Benchmarking the BFGS Algorithm on the BBOB-2009 Function Testbed

Raymond Ros
Univ. Paris-Sud, LRI
UMR 8623 / INRIA Saclay, projet TAO
F-91405 Orsay, France
raymond.ros@lri.fr

ABSTRACT

The BFGS quasi-Newton method is benchmarked on the noiseless BBOB-2009 testbed. A multistart strategy is applied with a maximum number of function evaluations of 10^5 times the search space dimension, resulting in the algorithm solving six functions.

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, BFGS, Quasi-newton

1. INTRODUCTION

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method [1, 3, 4, 7] is a real-parameter unconstrained non-linear optimization method. The BFGS method belongs to the class of quasi-Newton methods which, by supposing that the objective function can be locally approximated by a quadratic function near its optimum, tries to approximate the Hessian matrix of this quadratic function. The BFGS method is tested here on a testbed of noiseless functions.

2. ALGORITHM PRESENTATION

Quasi-Newton methods address the problem of unconstrained black-box optimization by the determination of the stationary point of a function using a second-order approximation. The Hessian matrix is iteratively approximated by finding a search direction using the gradient of the current point and operating a line search to find the step size. For each iteration $D + 1$ function evaluations (where D denotes the problem dimension) or double if the step size changes, are done to evaluate the gradient and the function value at the current point. A simple stochastic independent restart procedure (as advised in [5]) was added.

3. EXPERIMENTAL PROCEDURE

The Matlab implementation of the BFGS method was used. It is accessible using the generic function `fminunc` (revision 1.1.6.3) that proposes, among others, the BFGS method for the update of the Hessian Matrix. The starting point is chosen uniformly in $[-5, 5]^D$. The stopping criteria were chosen such that a restart occurs due to numerical errors. The multistart strategy was used with at most 100 restarts to reduce the duration of an experiment. For the same reason, a run is limited to at most $10^5 \times D$ function evaluations. The algorithm used is presented in Figure 1. No parameter tuning was done, the CrE [5] is computed to zero.

4. RESULTS AND DISCUSSION

Results from experiments according to [5] on the benchmark functions given in [2, 6] are presented in Figures 2 and 3 and in Table 1. As expected, poor results were obtained on multimodal functions f_3, f_4, f_{15-19}, f_{24}. The algorithm could not solve the rugged functions f_7, f_{13}, f_{23} neither. It only solves f_6, in dimensions lesser than 5. Though the functions f_{21} and f_{22} are multimodal and have weak global structure, the algorithm still solves them even for dimensions larger than 10. The algorithm solves f_1 and f_5 with a scaling of its performances close to linear. The scaling is close to quadratic for f_2, f_8 and f_9. The algorithm is coordinate-dependent as a result of the computation of the gradient using finite differences which can yield numerical errors on ill-conditioned functions as can be seen in the results on f_{10} compared to f_9. The effects of ill-conditioning on the performances of the algorithm can also be seen in the results on f_{11} and f_{14}.

5. CPU TIMING EXPERIMENT

For the timing experiment, the proposed algorithm was run on f_8 and restarted until at least 30 seconds have passed (according to Figure 2 in [5]). The experiments were conducted with an Intel Core 2 6700 processor (2.66GHz) with Matlab R2008a on Linux 2.6.24.7. The results were 6.0, 4.7, 3.7, 3.0, 2.9, 2.9 and 2.8 $\times 10^{-4}$ seconds per function evaluations in dimension 2, 3, 5, 10, 20, 40 and 80 respectively.

