SPSA with Hessian Approximation on the Noisy Function Testbed

Steffen Finck
Vorarlberg University of Applied Sciences
Hochschulstrasse 1
Dornbirn, Austria
steffen.finck@fhv.at

ABSTRACT

This paper benchmarks the Simultaneous Perturbation Stochastic Algorithm (SPSA) with Hessian Matrix Approximation [5] on the BBOB 2009 noisy testbed. SPSA is a widely used optimization algorithm with its main application in noisy optimization. The paper presents briefly the algorithm and used parameter setting for the testbed.

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, stochastic optimization

1. INTRODUCTION

The SPSA algorithm [4] is a very common and widely used optimization algorithm and primarily designed for noisy optimization. In this paper the basic variant is coupled with an additional iteration for the approximation of the Hessian Matrix. As for the basic variant a multistart procedure is used to effectively use the computational resources and increase the convergence towards the target value f_{opt}.

2. ALGORITHM PRESENTATION

In Fig. 1 the main algorithm is presented.

3. EXPERIMENTAL PROCEDURE

As recommend the gain rates are set to $\texttt{alpha} = 0.602$ and $\texttt{gamma} = 0.101$.

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

4. RESULTS

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

5. CPU TIMING EXPERIMENT

For the timing experiment the same multistart algorithm was run on f_8 and restarted until at least 30 seconds had passed (according to Figure 2 in [2]). The results were $7.6; 7.6; 7.9; 7.9; 7.7$ and 18×10^{-4} seconds per function evaluation in dimension $2; 3; 5; 10; 20$ and 40, respectively. The dependency of CPU time on the search space dimensionality is negligible for the small dimensions, but for DIM $= 40$ the time increases by factor about 2.5.

6. CONCLUSION

This paper reports the result for the SPSA with Hessian approximation on the BBOB 2009 noisy testbed.

Acknowledgments

The author would like to acknowledge the great work of the BBOB team with particular kudos the Anne Auger, Nikolaus Hansen and Raymond Ros. This work was supported by the Austrian Science Fund (FWF) under grant P19069-N18.

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

[3] 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.

[4] J. C. Spall. Multivariate Stochastic Approximation Using a Simultaneous Perturbation Gradient Approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, March 1992.

[5] J. C. Spall. Feedback and Weighting Mechanisms for Improving Jacobian Estimates in the Adaptive Simultaneous Perturbation Algorithm. *IEEE Transactions on Automatic Control*, 54, 2009.

