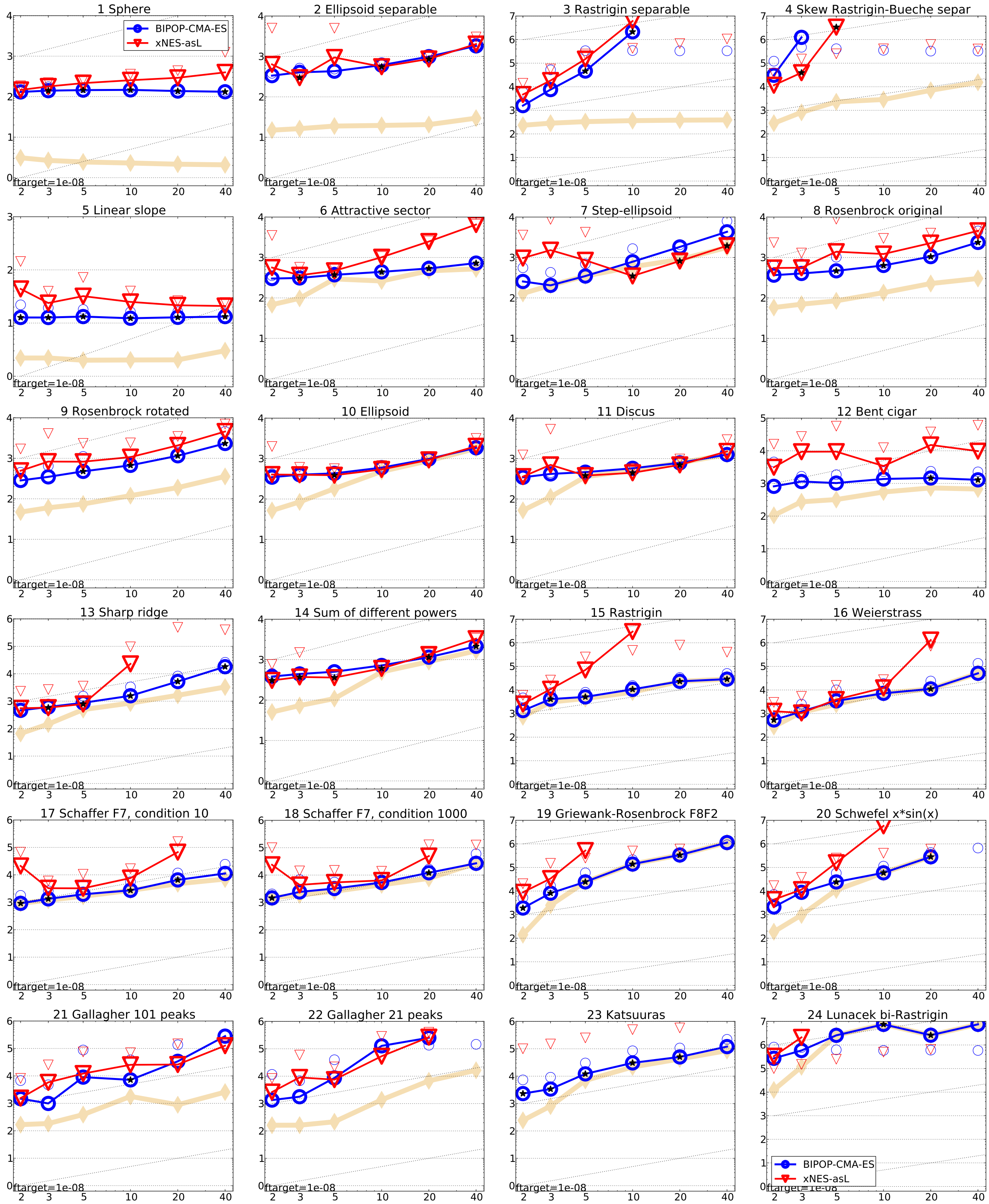

Figure 1: Expected running time (ERT in number of f-evaluations) divided by dimension for target function value 10^{-8} as $\log_{10}$ values versus dimension. Different symbols correspond to different algorithms given in the legend of f_1 and f_{24}. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Horizontal lines give linear scaling, slanted dotted lines give quadratic scaling. Black stars indicate statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: ○:BIPOP-CMA-ES, ▽:xNES-as.