Δf	**f_1 in 5-D**, N=15, mFE=18					**f_1 in 20-D**, N=15, mFE=63				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e1	1.3e1	1.3e1	1.3e1	15	4.3e1	4.3e1	4.3e1	4.3e1
1	15	1.3e1	1.3e1	1.3e1	1.3e1	15	4.3e1	4.3e1	4.3e1	4.3e1
1e−1	15	1.3e1	1.3e1	1.3e1	1.3e1	15	4.3e1	4.3e1	4.3e1	4.3e1
1e−3	15	1.3e1	1.3e1	1.3e1	1.3e1	15	4.3e1	4.3e1	4.3e1	4.3e1
1e−5	15	1.3e1	1.3e1	1.3e1	1.3e1	15	4.3e1	4.3e1	4.3e1	4.3e1
1e−8	15	1.3e1	1.3e1	1.3e1	1.3e1	15	4.3e1	4.3e1	4.3e1	4.3e1

Δf	**f_2 in 5-D**, N=15, mFE=948					**f_2 in 20-D**, N=15, mFE=12516				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.2e2	2.8e2	3.7e2	3.2e2	15	7.5e3	7.2e3	7.9e3	7.5e3
1	15	4.9e2	4.5e2	5.4e2	4.9e2	15	9.2e3	8.7e3	9.7e3	9.2e3
1e−1	15	5.5e2	5.1e2	5.9e2	5.5e2	15	1.0e4	9.5e3	1.0e4	1.0e4
1e−3	15	5.9e2	5.6e2	6.3e2	5.9e2	15	1.1e4	1.0e4	1.1e4	1.1e4
1e−5	15	6.4e2	6.0e2	6.8e2	6.4e2	15	1.1e4	1.0e4	1.1e4	1.1e4
1e−8	15	6.9e2	6.4e2	7.3e2	6.9e2	15	1.1e4	1.1e4	1.1e4	1.1e4

Δf	**f_3 in 5-D**, N=15, mFE=18540					**f_3 in 20-D**, N=15, mFE=129192				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	3	7.6e4	4.6e4	2.2e5	1.8e4	0	*28e+1*	*18e+1*	*32e+1*	5.6e4
1	0	*21e+0*	*70e−1*	*24e+0*	7.1e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_4 in 5-D**, N=15, mFE=20964					**f_4 in 20-D**, N=15, mFE=169596				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	2	1.4e5	7.2e4	>3e5	2.0e4	0	*40e+1*	*29e+1*	*45e+1*	6.3e4
1	0	*24e+0*	*99e−1*	*34e+0*	1.0e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_5 in 5-D**, N=15, mFE=48					**f_5 in 20-D**, N=15, mFE=189				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.9e1	1.8e1	2.0e1	1.9e1	15	9.6e1	9.1e1	1.0e2	9.6e1
1	15	3.0e1	2.7e1	3.2e1	3.0e1	15	1.1e2	1.1e2	1.2e2	1.1e2
1e−1	15	3.1e1	2.9e1	3.3e1	3.1e1	15	1.1e2	1.1e2	1.2e2	1.1e2
1e−3	15	3.1e1	2.9e1	3.3e1	3.1e1	15	1.1e2	1.1e2	1.2e2	1.1e2
1e−5	15	3.1e1	2.9e1	3.3e1	3.1e1	15	1.1e2	1.1e2	1.2e2	1.1e2
1e−8	15	3.1e1	2.9e1	3.3e1	3.1e1	15	1.1e2	1.1e2	1.2e2	1.1e2

Δf	**f_6 in 5-D**, N=15, mFE=155790					**f_6 in 20-D**, N=15, mFE=2.00e6				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.5e2	2.8e2	4.1e2	3.5e2	15	4.6e3	4.1e3	5.1e3	4.6e3
1	15	7.1e2	6.3e2	7.9e2	7.1e2	15	8.1e3	7.4e3	8.9e3	8.1e3
1e−1	15	9.5e2	8.6e2	1.0e3	9.5e2	15	1.2e4	1.1e4	1.3e4	1.2e4
1e−3	15	1.4e3	1.3e3	1.6e3	1.4e3	15	1.8e4	1.7e4	2.0e4	1.8e4
1e−5	15	2.1e3	1.9e3	2.3e3	2.1e3	15	2.4e4	2.3e4	2.6e4	2.4e4
1e−8	15	7.1e4	5.5e4	8.7e4	7.1e4	0	*44e−9*	*31e−9*	*77e−9*	8.9e5