Figure 1: SPSA with Hessian approximation in Matlab

```
% SPSA2 with Feedback and Weighting Mechanism for BBOB Workshop

function SPSA2(FUN, DIM, ftarget, maxfunevals)

    % multistart such that ftarget is reached with reasonable prob.
    for ilaunch = 1:100;  % relaunch optimizer up to 100 times

        if ilaunch == 1                    % initial scenario
           xstart = 8 * rand(DIM, 1) - 4;
           lambda = 1;
        else

            choice = round(2*rand) + 1;

            switch choice

                case 1     % new point
                    xstart = 8 * rand(DIM, 1) - 4;

                case 2     % improve old point
                    if max(abs(x)) < 5
                        xstart = x;
                    else
                        xstart = 8 * rand(DIM, 1) - 4;
                    end

                case 3     % increase lambda
                    lambda = ceil(lambda * sqrt(2));

            end % switch case

        end

        % try spsa
        [x,termvalue] = alg(FUN,xstart, DIM,ftarget,maxfunevals,lambda);

        if termvalue == 1
            break;
        end

    end

end % of function


function [x,termvalue] = alg(FUN,x, DIM, ftarget, maxfunevals,lambda)

    % initialize parameter
    alpha = 0.602;
    gamma = 0.101;
    if isinf(maxfunevals)
        kmax = 1e5*DIM;
    else
        kmax = maxfunevals/4/lambda;
    end

    A = kmax*0.1;

    % initialize counters
    k = 0; % iteration counter

    % initialize hessian matrix and sum of loss measurements
    HkBar = eye(DIM);
    HkBarBar = zeros(DIM);
    Gk = zeros(DIM,1);
    dGk = zeros(DIM,1);
    Psik = zeros(DIM);
    sumck = 0;
    HkHat = zeros(DIM);

    % determine initial parameter
    % a0
    a0 = 1;
    X = repmat(x,1,10);
    % c0
    dummy = feval(FUN,X);
    c0 = max([5*std(dummy,1),1e-5]);
    c0Bar = 2*c0;

    while k < kmax

        % gain sequences
        ck = c0*(k+1)^(-gamma);
        ckBar = c0Bar*(k+1)^(-gamma);
        sumck = sumck + ck^2*ckBar^2;
        ak = a0*(k + 1 + A)^(-alpha);

        % gradient and hessian approximation
        % generation of the simultaneous perturbation vector
        for i = 1:lambda

            delta = 2*round(rand(DIM,1))-1;    % for gradient recursion
            deltaH = 2*round(rand(DIM,1))-1;   % for hessian recursion

            % function evaluation
            yplus = FUN(x + ck.*delta);
            yminus = FUN(x - ck.*delta);
            yplusH = FUN(x + ck.*delta + ckBar.*deltaH);
            yminusH = FUN(x - ck.*delta + ckBar.*deltaH);

            % gradient approximation
            Gk = (i-1)/i*Gk + 1/i*(yplus-yminus)./(2*ck*delta);

            % gradient approximation for hessian matrix
            dGk = (i-1)/i*dGk + 1/i*((yplusH-yplus)./(ckBar.*deltaH) - ...
                  (yminusH-yminus)./(ckBar.*deltaH));

            hhat = dGk./(2*ck)*(delta.^(-1))';
            HkHat = (i-1)/i*HkHat + 1/(2*i)*(hhat + hhat');

            % feedback term
            Dk = delta*(1./delta)' - eye(DIM);
            DkBar = deltaH*(1./deltaH)' - eye(DIM);
            psik = DkBar'*HkBarBar*Dk + DkBar'*HkBarBar+HkBarBar*Dk;
            Psik = (i-1)/i*Psik + 1/(2*i)*(psik + psik');
        end

        % weights
        wk = ck^2*ckBar^2/sumck; % for noisy testbed

        % hessian matrix recursion
        HkBar = (1-wk)*HkBar + wk*(HkHat - Psik);
        HkBarBar = diag(diag(HkBar + 1e-3*exp(-k+1)*eye(DIM)));

        % update of the search point
        xnew = x - ak*(HkBarBar\Gk);

        % blocking
        if max(abs(xnew - x)) < 10
            x = xnew;
        end

        % termination criteria
        fit = feval(FUN,x);
        if max(isnan(x)) == 1 || max(isinf(x)) == 1 || fit > 1e30
            termvalue = 0;
            break;
        end

        if feval(FUN, 'fbest') < ftarget || ...
           feval(FUN, 'evaluations') >= maxfunevals
            termvalue = 1;
            break;
        end

        k = k + 1;

    end % of iteration

end % of function
```