5-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
f_1	11	12	12	12	12	15/15
1: CMA	3.2(2)	15(4)	**27**(5)$^{\star 2}$	**40**(4)$^{\star 3}$	**53**(6)$^{\star 2}$	15/15
2: NES	2.9(2)	16(5)	37(8)	60(12)	78(17)	15/15
f_2	83	88	90	92	94	15/15
1: CMA	13(4)	18(2)	20(2)	21(2)	22(2)	15/15
2: NES	11(5)	19(18)	39(62)	43(63)	49(92)	15/15
f_3	716	1637	1646	1650	1654	15/15
1: CMA	1.4(1)	**139**(107)*	**139**(107)*	**139**(107)*	**140**(107)*	14/15
2: NES	1.5(0.7)	454(357)	452(470)	451(383)	450(379)	13/15
f_4	809	1688	1817	1886	1903	15/15
1: CMA	2.7(3)	∞	∞	∞	∞ *1.8e6*	0/15
2: NES	3.8(5)	**9998**(10972)$^{\star 2}$	**9287**(10419)$^{\star 2}$	**8949**(9116)$^{\star 2}$	**8868**(10471)$^{\star 2}$	1/15
f_5	10	10	10	10	10	15/15
1: CMA	**4.5**(2)$^{\star 2}$	**6.6**(2)$^{\star 3}$	**6.6**(2)$^{\star 3}$	**6.6**(2)$^{\star 3}$	**6.6**(2)$^{\star 3}$	15/15
2: NES	10(4)	16(9)	16(8)	16(8)	16(8)	15/15
f_6	114	281	580	1038	1332	15/15
1: CMA	2.3(1)	2.2(0.6)	**1.7**(0.2)*	1.3(0.3)	**1.3**(0.2)$^{\star 2}$	15/15
2: NES	1.5(1)	2.4(0.6)	2.0(0.2)	1.5(0.2)	1.6(0.2)	15/15
f_7	24	1171	1572	1572	1597	15/15
1: CMA	5.0(5)	1(1)	1(0.9)	1(0.9)	1(0.9)	15/15
2: NES	4.4(3)	3.2(4)	2.6(3)	2.6(3)	2.6(3)	15/15
f_8	73	336	391	410	422	15/15
1: CMA	3.2(1)	**4.5**(2)*	**4.8**(2)$^{\star 2}$	**5.1**(2)$^{\star 2}$	**5.4**(2)$^{\star 2}$	15/15
2: NES	3.4(2)	8.7(4)	16(13)	16(13)	16(12)	15/15
f_9	35	214	300	335	369	15/15
1: CMA	5.8(2)	7.2(2)	6.4(2)	6.3(1)	6.2(1)	15/15
2: NES	6.4(2)	12(3)	11(6)	11(6)	11(6)	15/15
f_{10}	349	574	626	829	880	15/15
1: CMA	3.5(0.8)	2.7(0.4)	2.8(0.2)	2.3(0.2)	2.4(0.1)	15/15
2: NES	**2.0**(0.8)$^{\star 2}$	**1.8**(0.6)$^{\star 2}$	**2.0**(0.5)$^{\star 2}$	**1.9**(0.4)$^{\star 2}$	**2.0**(0.3)*	15/15
f_{11}	143	763	1177	1467	1673	15/15
1: CMA	8.4(3)	2.2(0.3)	1.6(0.2)	1.4(0.1)	1.3(0.1)	15/15
2: NES	**4.2**(3)$^{\star 2}$	**1.3**(0.3)$^{\star 3}$	**1.1**(0.2)$^{\star 3}$	**1.0**(0.1)$^{\star 3}$	**1.1**(0.1)$^{\star 3}$	15/15
f_{12}	108	371	461	1303	1494	15/15
1: CMA	11(12)	7.4(6)	7.7(5)	3.3(2)	3.3(2)	15/15
2: NES	16(28)	36(58)	51(97)	21(35)	31(34)	15/15
f_{13}	132	250	1310	1752	2255	15/15
1: CMA	3.9(3)	5.9(3)	1.6(0.3)	1.5(0.2)	1.7(0.8)	15/15
2: NES	3.3(0.6)	4.0(0.5)	**1.3**(0.2)$^{\star 2}$	1.4(0.2)	**1.5**(0.2)*	15/15
f_{14}	10	58	139	251	476	15/15
1: CMA	1.1(1.0)	3.7(0.9)	4.6(0.7)	5.4(0.5)	4.5(0.3)	15/15
2: NES	2.0(2)	3.9(1)	4.9(1)	**4.6**(0.5)$^{\star 2}$	**3.3**(0.3)$^{\star 3}$	15/15
f_{15}	511	19369	20073	20769	21359	14/15
1: CMA	1.6(2)	**1.2**(0.7)*	**1.2**(0.7)*	**1.2**(0.7)*	**1.2**(0.7)*	15/15
2: NES	3.6(6)	18(20)	18(19)	17(19)	16(18)	14/15
f_{16}	120	2662	10449	11644	12095	15/15
1: CMA	3.0(3)	2.6(1)	1.3(2)	1.4(2)	1.4(2)	15/15
2: NES	2.3(2)	1.7(3)	1.8(2)	1.6(2)	1.6(2)	15/15
f_{17}	5.2	899	3669	6351	7934	15/15
1: CMA	3.4(3)	1(2)	1(0.7)	1(0.5)	1.2(0.5)	15/15
2: NES	6.8(7)	1.1(0.7)	0.81(0.7)	1.4(1)	2.0(3)	15/15
f_{18}	103	3968	9280	10905	12469	15/15
1: CMA	1(0.7)	1(1)	1(0.3)	1.2(0.7)	1.3(0.6)	15/15
2: NES	0.80(0.5)	0.25(0.1)	**0.43**(0.5)$^{\star\downarrow}$	1.4(0.9)	2.0(3)	15/15
f_{19}	1	242	1.2e5	1.2e5	1.2e5	15/15
1: CMA	20(16)	161(175)	**1**(0.7)$^{\star 3}$	**1**(0.7)$^{\star 3}$	**1**(0.7)$^{\star 3}$	15/15
2: NES	17(18)	542(792)	11(9)	20(21)	21(21)	6/15
f_{20}	16	38111	54470	54861	55313	14/15
1: CMA	3.3(3)	**2.8**(1)*	**2.1**(0.8)*	**2.2**(0.8)*	**2.2**(0.8)*	15/15
2: NES	3.2(3)	21(25)	15(17)	15(17)	14(16)	12/15
f_{21}	41	1674	1705	1729	1757	14/15
1: CMA	2.3(2)	24(35)	25(36)	25(36)	25(36)	15/15
2: NES	12(1)	36(57)	35(56)	35(55)	35(54)	15/15
f_{22}	71	938	1008	1040	1068	14/15
1: CMA	6.9(11)	45(94)	42(88)	41(85)	40(83)	15/15
2: NES	46(61)	39(42)	37(39)	36(37)	35(37)	15/15
f_{23}	3.0	14249	31654	33030	34256	15/15
1: CMA	1.7(2)	**3.7**(4)$^{\star 3}$	**1.8**(2)$^{\star 3}$	**1.8**(2)$^{\star 3}$	**1.8**(2)$^{\star 3}$	15/15
2: NES	2.2(2)	∞	∞	∞	∞ *1.3e6*	0/15
f_{24}	1622	6.4e6	9.6e6	1.3e7	1.3e7	3/15
1: CMA	2.1(2)	1(1)	1(1)	1(1)	1(1)	3/15
2: NES	6.2(8)	∞	∞	∞	∞ *1.3e6*	0/15