Δf	**f_7 in 5-D**, N=15, mFE=600					**f_7 in 20-D**, N=15, mFE=2100				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	0	*32e+0*	*13e+0*	*58e+0*	3.2e2	0	*67e+1*	*56e+1*	*10e+2*	1.3e3
1	.	.	.	.	.	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_8 in 5-D**, N=15, mFE=1968					**f_8 in 20-D**, N=15, mFE=6531				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.5e2	1.3e2	1.8e2	1.5e2	15	3.7e3	3.5e3	3.8e3	3.7e3
1	15	4.9e2	3.7e2	6.3e2	4.9e2	15	4.7e3	4.6e3	4.9e3	4.7e3
1e−1	15	5.5e2	4.3e2	6.8e2	5.5e2	15	5.0e3	4.8e3	5.1e3	5.0e3
1e−3	15	5.9e2	4.7e2	7.3e2	5.9e2	15	5.1e3	5.0e3	5.3e3	5.1e3
1e−5	15	6.0e2	4.9e2	7.4e2	6.0e2	15	5.2e3	5.0e3	5.3e3	5.2e3
1e−8	15	6.2e2	4.9e2	7.5e2	6.2e2	15	5.2e3	5.1e3	5.4e3	5.2e3

Δf	**f_9 in 5-D**, N=15, mFE=936					**f_9 in 20-D**, N=15, mFE=14028				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e2	1.0e2	1.5e2	1.3e2	15	3.8e3	3.6e3	3.9e3	3.8e3
1	15	3.8e2	3.1e2	4.5e2	3.8e2	15	6.7e3	5.7e3	7.7e3	6.7e3
1e−1	15	4.4e2	3.7e2	5.1e2	4.4e2	15	6.9e3	5.9e3	7.9e3	6.9e3
1e−3	15	4.8e2	4.2e2	5.5e2	4.8e2	15	7.1e3	6.1e3	8.1e3	7.1e3
1e−5	15	5.0e2	4.3e2	5.7e2	5.0e2	15	7.1e3	6.1e3	8.2e3	7.1e3
1e−8	15	5.1e2	4.4e2	5.8e2	5.1e2	15	7.2e3	6.2e3	8.2e3	7.2e3

Δf	**f_{10} in 5-D**, N=15, mFE=78708					**f_{10} in 20-D**, N=15, mFE=1.11e6				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.5e2	3.1e2	3.9e2	3.5e2	15	7.5e3	7.1e3	7.8e3	7.5e3
1	15	5.0e2	4.6e2	5.4e2	5.0e2	15	8.7e3	8.3e3	9.0e3	8.7e3
1e−1	15	5.7e2	5.3e2	6.2e2	5.7e2	15	1.1e4	9.6e3	1.2e4	1.1e4
1e−3	15	6.3e2	5.8e2	6.7e2	6.3e2	15	1.6e4	1.4e4	1.8e4	1.6e4
1e−5	15	9.0e2	6.9e2	1.1e3	9.0e2	15	5.2e4	2.5e4	8.2e4	5.2e4
1e−8	5	2.0e5	1.3e5	3.6e5	6.1e4	0	*77e−8*	*19e−8*	*16e−7*	5.6e5

Δf	**f_{11} in 5-D**, N=15, mFE=49422					**f_{11} in 20-D**, N=15, mFE=211239				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.4e2	1.3e2	1.5e2	1.4e2	15	1.0e3	8.5e2	1.2e3	1.0e3
1	15	2.0e2	1.8e2	2.3e2	2.0e2	15	2.2e3	1.8e3	2.7e3	2.2e3
1e−1	15	8.4e2	6.4e2	1.1e3	8.4e2	15	8.3e3	4.7e3	1.2e4	8.3e3
1e−3	14	9.6e3	6.2e3	1.3e4	9.6e3	2	1.4e6	7.3e5	>3e6	2.0e5
1e−5	2	2.9e5	1.5e5	>6e5	4.5e4	0	*31e−4*	*83e−5*	*47e−4*	7.9e4
1e−8	0	*32e−6*	*80e−7*	*33e−5*	2.8e4	.	.	.	.	.