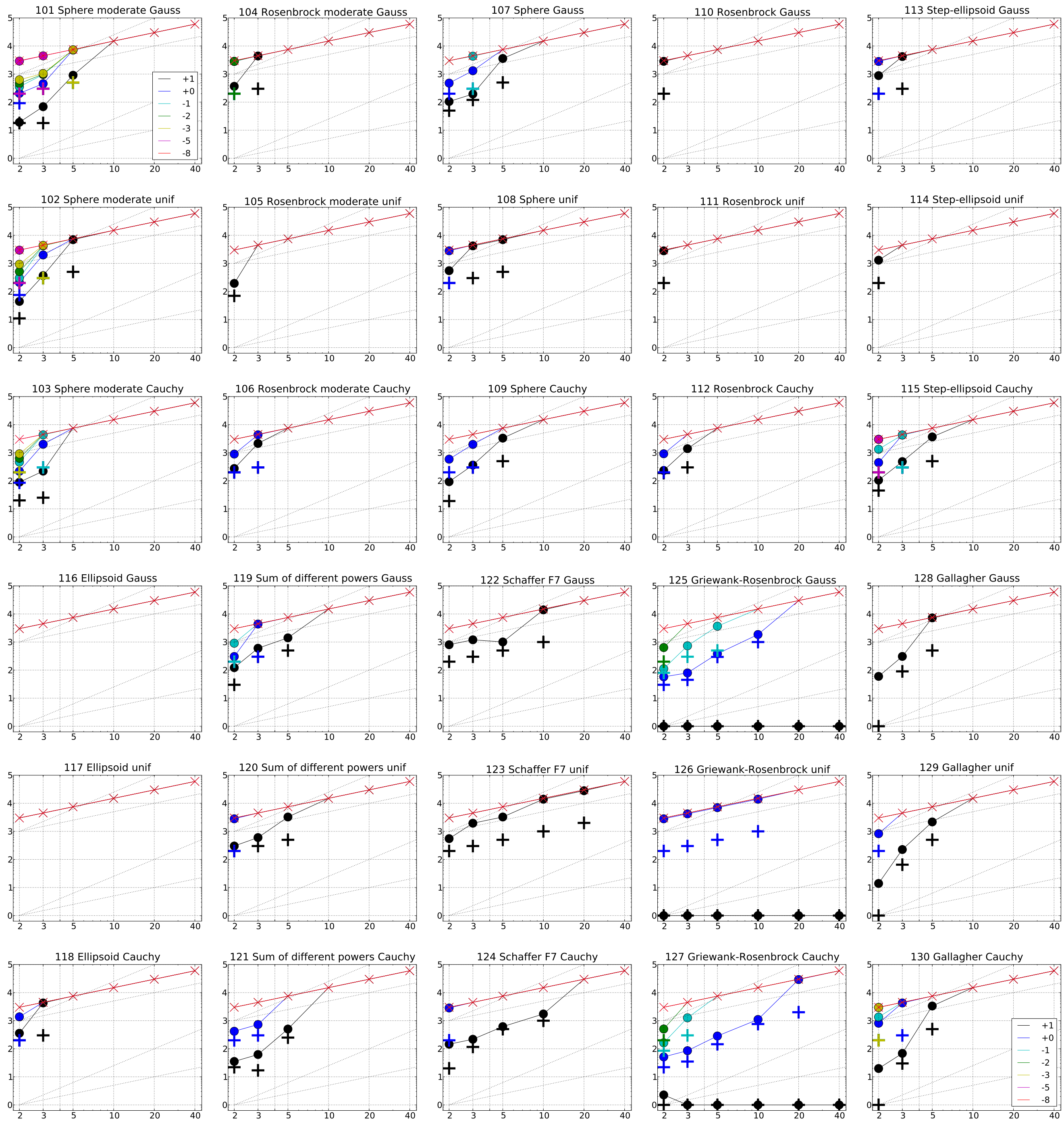

Figure 2: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_{101} and f_{130}) versus dimension in 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	**f101 in 5-D**, N=15, mFE=500					**f101 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	6	9.1e2	7.4e2	1.1e3	2.9e2	0	*19e+1*	*13e+1*	*30e+1*	1.4e1
1	1	7.2e3	6.9e3	7.5e3	5.0e2	.	.	.	.	.
1e−1	1	7.2e3	7.0e3	7.5e3	5.0e2	.	.	.	.	.
1e−3	1	7.3e3	7.2e3	7.5e3	5.0e2	.	.	.	.	.
1e−5	0	*10e+0*	*20e−1*	*27e+0*	2.2e2	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f102 in 5-D**, N=15, mFE=500					**f102 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	1	7.0e3	6.5e3	7.5e3	5.0e2	0	*23e+1*	*15e+1*	*28e+1*	3.9e1
1	0	*26e+0*	*11e+0*	*52e+0*	2.2e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f103 in 5-D**, N=15, mFE=500					**f103 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*46e+0*	*25e+0*	*97e+0*	1.9e1	0	*22e+1*	*15e+1*	*30e+1*	2.8e1
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f104 in 5-D**, N=15, mFE=500					**f104 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*48e+2*	*24e+1*	*14e+4*	2.8e2	0	*47e+4*	*54e+3*	*83e+4*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f105 in 5-D**, N=15, mFE=500					**f105 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*24e+3*	*33e+2*	*22e+4*	2.2e1	0	*34e+4*	*17e+4*	*50e+4*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f106 in 5-D**, N=15, mFE=500					**f106 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*12e+3*	*68e+1*	*31e+4*	2.8e2	0	*30e+4*	*14e+4*	*84e+4*	2.2e1
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f107 in 5-D**, N=15, mFE=500					**f107 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	2	3.6e3	3.4e3	3.8e3	5.0e2	0	*21e+1*	*12e+1*	*24e+1*	1.0e0
1	0	*21e+0*	*79e−1*	*62e+0*	4.0e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f108 in 5-D**, N=15, mFE=500					**f108 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	1	7.0e3	6.5e3	7.5e3	1.0e0	0	*22e+1*	*13e+1*	*27e+1*	1.0e0
1	0	*29e+0*	*11e+0*	*89e+0*	1.0e0	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f109 in 5-D**, N=15, mFE=500					**f109 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	2	3.3e3	2.9e3	3.8e3	2.7e2	0	*15e+1*	*11e+1*	*21e+1*	7.1e2