20-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
f_1	43	43	43	43	43	15/15
1: CMA	7.9(2)	**20**(2)$^{\star 3}$	**33**(4)$^{\star 3}$	**45**(3)$^{\star 3}$	**57**(3)$^{\star 3}$	15/15
2: NES	7.3(2)	61(16)	88(23)	110(25)	128(32)	15/15
f_2	385	387	390	391	393	15/15
1: CMA	35(7)	44(4)	47(2)	48(2)	50(2)	15/15
2: NES	**26**(1)$^{\star 3}$	**34**(3)$^{\star 3}$	**38**(4)$^{\star 3}$	**41**(6)$^{\star 2}$	**43**(6)$^{\star 2}$	15/15
f_3	5066	7635	7643	7646	7651	15/15
1: CMA	**12**(7)$^{\star 3}$	∞$^{\star 3}$	∞$^{\star 3}$	∞$^{\star 3}$	∞ ***5.7e6***$^{\star 3}$	0/15
2: NES	1055(1344)	∞	∞	∞	∞ *1.3e7*	0/15
f_4	4722	7666	7700	7758	1.4e5	9/15
1: CMA	∞	∞	∞	∞	∞ *5.5e6*	0/15
2: NES	4193(4572)	∞	∞	∞	∞ *1.2e7*	0/15
f_5	41	41	41	41	41	15/15
1: CMA	**5.1**(0.8)$^{\star 3}$	**6.3**(1)$^{\star 3}$	**6.3**(1)$^{\star 3}$	**6.3**(1)$^{\star 3}$	**6.3**(1)$^{\star 3}$	15/15
2: NES	8.6(1)	11(2)	11(2)	11(2)	11(2)	15/15
f_6	1296	3413	5220	6728	8409	15/15
1: CMA	**1.5**(0.4)$^{\star 3}$	**1.2**(0.2)$^{\star 3}$	**1.1**(0.2)$^{\star 3}$	**1.2**(0.1)$^{\star 3}$	**1.2**(0.1)$^{\star 3}$	15/15
2: NES	4.8(0.2)	4.5(0.1)	4.8(0.1)	5.2(0.1)	5.3(0.1)	15/15
f_7	1351	9503	16524	16524	16969	15/15
1: CMA	**1**(0.5)$^{\star 2}$	3.5(0.6)	2.2(0.3)	2.2(0.3)	2.1(0.3)	15/15
2: NES	1.9(0.2)	**1.0**(0.1)$^{\star 3}$	**0.89**(0.1)$^{\star 3}$	**0.89**(0.1)$^{\star 3}$	**0.91**(0.1)$^{\star 3}$	15/15
f_8	2039	4040	4219	4371	4484	15/15
1: CMA	**4.0**(1)$^{\star 3}$	**4.3**(0.6)$^{\star 3}$	**4.5**(0.6)$^{\star 3}$	**4.6**(0.6)$^{\star 3}$	**4.6**(0.6)$^{\star 3}$	15/15
2: NES	7.2(0.6)	9.1(3)	9.4(4)	10(4)	10(4)	15/15
f_9	1716	3277	3455	3594	3727	15/15
1: CMA	**4.7**(2)$^{\star 3}$	**6.0**(1)$^{\star 2}$	**6.1**(1)$^{\star 2}$	**6.1**(1.0)$^{\star 2}$	**6.1**(0.9)$^{\star 2}$	15/15
2: NES	8.1(1)	10(2)	10(2)	11(2)	11(2)	15/15
f_{10}	7413	10735	14920	17073	17476	15/15
1: CMA	1.9(0.2)	1.6(0.1)	1.2(0.0)	1.1(0.0)	1.1(0.0)	15/15
2: NES	**1.3**(0.1)$^{\star 3}$	**1.3**(0.1)$^{\star 3}$	**1.0**(0.1)$^{\star 3}$	**0.99**(0.1)$^{\star 2}$	1.0(0.1)	15/15
f_{11}	1002	6278	9762	12285	14831	15/15
1: CMA	10(0.5)	1.9(0.1)	1.4(0.0)	1.2(0.0)	1.0(0.0)	15/15
2: NES	**4.8**(0.3)$^{\star 3}$	**1.4**(0.1)$^{\star 3}$	**1.1**(0.2)$^{\star 2}$	**1.00**(0.2)$^{\star 2}$	0.91(0.2)	15/15
f_{12}	1042	2740	4140	12407	13827	15/15
1: CMA	**3.0**(2)$^{\star 2}$	4.5(3)	**4.5**(2)$^{\star 2}$	**1.9**(0.7)$^{\star 2}$	**2.0**(0.7)$^{\star 3}$	15/15
2: NES	6.6(4)	18(18)	35(38)	21(21)	22(21)	15/15
f_{13}	652	2751	18749	24455	30201	15/15
1: CMA	4.3(6)	5.1(6)	1.5(0.8)	**2.3**(2)$^{\star 2}$	**3.0**(2)$^{\star 3}$	15/15
2: NES	7.0(3)	19(28)	17(21)	40(33)	81(83)	0/15
f_{14}	75	304	932	1648	15661	15/15
1: CMA	3.9(1)	**3.7**(0.4)$^{\star 3}$	**4.1**(0.3)$^{\star 3}$	**6.2**(0.5)$^{\star 3}$	**1.2**(0.1)$^{\star 3}$	15/15
2: NES	**2.6**(0.8)*	14(3)	12(1)	10(1)	1.5(0.1)	15/15
f_{15}	30378	3.1e5	3.2e5	4.5e5	4.6e5	15/15
1: CMA	**1**(0.4)$^{\star 3}$	**1.4**(0.5)$^{\star 3}$	**1.4**(0.5)$^{\star 3}$	**1**(0.3)$^{\star 3}$	**1**(0.3)$^{\star 3}$	15/15
2: NES	44(52)	∞	∞	∞	∞ *1.6e7*	0/15
f_{16}	1384	77015	1.9e5	2.0e5	2.2e5	15/15
1: CMA	**1.7**(0.4)$^{\star 3}$	**1.2**(0.7)$^{\star 2}$	**1**(0.7)$^{\star 3}$	**1**(0.7)$^{\star 3}$	**1**(0.7)$^{\star 3}$	15/15
2: NES	20(10)	9.2(7)	108(134)	133(157)	119(134)	6/15
f_{17}	63	4005	30677	56288	80472	15/15
1: CMA	2.2(2)	**1**(1)*	1.2(1)	1.3(0.6)	**1.4**(0.7)$^{\star 3}$	15/15
2: NES	2.1(1.0)	2.9(0.2)	0.92(0.0)	1.4(0.7)	12(9)	15/15
f_{18}	621	19561	67569	1.3e5	1.5e5	15/15
1: CMA	1.0(0.4)	1.2(0.9)	1.1(0.6)	1.7(0.7)	1.6(0.6)	15/15
2: NES	1.2(0.5)	0.81(0.0)$^{\downarrow 3}$	**0.48**(0.0)$^{\star 2\downarrow 4}$	**0.58**(0.3)$^{\star 2\downarrow 2}$	5.6(6)	15/15
f_{19}	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
1: CMA	169(74)	**1.2**(0.6)$^{\star 3}$	**1**(0.3)$^{\star 3}$	**1**(0.3)$^{\star 3}$	**1**(0.3)$^{\star 3}$	15/15
2: NES	**108**(35)*	∞	∞	∞	∞ *1.2e7*	0/15
f_{20}	82	3.1e6	5.5e6	5.6e6	5.6e6	14/15
1: CMA	4.3(1)	**1**(0.5)$^{\star 3}$	**1**(0.3)$^{\star 3}$	**1**(0.3)$^{\star 3}$	**1**(0.3)$^{\star 3}$	14/15
2: NES	5.4(3)	∞	∞	∞	∞ *1.2e7*	0/15
f_{21}	561	14103	14643	15567	17589	15/15
1: CMA	3.2(6)	48(98)	46(84)	43(85)	39(75)	13/15
2: NES	45(71)	37(57)	35(42)	33(40)	30(45)	30/30
f_{22}	467	23491	24948	26847	1.3e5	12/15
1: CMA	6.8(13)	215(309)	202(277)	188(247)	37(46)	5/15
2: NES	50(86)	226(238)	213(223)	198(208)	39(41)	22/30
f_{23}	3.2	67457	4.9e5	8.1e5	8.4e5	15/15
1: CMA	4.3(5)	**1**(0.8)$^{\star 5}$	**2.0**(1)$^{\star 5}$	**1.2**(0.9)$^{\star 5}$	**1.2**(0.9)$^{\star 5}$	15/15
2: NES	1.5(2)	∞	∞	∞	∞ *1.1e7*	0/30
f_{24}	1.3e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
1: CMA	**1**(0.9)$^{\star 5}$	**1**(1)$^{\star 5}$	**1**(1)$^{\star 5}$	**1**(1)$^{\star 5}$	**1**(1.0)$^{\star 5}$	3/15
2: NES	∞	∞	∞	∞	∞ *1.2e7*	0/30