Δf	**f_{12} in 5-D**, N=15, mFE=97074					**f_{12} in 20-D**, N=15, mFE=612423				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.2e2	9.4e1	1.5e2	1.2e2	15	1.6e3	1.1e3	2.2e3	1.6e3
1	15	2.7e2	2.3e2	3.1e2	2.7e2	15	3.2e3	2.4e3	4.0e3	3.2e3
1e−1	15	3.7e2	3.1e2	4.3e2	3.7e2	15	4.5e3	3.7e3	5.3e3	4.5e3
1e−3	15	4.6e2	3.8e2	5.4e2	4.6e2	15	6.5e3	4.8e3	8.3e3	6.5e3
1e−5	15	2.5e3	1.6e3	3.6e3	2.5e3	15	2.2e4	1.6e4	2.9e4	2.2e4
1e−8	5	1.4e5	8.5e4	2.9e5	3.7e4	1	7.9e6	3.7e6	>8e6	1.2e5

Δf	**f_{13} in 5-D**, N=15, mFE=55584					**f_{13} in 20-D**, N=15, mFE=580545				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.3e2	1.2e2	1.4e2	1.3e2	15	1.1e3	1.0e3	1.1e3	1.1e3
1	15	1.9e2	1.9e2	2.0e2	1.9e2	15	2.0e3	2.0e3	2.0e3	2.0e3
1e−1	15	2.5e2	2.5e2	2.5e2	2.5e2	15	2.8e3	2.7e3	2.8e3	2.8e3
1e−3	14	6.2e3	1.7e3	1.1e4	6.2e3	9	4.2e5	2.8e5	6.5e5	2.4e5
1e−5	3	2.4e5	1.4e5	7.2e5	4.4e4	0	*96e−5*	*20e−6*	*19e−4*	1.8e5
1e−8	0	*37e−6*	*46e−7*	*79e−5*	2.8e4	.	.	.	.	.

Δf	**f_{14} in 5-D**, N=15, mFE=37158					**f_{14} in 20-D**, N=15, mFE=257859				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.2e1	1.7e1	2.7e1	2.2e1	15	2.0e2	1.8e2	2.3e2	2.0e2
1	15	7.0e1	5.7e1	8.4e1	7.0e1	15	4.4e2	4.0e2	4.7e2	4.4e2
1e−1	15	1.0e2	8.7e1	1.2e2	1.0e2	15	6.1e2	5.6e2	6.6e2	6.1e2
1e−3	15	1.7e2	1.5e2	2.0e2	1.7e2	15	1.1e3	1.0e3	1.2e3	1.1e3
1e−5	15	2.5e2	2.3e2	2.7e2	2.5e2	15	1.9e3	1.8e3	2.0e3	1.9e3
1e−8	0	*16e−8*	*75e−9*	*38e−8*	1.6e4	0	*18e−7*	*14e−7*	*32e−7*	1.0e5