1	0	*30e+0*	*88e−1*	*80e+0*	7.9e1	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f110 in 5-D**, N=15, mFE=500					**f110 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*57e+3*	*38e+2*	*26e+4*	1.0e0	0	*44e+4*	*34e+4*	*81e+4*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f111 in 5-D**, N=15, mFE=500					**f111 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*11e+4*	*67e+2*	*26e+4*	1.0e0	0	*34e+4*	*15e+4*	*62e+4*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f112 in 5-D**, N=15, mFE=500					**f112 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*19e+2*	*12e+1*	*18e+3*	1.0e2	0	*37e+4*	*92e+3*	*57e+4*	5.0e1
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f113 in 5-D**, N=15, mFE=500					**f113 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*34e+1*	*12e+1*	*78e+1*	1.0e0	0	*19e+2*	*94e+1*	*56e+2*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f114 in 5-D**, N=15, mFE=500					**f114 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*29e+1*	*10e+1*	*14e+2*	1.0e0	0	*24e+2*	*14e+2*	*37e+2*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f115 in 5-D**, N=15, mFE=500					**f115 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	2	3.7e3	3.6e3	3.8e3	5.0e2	0	*11e+2*	*73e+1*	*25e+2*	5.6e2
1	0	*42e+0*	*91e−1*	*15e+1*	3.2e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f116 in 5-D**, N=15, mFE=500					**f116 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*89e+3*	*17e+3*	*59e+4*	1.0e0	0	*14e+4*	*59e+3*	*28e+4*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f117 in 5-D**, N=15, mFE=500					**f117 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*80e+3*	*17e+3*	*38e+4*	1.0e0	0	*16e+4*	*85e+3*	*34e+4*	1.0e0
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f118 in 5-D**, N=15, mFE=500					**f118 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	0	*94e+2*	*90e+1*	*59e+3*	2.0e2	0	*83e+3*	*29e+3*	*32e+4*	6.3e2
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f119 in 5-D**, N=15, mFE=500					**f119 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	4	1.4e3	1.2e3	1.7e3	3.8e2	0	*12e+1*	*78e+0*	*20e+1*	1.0e0
1	0	*14e+0*	*28e−1*	*69e+0*	1.9e1	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f120 in 5-D**, N=15, mFE=500					**f120 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$	#	ERT	10%	90%	$\mathrm{RT}_{\mathrm{succ}}$
10	2	3.3e3	2.8e3	3.8e3	5.0e2	0	*11e+1*	*81e+0*	*31e+1*	1.0e0
1	0	*35e+0*	*81e−1*	*20e+1*	1.0e0	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: 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 $f_{\mathrm{opt}} + \Delta f$ (ERT, see Figure 2); 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 $f_{\mathrm{opt}} + \Delta f$ 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 2 for the names of functions.