Table 2: ERT in number of function evaluations divided by the best ERT measured during BBOB-2009 given in the respective first row with the central 80% range divided by two in brackets for different Δf values. #succ is the number of trials that reached the final target $f_{\rm opt} + 10^{-8}$. 1:CMA is BIPOP-CMA-ES and 2:NES is xNES-as. Bold entries are statistically significantly better compared to the other algorithm, with $p = 0.05$ or $p = 10^{-k}$ where $k \in \{2, 3, 4, \dots\}$ is the number following the $\star$ symbol, with Bonferroni correction of 48. A $\downarrow$ indicates the same tested against the best BBOB-2009.

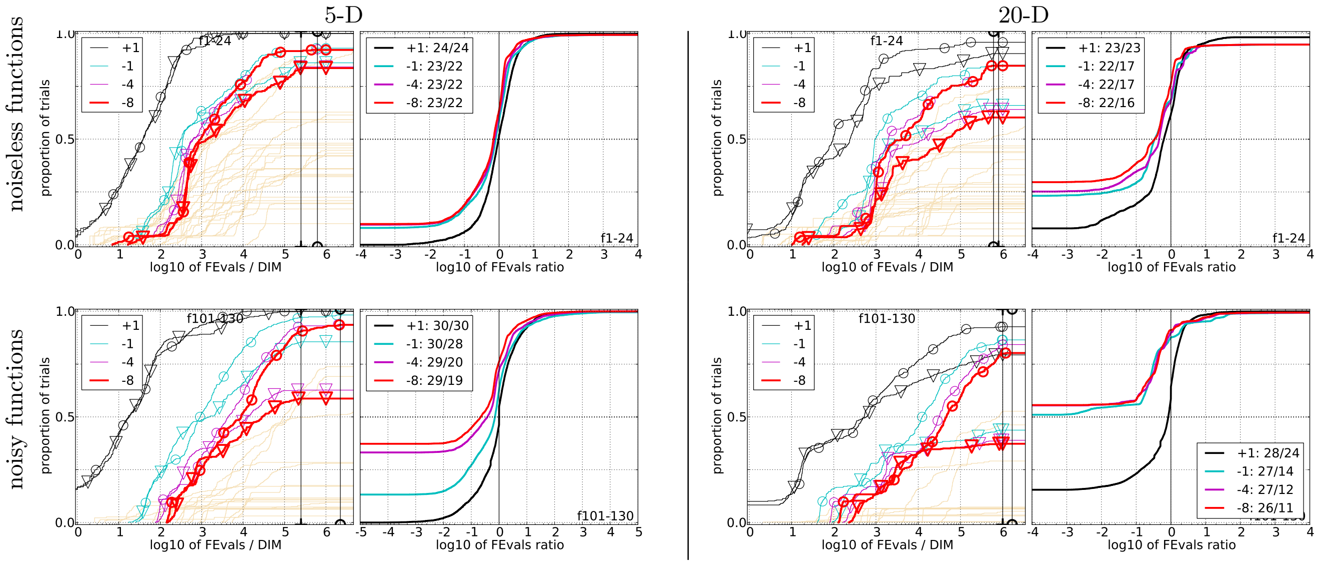

Figure 2: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/D) to reach a target value $f_{\mathrm{opt}} + \Delta f$ with $\Delta f = 10^k$, where $k \in \{1, -1, -4, -8\}$ is given by the first value in the legend, for BIPOP-CMA-ES (○) and xNES-as (▽). Light beige lines show the ECDF of FEvals for target value $\Delta f = 10^{-8}$ of all algorithms benchmarked during BBOB-2009. Right sub-columns: ECDF of FEval ratios of BIPOP-CMA-ES divided by xNES-as, 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 (BIPOP-CMA-ES first).