Δf	**f_{15} in 5-D**, N=15, mFE=18486					**f_{15} in 20-D**, N=15, mFE=128646				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	5	4.4e4	2.9e4	7.9e4	1.4e4	0	*25e+1*	*18e+1*	*32e+1*	7.1e4
1	0	*13e+0*	*70e−1*	*22e+0*	1.1e4	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{16} in 5-D**, N=15, mFE=43056					**f_{16} in 20-D**, N=15, mFE=334509				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.8e4	1.5e4	2.2e4	1.8e4	0	*26e+0*	*21e+0*	*37e+0*	1.8e5
1	1	5.9e5	2.8e5	>6e5	4.1e4	.	.	.	.	.
1e−1	0	*49e−1*	*23e−1*	*89e−1*	2.5e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{17} in 5-D**, N=15, mFE=22386					**f_{17} in 20-D**, N=15, mFE=410025				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	6.2e2	3.7e2	8.9e2	6.2e2	15	2.3e4	1.3e4	3.3e4	2.3e4
1	2	1.4e5	7.0e4	>3e5	1.8e4	0	*56e−1*	*46e−1*	*68e−1*	2.5e5
1e−1	0	*19e−1*	*87e−2*	*28e−1*	1.3e4	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{18} in 5-D**, N=15, mFE=22008					**f_{18} in 20-D**, N=15, mFE=426468				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	14	5.9e3	3.9e3	8.0e3	5.6e3	0	*21e+0*	*18e+0*	*26e+0*	2.0e5
1	0	*51e−1*	*28e−1*	*10e+0*	8.9e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{19} in 5-D**, N=15, mFE=37734					**f_{19} in 20-D**, N=15, mFE=255570				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.7e3	1.2e3	2.2e3	1.7e3	3	1.2e6	7.0e5	3.5e6	2.5e5
1	12	2.2e4	1.7e4	2.8e4	1.7e4	0	*12e+0*	*71e−1*	*15e+0*	1.6e5
1e−1	1	4.3e5	2.1e5	>4e5	2.8e4	.	.	.	.	.
1e−3	0	*62e−2*	*22e−2*	*12e−1*	1.8e4	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{20} in 5-D**, N=15, mFE=28008					**f_{20} in 20-D**, N=15, mFE=420504				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.8e1	2.5e1	3.2e1	2.8e1	15	1.7e2	1.6e2	1.8e2	1.7e2
1	15	2.2e3	1.8e3	2.5e3	2.2e3	13	2.7e5	2.1e5	3.4e5	2.2e5
1e−1	1	3.9e5	1.9e5	>4e5	2.7e4	0	*90e−2*	*82e−2*	*10e−1*	2.8e5
1e−3	1	3.9e5	1.9e5	>4e5	2.7e4	.	.	.	.	.
1e−5	1	3.9e5	1.9e5	>4e5	2.7e4	.	.	.	.	.
1e−8	1	3.9e5	1.9e5	>4e5	2.7e4	.	.	.	.	.

Δf	**f_{21} in 5-D**, N=15, mFE=12012					**f_{21} in 20-D**, N=15, mFE=140112				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	1.6e2	9.4e1	2.3e2	1.6e2	15	1.1e3	6.4e2	1.6e3	1.1e3
1	15	1.6e3	1.1e3	2.1e3	1.6e3	15	3.6e4	2.4e4	4.7e4	3.6e4
1e−1	15	3.2e3	2.1e3	4.3e3	3.2e3	13	6.5e4	4.8e4	8.6e4	5.3e4
1e−3	15	3.3e3	2.2e3	4.3e3	3.3e3	13	6.6e4	4.9e4	8.7e4	5.4e4
1e−5	15	3.4e3	2.3e3	4.4e3	3.4e3	13	6.7e4	5.0e4	8.7e4	5.5e4
1e−8	15	3.7e3	2.5e3	4.9e3	3.7e3	2	8.6e5	4.3e5	>2e6	9.1e4

Δf	**f_{22} in 5-D**, N=15, mFE=17538					**f_{22} in 20-D**, N=15, mFE=164409				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	2.2e2	1.6e2	2.8e2	2.2e2	15	1.2e3	8.2e2	1.6e3	1.2e3
1	15	1.1e3	7.9e2	1.5e3	1.1e3	15	1.0e4	5.7e3	1.5e4	1.0e4
1e−1	15	2.0e3	1.5e3	2.4e3	2.0e3	7	1.9e5	1.2e5	3.2e5	7.2e4
1e−3	15	2.1e3	1.6e3	2.5e3	2.1e3	7	1.9e5	1.3e5	3.2e5	7.3e4
1e−5	15	2.1e3	1.7e3	2.6e3	2.1e3	6	2.6e5	1.7e5	4.5e5	9.0e4
1e−8	14	6.3e3	4.3e3	8.4e3	6.3e3	0	*69e−2*	*78e−8*	*69e−2*	2.5e4