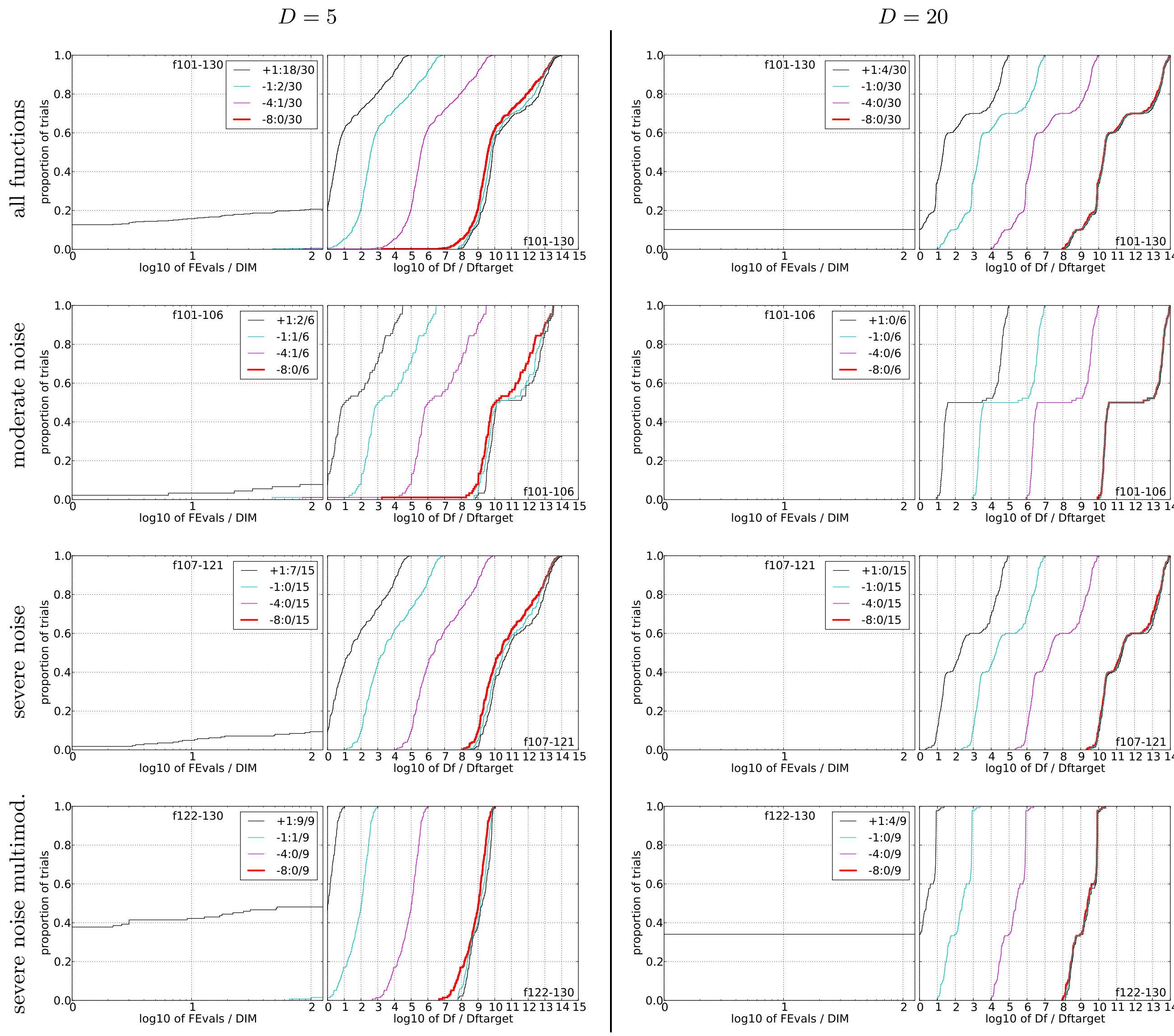

Figure 3: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, 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.