5-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
f101	11	44	62	69	75	15/15
1: CMA	3.2(2)	4.6(0.9)	**6.1**(0.5)*	**8.0**(0.4)*	**10**(0.7)*	15/15
2: NES	2.8(2)	5.2(2)	7.9(2)	10(2)	13(3)	15/15
f102	11	50	72	86	99	15/15
1: CMA	2.7(2)	4.0(0.6)	**5.1**(0.5)$^{\star 4}$	**6.3**(0.5)$^{\star 3}$	**7.2**(0.7)$^{\star 4}$	15/15
2: NES	3.3(3)	4.7(1)	7.3(1)	9.2(1)	10(2)	15/15
f103	11	30	31	35	115	15/15
1: CMA	3.5(4)	7.4(1)	**13**(1)$^{\star 2}$	**17**(2)$^{\star 4}$	**6.9**(0.9)$^{\star 4}$	15/15
2: NES	2.7(2)	7.2(2)	16(3)	24(5)	11(2)	15/15
f104	173	1287	1768	2040	2284	15/15
1: CMA	1.4(0.3)	2.0(0.3)	2.0(0.2)	1.9(0.2)	1.8(0.2)	15/15
2: NES	1.4(0.4)	2.7(6)	2.3(4)	2.1(4)	2.0(3)	15/15
f105	167	5174	10388	10824	11202	15/15
1: CMA	1.7(0.4)	**1.7**(0.9)$^{\star 2}$	**1**(0.4)$^{\star 3}$	**1**(0.4)$^{\star 3}$	**1**(0.4)$^{\star 3}$	15/15
2: NES	1.4(0.5)	14(13)	11(7)	11(7)	10(7)	15/15
f106	92	1050	2666	2887	3087	15/15
1: CMA	3.3(0.9)	3.2(3)	1.6(1)	1.7(1)	1.7(1)	15/15
2: NES	2.7(0.8)	3.7(5)	2.2(3)	2.2(3)	2.7(3)	15/15
f107	40	453	940	1376	1850	15/15
1: CMA	1.7(2)	1(0.5)	**1**(0.3)$^{\star 2}$	**1**(0.2)$^{\star 4}$	**1**(0.2)$^{\star 4}$	15/15
2: NES	3.2(5)	4.2(0.5)	6.9(11)	7.2(8)	7.0(10)	15/15
f108	87	14469	30935	58628	80667	15/15
1: CMA	6.1(10)	**1**(0.8)$^{\star 2}$	**1**(0.6)$^{\star 4}$	**1**(0.4)$^{\star 4}$	**1**(0.3)$^{\star 4}$	15/15
2: NES	69(120)	56(59)	∞	∞	∞ *1.3e6*	0/15
f109	11	216	572	873	946	15/15
1: CMA	3.5(2)	1.1(0.3)	**1.1**(0.2)$^{\star 2}$	**1.1**(0.3)$^{\star 4}$	**1.5**(0.3)$^{\star 4}$	15/15
2: NES	3.6(2)	1.3(0.5)	1.8(0.5)	2.3(0.6)	4.1(0.8)	15/15
f110	949	1.2e5	5.9e5	6.0e5	6.1e5	15/15
1: CMA	1(1)	3.7(4)	1(0.7)	1(0.7)	1(0.6)	15/15
2: NES	0.44(0.2)	3.6(5)	1.2(1)	1.6(2)	2.3(3)	8/15
f111	6856	8.8e6	2.3e7	3.1e7	3.1e7	3/15
1: CMA	1(1.0)	1(1)	1(0.9)	1(1.0)	1(1.0)	3/15
2: NES	5.4(8)	0.96(1)	∞	∞	∞ *1.3e6*	0/15
f112	107	3421	4502	5132	5596	15/15
1: CMA	4.0(2)	1.2(0.2)	**1.3**(0.2)$^{\star 2}$	**1.3**(0.2)$^{\star 3}$	**1.3**(0.2)$^{\star 4}$	15/15
2: NES	2.3(1)	7.9(10)	17(18)	28(23)	37(38)	15/15
f113	133	8081	24128	24128	24402	15/15
1: CMA	1.5(1.0)	1.7(2)	1.1(1)	1.1(1)	1.1(1)	15/15
2: NES	1.2(1)	2.5(4)	1.4(2)	1.4(2)	1.5(2)	15/15
f114	767	56311	83272	83272	84949	15/15
1: CMA	2.2(2)	**1**(0.7)$^{\star 4}$	**1**(0.7)$^{\star 4}$	**1**(0.7)$^{\star 4}$	**1**(0.7)$^{\star 4}$	15/15
2: NES	8.0(13)	157(178)	∞	∞	∞ *1.3e6*	0/15
f115	64	1829	2550	2550	2970	15/15
1: CMA	1.5(0.8)	6.5(7)	5.9(6)	5.9(6)	5.7(5)	15/15
2: NES	1.3(0.9)	**1.8**(4)*	**1.8**(4)*	**1.8**(4)*	**1.6**(3)*	15/15