Δf	**f_{23} in 5-D**, N=15, mFE=26718					**f_{23} in 20-D**, N=15, mFE=115269				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	15	3.2e1	1.9e1	4.6e1	3.2e1	15	1.5e2	7.9e1	2.2e2	1.5e2
1	12	1.6e4	1.3e4	2.0e4	1.4e4	3	4.9e5	3.0e5	1.5e6	1.1e5
1e−1	0	*69e−2*	*31e−2*	*13e−1*	1.4e4	0	*13e−1*	*85e−2*	*23e−1*	5.6e4
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	**f_{24} in 5-D**, N=15, mFE=16734					**f_{24} in 20-D**, N=15, mFE=122619				
	#	ERT	10%	90%	$\mathrm{RT_{succ}}$	#	ERT	10%	90%	$\mathrm{RT_{succ}}$
10	2	1.1e5	5.9e4	>2e5	1.6e4	0	*31e+1*	*27e+1*	*37e+1*	7.1e4
1	0	*17e+0*	*81e−1*	*20e+0*	5.0e3	.	.	.	.	.
1e−1	.	.	.	.	.	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Table 1: Shown are, 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 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 $f_{\rm opt}+\Delta f$ 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 2 for the names of functions.

Figure 1: Matlab code: Multistart BFGS

```
function [x, ilaunch] = MY_OPTIMIZER(FUN, D, ftarget, maxfunevals)
% minimizes FUN in D dimensions by independent restarts of fminunc (BFGS).
% ftarget and maxfunevals are additional external termination conditions.
% Search space is [-5, 5]^D

  % set options, make sure we always terminate
  options = optimset('fminunc');
  options = optimset(options, 'LargeScale', 'off'); % BFGS algorithm
  options = optimset(options, 'MaxIter', inf, 'Tolfun', 1e-11, 'TolX', 0, ...
                     'OutputFcn', @callback, 'Display', 'off');

  maxfunevals = min(1e4*DIM, maxfunevals);

  % multistart such that ftarget is reached with reasonable prob.
  for ilaunch = 1:100;  % relaunch optimizer up to 100 times
    options = optimset(options, 'MaxFunEvals', ...
                       maxfunevals - feval(FUN, 'evaluations'));
    x = fminunc(FUN, 10*rand(DIM,1)-5, options);
    if (feval(FUN, 'fbest') < ftarget || ...
        feval(FUN, 'evaluations') >= maxfunevals)
      break;
    end
  end

  function stop = callback(x, optimValues, state)
    stop = false;
    if optimValues.fval < ftarget
      stop = true;
    end
  end
end % function
```

6. CONCLUSION

The results of a quasi-Newton method algorithm with restarts were presented. The algorithm performs well on smooth functions, but its performances decrease on rugged and multimodal functions. Furthermore due to numerical error in the computation of the gradient in ill-conditioned problems, the algorithm is affected by coordinate systems transformations.

Acknowledgments

The first author would like to acknowledge the support, help, and work of the BBOB team with particular kudos to Anne Auger, Steffen Finck and Nikolaus Hansen.

7. REFERENCES

[1] C. G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, 6:76–90, 1970.

[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.

[3] R. Fletcher. A new approach to variable metric algorithms. *Computer journal*, 13:317–322, 1970.

[4] D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24:23–26, 1970.

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

[6] 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.

[7] D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970.