Δf	**f121 in 5-D**, N=15, mFE=500					**f121 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	8	5.0e2	3.6e2	6.5e2	3.0e2	0	*77e+0*	*25e+0*	*16e+1*	2.5e2
1	0	*81e−1*	*21e−1*	*28e+1*	1.6e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f122 in 5-D**, N=15, mFE=500					**f122 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	5	1.0e3	8.1e2	1.2e3	3.1e2	0	*31e+0*	*14e+0*	*62e+0*	1.0e0
1	0	*23e+0*	*58e−1*	*67e+0*	1.0e0	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f123 in 5-D**, N=15, mFE=500					**f123 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	2	3.3e3	2.8e3	3.8e3	2.5e2	1	2.8e4	2.6e4	3.0e4	2.0e3
1	0	*18e+0*	*80e−1*	*80e+0*	1.0e0	0	*27e+0*	*16e+0*	*85e+0*	1.0e0
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f124 in 5-D**, N=15, mFE=500					**f124 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	7	6.2e2	4.4e2	7.9e2	2.5e2	0	*17e+0*	*12e+0*	*30e+0*	4.5e2
1	0	*11e+0*	*44e−1*	*30e+0*	8.9e1	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f125 in 5-D**, N=15, mFE=500					**f125 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	11	3.8e2	2.8e2	4.7e2	2.7e2	0	*29e−1*	*15e−1*	*41e−1*	1.0e0
1e−1	2	3.7e3	3.6e3	3.8e3	5.0e2	.	.	.	.	.
1e−3	0	*52e−2*	*78e−3*	*25e−1*	2.8e2	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f126 in 5-D**, N=15, mFE=500					**f126 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	1	7.0e3	6.5e3	7.5e3	5.0e2	0	*36e−1*	*26e−1*	*51e−1*	1.0e0
1e−1	0	*28e−1*	*12e−1*	*46e−1*	1.0e0	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f127 in 5-D**, N=15, mFE=500					**f127 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	15	1.0e0	1.0e0	1.0e0	1.0e0	15	1.0e0	1.0e0	1.0e0	1.0e0
1	11	2.8e2	2.0e2	3.7e2	2.2e2	1	2.9e4	2.8e4	3.0e4	2.0e3
1e−1	0	*70e−2*	*23e−2*	*43e−1*	2.5e2	0	*15e−1*	*10e−1*	*25e−1*	1.3e3
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f128 in 5-D**, N=15, mFE=500					**f128 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	1	7.3e3	7.0e3	7.5e3	5.0e2	0	*83e+0*	*75e+0*	*85e+0*	1.9e1
1	0	*31e+0*	*11e+0*	*40e+0*	1.6e2	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f129 in 5-D**, N=15, mFE=500					**f129 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	3	2.2e3	2.0e3	2.4e3	3.6e2	0	*82e+0*	*79e+0*	*85e+0*	1.9e1
1	0	*19e+0*	*56e−1*	*45e+0*	8.9e1	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f130 in 5-D**, N=15, mFE=500					**f130 in 20-D**, N=15, mFE=2000				
	#	ERT	10%	90%	RT_{succ}	#	ERT	10%	90%	RT_{succ}
10	2	3.3e3	2.9e3	3.8e3	3.1e2	0	*79e+0*	*71e+0*	*85e+0*	2.8e2
1	0	*14e+0*	*84e−1*	*27e+0*	7.9e1	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 2: 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 $f_{\mathrm{opt}} + \Delta f$ (ERT, see Figure 2); 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 $f_{\mathrm{opt}} + \Delta f$ 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 2 for the names of functions.