20-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
f101	59	571	700	739	783	15/15
1: CMA	6.1(1)	**1.6**(0.1)$^{\star 5}$	**2.1**(0.1)$^{\star 5}$	**2.7**(0.1)$^{\star 5}$	**3.3**(0.2)$^{\star 5}$	15/15
2: NES	4.8(2)	5.7(2)	6.7(3)	7.7(2)	8.4(3)	30/30
f102	231	579	921	1157	1407	15/15
1: CMA	1.6(0.3)	**1.6**(0.2)$^{\star 5}$	**1.6**(0.1)$^{\star 5}$	**1.8**(0.1)$^{\star 5}$	**1.8**(0.1)$^{\star 5}$	15/15
2: NES	1.5(0.5)	6.2(2)	5.6(2)	5.4(2)	5.4(2)	30/30
f103	65	629	1313	1893	2464	14/15
1: CMA	5.5(1)	**1.5**(0.1)$^{\star 5}$	**1.2**(0.1)$^{\star 5}$	**1.2**(0.1)$^{\star 5}$	**1.2**(0.1)$^{\star 5}$	15/15
2: NES	5.0(2)	4.9(2)	4.1(2)	5.5(2)	6.8(2)	30/30
f104	23690	1.7e5	1.8e5	1.9e5	2.0e5	15/15
1: CMA	10(7)	**1.7**(1)$^{\star 5}$	**1.6**(1)$^{\star 5}$	**1.6**(1.0)$^{\star 5}$	**1.6**(0.9)$^{\star 5}$	15/15
2: NES	24(29)	∞	∞	∞	∞ *1.8e7*	0/30
f105	1.9e5	6.3e5	6.5e5	6.6e5	6.7e5	15/15
1: CMA	2.7(2)	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	15/15
2: NES	24(22)	∞	∞	∞	∞ *1.6e7*	0/21
f106	11480	23746	25470	26492	27360	15/15
1: CMA	1.0(0.3)	**1.4**(1)*	1.5(1)	**1.5**(1)$^{\star 2}$	**1.5**(1)$^{\star 2}$	15/15
2: NES	1.2(0.2)	1.5(0.3)	1.7(0.3)	1.8(0.7)	2.0(0.8)	30/30
f107	8571	16226	27357	52486	65052	15/15
1: CMA	1(0.4)	**1**(0.6)$^{\star 4}$	**1**(0.4)$^{\star 4}$	**1**(0.8)$^{\star 4}$	**1**(0.8)$^{\star 4}$	15/15
2: NES	1.6(0.9)	786(980)	∞	∞	∞ *1.5e7*	0/15
f108	58063	2.0e5	4.5e5	6.3e5	9.0e5	15/15
1: CMA	**1**(0.5)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.4)$^{\star 4}$	15/15
2: NES	∞	∞	∞	∞	∞ *1.1e7*	0/15
f109	333	1138	2287	3583	4952	15/15
1: CMA	1.2(0.3)	**1.1**(0.2)$^{\star 4}$	**1.1**(0.1)$^{\star 4}$	**1.1**(0.1)$^{\star 4}$	**1.0**(0.1)$^{\star 4}$	15/15
2: NES	1.0(0.2)	6.9(0.8)	9.4(0.7)	10(0.5)	13(8)	15/15
f110	∞	∞	∞	∞	∞	0
1: CMA	∞	∞	∞	∞	∞	0/15
2: NES	∞	∞	∞	∞	∞	0/15
f111	∞	∞	∞	∞	∞	0
1: CMA	∞	∞	∞	∞	∞	0/15
2: NES	∞	∞	∞	∞	∞	0/15
f112	25552	69621	73557	76137	78238	15/15
1: CMA	1(0.3)	**1.1**(0.8)$^{\star 4}$	**1.2**(0.7)$^{\star 4}$	**1.2**(0.7)$^{\star 4}$	**1.2**(0.7)$^{\star 4}$	15/15
2: NES	0.99(0.2)	952(854)	3878(4060)	3747(3812)	3646(4032)	1/15
f113	50123	5.6e5	5.9e5	5.9e5	5.9e5	15/15
1: CMA	1(1.0)	**1**(0.4)$^{\star 4}$	**1**(0.4)$^{\star 4}$	**1**(0.4)$^{\star 4}$	**1**(0.4)$^{\star 4}$	15/15
2: NES	1.4(1)	∞	∞	∞	∞ *1.5e7*	0/15
f114	2.1e5	1.4e6	1.6e6	1.6e6	1.6e6	15/15
1: CMA	**1**(0.4)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.5)$^{\star 4}$	15/15
2: NES	∞	∞	∞	∞	∞ *1.0e7*	0/15
f115	2405	91749	1.3e5	1.3e5	1.3e5	15/15
1: CMA	1(1)	3.9(2)	3.0(1)	3.0(1)	3.0(2)	15/15
2: NES	1.0(0.1)	**0.14**(0.0)$^{\star 4 \downarrow}$	**0.37**(0.3)$^{\star 3 \downarrow 3}$	**0.37**(0.3)$^{\star 3 \downarrow 3}$	**0.46**(0.5)$^{\star 3 \downarrow 2}$	15/15

Table 3: Relative ERT in number of f-evaluations, see Table 2 for details.

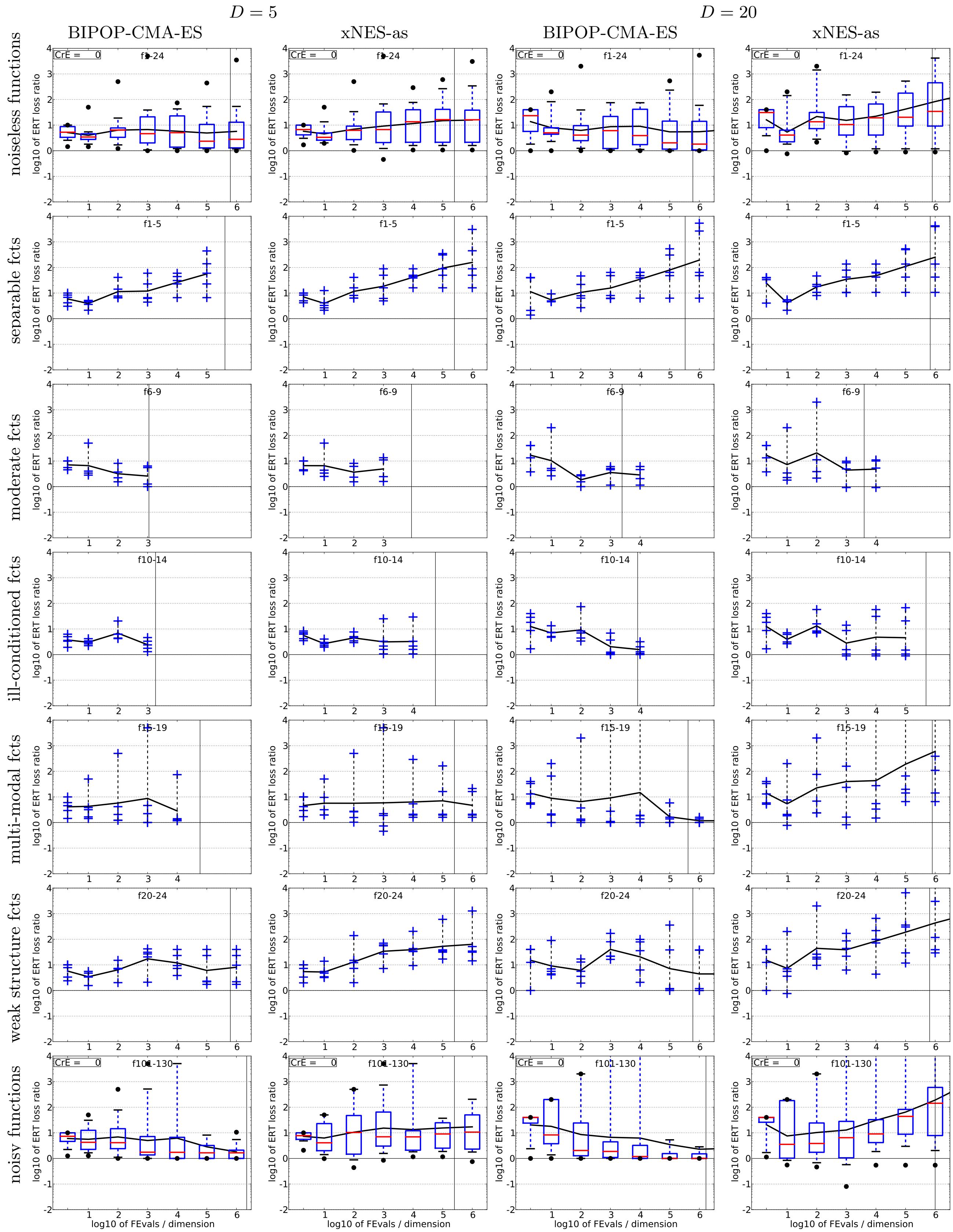

Figure 3: ERT loss ratio vs. a given budget FEvals. Each cross (+) represents a single function. The target value f_t used for a given FEvals is the smallest (best) recorded function value such that $\mathrm{ERT}(f_t) \leq$ FEvals for the presented algorithm. Shown is FEvals divided by the respective best $\mathrm{ERT}(f_t)$ from BBOB-2009 for functions f_1–f_{24} in 5-D and 20-D. Line: geometric mean. Box-Whisker error bar: 25-75%-ile with median (box), 10-90%-ile (caps), and minimum and maximum ERT loss ratio (points). The vertical line gives the maximal number of function evaluations in a single trial in this function subset.