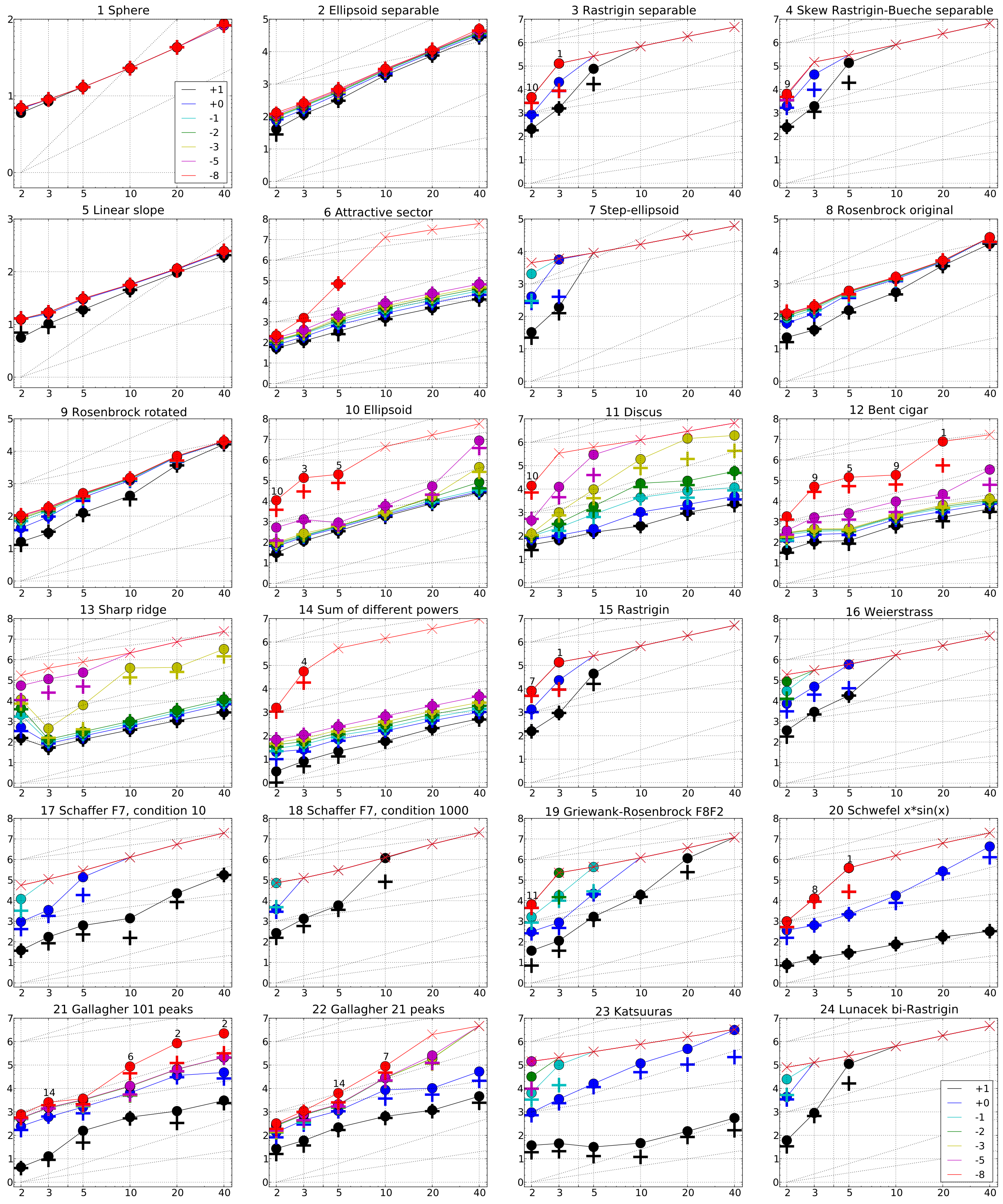

Figure 2: Expected Running Time (ERT, ●) to reach $f_{\text{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation. The $\text{ERT}(\Delta f)$ equals to $\#\text{FEs}(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\text{opt}} + \Delta f$ was surpassed during the trial. The $\#\text{FEs}(\Delta f)$ are the total number of function evaluations while $f_{\text{opt}} + \Delta f$ was not surpassed during the trial from all respective trials (successful and unsuccessful), and f_{opt} denotes the optimal function value. Crosses ($\times$) indicate the total number of function evaluations $\#\text{FEs}(-\infty)$. 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.