5-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
f116	5730	22311	26868	30329	31661	15/15
1: CMA	1.2(1)	1.9(2)	2.1(2)	2.0(2)	2.0(2)	15/15
2: NES	1.4(2)	1.1(2)	1.3(1)	1.6(2)	1.8(2)	15/15
f117	26686	1.1e5	1.4e5	1.7e5	1.9e5	15/15
1: CMA	**1**(0.7)$^{\star 2}$	**1**(0.7)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.5)$^{\star 4}$	15/15
2: NES	8.4(6)	∞	∞	∞	∞ *1.3e6*	0/15
f118	429	1555	1998	2430	2913	15/15
1: CMA	3.2(1)	1.9(0.7)	2.1(0.4)	2.0(0.4)	1.8(0.3)	15/15
2: NES	**1.00**(0.4)$^{\star 4}$	**0.48**(0.1)$^{\star 4 \downarrow 4}$	**0.69**(0.2)$^{\star 4 \downarrow 2}$	**0.96**(0.2)$^{\star 4}$	**1.1**(0.2)$^{\star 3}$	15/15
f119	12	1136	10372	35296	49747	15/15
1: CMA	1.9(3)	1(2)	1(0.6)	1.5(0.8)	2.3(1)	15/15
2: NES	3.8(4)	6.3(13)	1.6(2)	2.3(3)	5.9(6)	15/15
f120	16	18698	72438	3.3e5	5.5e5	15/15
1: CMA	17(16)	1(0.6)	**1**(0.8)$^{\star 4}$	**1**(0.5)$^{\star 4}$	**1**(0.4)$^{\star 4}$	15/15
2: NES	51(18)	25(38)	∞	∞	∞ *1.3e6*	0/15
f121	8.6	273	1583	3870	6195	15/15
1: CMA	2.7(3)	1(0.2)	1.1(0.5)	2.0(0.2)	2.2(0.2)	15/15
2: NES	2.6(2)	0.96(0.5)	1.6(0.8)	1.5(2)	2.3(3)	15/15
f122	10	9190	30087	53743	1.1e5	15/15
1: CMA	2.2(2)	**1**(0.8)$^{\star 3}$	**1**(0.5)$^{\star 3}$	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	15/15
2: NES	5.0(4)	8.9(8)	583(686)	∞	∞ *1.3e6*	0/15
f123	11	81505	3.4e5	6.7e5	2.2e6	15/15
1: CMA	8.1(11)	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	1(0.9)	15/15
2: NES	6.6(9)	∞	∞	∞	∞ *1.3e6*	0/15
f124	10	1040	20478	45337	95200	15/15
1: CMA	1.5(2)	1(0.3)	1.1(0.7)	**1.2**(1.0)$^{\star 3}$	**1**(0.5)$^{\star 3}$	15/15
2: NES	2.9(3)	2.0(0.7)	1.1(1)	36(41)	60(60)	1/15
f125	1	1	2.4e5	2.4e5	2.5e5	15/15
1: CMA	1.1	3443(2609)	**1**(0.7)$^{\star 4}$	**1**(0.7)$^{\star 4}$	**1**(0.7)$^{\star 4}$	15/15
2: NES	1.3(0.5)	7786(7916)	∞	∞	∞ *1.3e6*	0/15
f126	1	1	∞	∞	∞	0
1: CMA	1	13292(10642)	∞	∞	∞	0/15
2: NES	1.1(0.5)	41598(46392)	∞	∞	∞	0/15
f127	1	1	3.4e5	3.9e5	4.0e5	15/15
1: CMA	1	2136(1530)	**1**(1.0)$^{\star 4}$	**1**(0.8)$^{\star 4}$	**1**(0.8)$^{\star 4}$	15/15
2: NES	1.1(0.5)	3858(5345)	∞	∞	∞ *1.3e6*	0/15
f128	111	7808	12447	17217	21162	15/15
1: CMA	2.2(2)	10(17)	6.6(11)	4.8(8)	3.9(6)	15/15
2: NES	22(67)	5.7(7)	3.6(4)	2.6(3)	2.1(2)	15/15
f129	64	59443	2.8e5	5.1e5	5.8e5	15/15
1: CMA	12(15)	9.2(2)	**3.9**(12)*	**2.2**(7)*	**1.9**(6)*	13/15
2: NES	17(16)	8.6(12)	14(15)	∞	∞ *1.3e6*	0/15
f130	55	3034	32823	33889	34528	10/15
1: CMA	1.9(1)	55(101)	5.1(9)	5.0(9)	5.0(9)	15/15
2: NES	17(0.9)	28(48)	2.6(4)	2.5(4)	2.5(4)	15/15