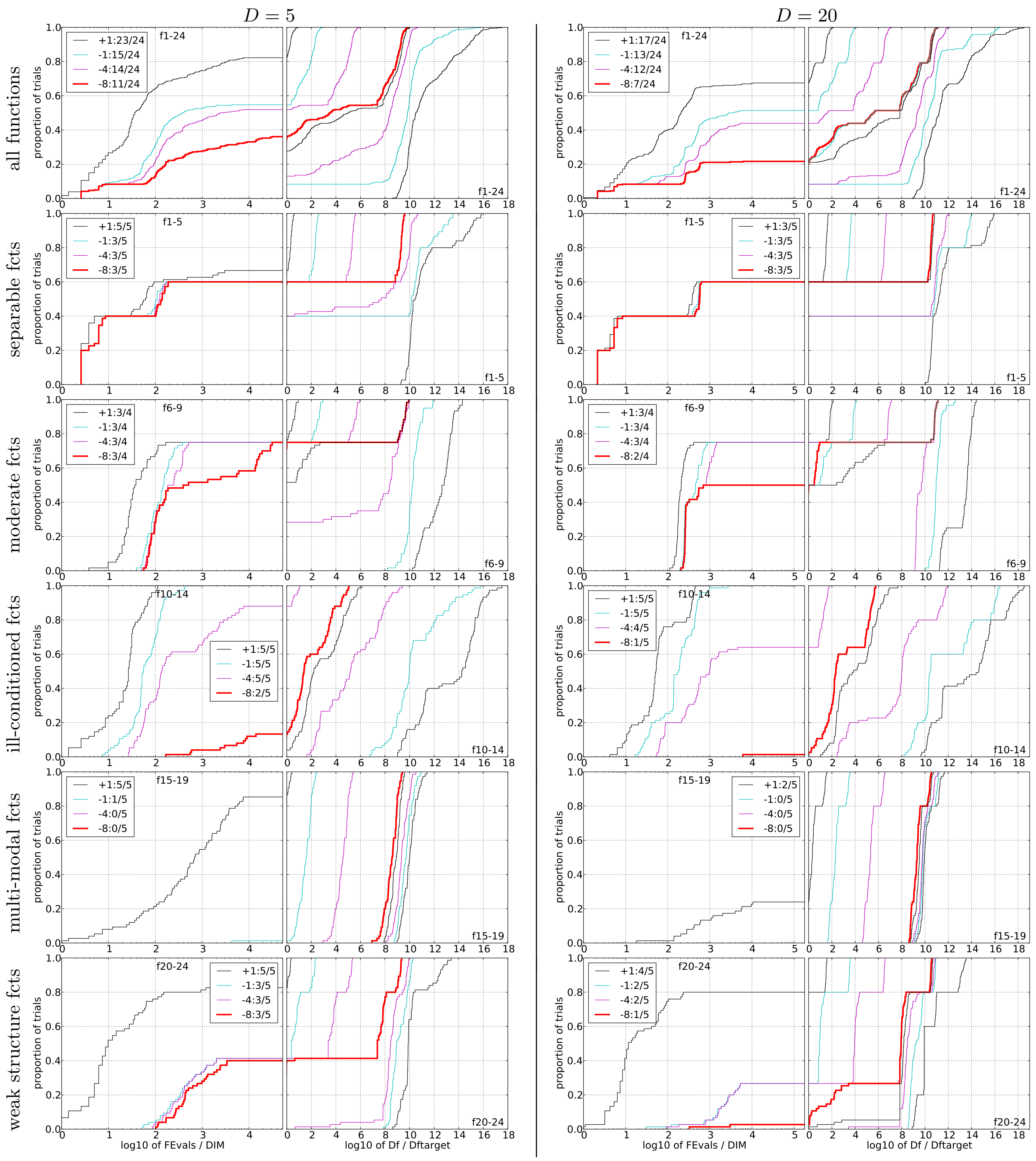

Figure 3: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\text{opt}} + \Delta f$ 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: separable functions; third row: misc. moderate functions; fourth row: ill-conditioned functions; fifth row: multi-modal functions with adequate structure; last row: multi-modal functions with weak structure. 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.