20-D

Δf	1e+1	1e-1	1e-3	1e-5	1e-7	#succ
f116	5.0e5	8.9e5	1.0e6	1.1e6	1.1e6	15/15
1: CMA	**1.4**(0.9)$^{\star 3}$	**1.1**(0.5)$^{\star 4}$	**1**(0.4)$^{\star 4}$	**1**(0.4)$^{\star 4}$	**1**(0.4)$^{\star 4}$	15/15
2: NES	39(47)	∞	∞	∞	∞ *1.6e7*	0/15
f117	1.8e6	2.6e6	2.9e6	3.2e6	3.6e6	15/15
1: CMA	**1**(0.5)$^{\star 5}$	**1**(0.2)$^{\star 5}$	**1**(0.2)$^{\star 5}$	**1**(0.2)$^{\star 5}$	**1**(0.2)$^{\star 5}$	15/15
2: NES	∞	∞	∞	∞	∞ *8.1e6*	0/30
f118	6908	17514	26342	30062	32659	15/15
1: CMA	1.9(0.4)	1.6(0.2)	1.5(0.1)	1.6(0.1)	**1.6**(0.1)$^{\star 2}$	15/15
2: NES	**0.94**(0.1)$^{\star 2}$	**0.80**(0.1)$^{\star 4}$	**1.1**(0.1)$^{\star 4}$	1.7(0.7)	2.1(0.7)	15/15
f119	2771	35930	4.1e5	1.4e6	1.9e6	15/15
1: CMA	1.6(1)	**1**(1)$^{\star 3}$	**1**(0.5)$^{\star 3}$	**1.3**(0.3)$^{\star 3}$	**1.1**(0.2)$^{\star 3}$	15/15
2: NES	0.53(0.4)$^{\downarrow}$	851(777)	∞	∞	∞ *1.4e7*	0/13
f120	36040	2.8e5	1.6e6	6.7e6	1.4e7	13/15
1: CMA	**1**(0.6)*	**1**(0.6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.4)$^{\star 4}$	1(0.4)	13/15
2: NES	6.9(7)	∞	∞	∞	∞ *1.1e7*	0/15
f121	249	1426	9304	34434	57404	15/15
1: CMA	1.2(0.5)	**1.2**(0.3)$^{\star 4}$	**1.1**(0.2)$^{\star 4}$	1.3(0.1)	1.9(0.1)	15/15
2: NES	0.81(0.3)	6.3(0.5)	2.9(0.2)	1.7(0.6)	1.7(1)	15/15
f122	692	1.4e5	7.9e5	2.0e6	5.8e6	15/15
1: CMA	1.8(2)	**1**(0.7)$^{\star 3}$	**1**(0.7)$^{\star 3}$	**1**(0.5)$^{\star 3}$	**1**(0.8)$^{\star 3}$	15/15
2: NES	0.81(0.9)	∞	∞	∞	∞ *1.2e7*	0/13
f123	1063	1.5e6	5.3e6	2.7e7	1.6e8	0
1: CMA	5.7(4)	**1**(0.7)$^{\star 4}$	**1**(0.6)$^{\star 4}$	1(0.9)	1(1)	0/15
2: NES	11(14)	∞	∞	∞	∞ *1.1e7*	0/15
f124	192	40840	1.3e5	3.9e5	8.0e5	15/15
1: CMA	1.1(0.5)	1(1.0)	**1**(0.9)*	**1**(0.8)$^{\star 4}$	**1**(0.4)$^{\star 4}$	15/15
2: NES	0.84(0.3)	0.59(0.1)	4.0(3)	∞	∞ *1.8e7*	0/15
f125	1	1	2.5e7	8.0e7	8.1e7	4/15
1: CMA	1	**9.8e6**(7e6)$^{\star 4}$	1(0.9)	1(1)	1(1)	4/15
2: NES	1.7(1)	∞	∞	∞	∞ *1.1e7*	0/15
f126	1	1	∞	∞	∞	0
1: CMA	1	∞$^{\star 4}$	∞$^{\star 4}$	∞$^{\star 4}$	∞$^{\star 4}$	0/15
2: NES	1.4(1)	∞	∞	∞	∞	0/15
f127	1	1	4.4e6	7.3e6	7.4e6	15/15
1: CMA	1	**9.0e5**(1e6)$^{\star 4}$	**1**(0.6)$^{\star 4}$	**1**(0.7)$^{\star 4}$	**1**(0.7)$^{\star 4}$	15/15
2: NES	1.5(0.5)	∞	∞	∞	∞ *1.1e7*	0/15
f128	1.4e5	1.7e7	1.7e7	1.7e7	1.7e7	9/15
1: CMA	**1**(2)$^{\star 2}$	1(1)	1(1)	1(1)	1(1)	9/15
2: NES	19(23)	0.78(0.8)	1.0(1)	1.0(1)	1.3(1)	6/15
f129	7.8e6	4.2e7	4.2e7	4.2e7	4.2e7	5/15
1: CMA	**1**(1)$^{\star 3}$	1(1)	1(1)	1(1)	1(1)	5/15
2: NES	∞	∞	∞	∞	∞ *1.1e7*	0/15
f130	4904	2.5e5	2.5e5	2.6e5	2.6e5	7/15
1: CMA	1.9(4)	14(28)	14(27)	14(27)	14(27)	15/15
2: NES	15(23)	5.4(9)	5.4(9)	5.4(9)	5.4(9)	15/15

Table 4: Relative ERT **in number of f-evaluations, see Table 2 for details.**

[10] 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.

[11] 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. Updated February 2010.

[12] N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *IEEE Transactions on Evolutionary Computation*, 9:159–195, 2001.

[13] 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.

[14] F. Rothlauf, editor. *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009, Companion Material*. ACM, 2009.

[15] T. Schaul. *Studies in Continuous Black-box Optimization*. Ph.D. thesis, Technische Universität München, 2011.

[16] T. Schaul. Benchmarking Natural Evolution Strategies with Adaptation Sampling on the Noiseless and Noisy Black-box Optimization Testbeds. In *Black-box Optimization Benchmarking Workshop, Genetic and Evolutionary Computation Conference*, Philadelphia, PA, 2012.

[17] T. Schaul. Natural Evolution Strategies Converge on Sphere Functions. In *Genetic and Evolutionary Computation Conference (GECCO)*, Philadelphia, PA, 2012.

[18] T. Schaul, J. Bayer, D. Wierstra, Y. Sun, M. Felder, F. Sehnke, T. Rückstieß, and J. Schmidhuber. PyBrain. *Journal of Machine Learning Research*, 11:743–746, 2010.

[19] T. Schaul and J. Schmidhuber. Metalearning. *Scholarpedia*, 5(6):4650, 2010.

[20] Y. Sun, D. Wierstra, T. Schaul, and J. Schmidhuber. Stochastic search using the natural gradient. In *International Conference on Machine Learning (ICML)*, 2009.

[21] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, and J. Schmidhuber. Natural Evolution Strategies. Technical report, 2011.

[22] D. Wierstra, T. Schaul, J. Peters, and J. Schmidhuber. Natural Evolution Strategies. In *Proceedings of the IEEE Congress on Evolutionary Computation (CEC)*, Hong Kong, China, 2008.