Black-Box Optimization Benchmarking for Noiseless Function Testbed using Particle Swarm Optimization

Mohammed El-Abd
University of Waterloo
Waterloo, Ontario, Canada
mhelabd@pami.uwaterloo.ca

Mohamed S. Kamel
University of Waterloo
Waterloo, Ontario, Canada
mkamel@pami.uwaterloo.ca

ABSTRACT

This paper benchmarks the particle swarm optimizer (PSO) algorithm using the noise-free BBOB 2009 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, Particle Swarm Optimization

1. INTRODUCTION

Particle Swarm Optimization (PSO) [1, 5] is an optimization method widely used to solve continuous nonlinear functions. It is a stochastic optimization technique that emerged from simulations of the birds flocking and fish schooling behaviors.

The algorithm used is a simple PSO algorithm utilizing the *gbest* (global best) model.

2. ALGORITHM PRESENTATION

The only design choice made was to select the absorbing boundaries to handle any particles leaving the search space, where the position is set to the boundary and the velocity is reset to zeros. Fig. 1 shows the MATLAB code for PSO algorithm.

The simulations for 2; 5; 10 and 20 D were done with the MATLAB-code and took 12 hours and 19 minutes. No parameter tuning was done and the crafting effort CrE [4] is computed to zero.

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

3. RESULTS

The swarm has 40 particles with the parameters set as $c1 = c2 = 1.4944$ and $w = 0.792$.

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

4. CPU TIMING EXPERIMENT

For the timing experiment, PSO was run with a maximum of 10^4 function evaluations and restarted until 30 seconds has passed (according to Figure 2 in [4]). The experiments have been conducted with an Intel Core 2 Quad 2.4 GHz under Windows XP using the MATLAB-code provided. The time per function evaluation was 1.1; 1.3; 1.4; 1.5; 1.8 times 10^{-5} seconds in dimensions 2; 3; 5; 10; 20 respectively.

5. REFERENCES

[1] R. C. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proc. of the 6th International Symposium on Micro Machine and Human Science*, pages 39–43, 1995.

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

[4] 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.

[5] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. of IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995.

Δf	f_1 in 5-D, N=15, mFE=7880 #	ERT	10%	90%	RT_{succ}	f_1 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	4.1e1	3.5e1	5.0e1	4.1e1	15	9.7e2	8.9e2	1.0e3	9.7e2
1	15	2.7e2	2.5e2	2.8e2	2.7e2	14	1.5e5	3.5e3	4.3e5	1.5e5
1e−1	15	6.8e2	6.2e2	7.1e2	6.8e2	14	1.5e5	5.7e3	1.5e5	1.5e5
1e−3	15	2.2e3	2.1e3	2.3e3	2.2e3	14	1.5e5	1.1e4	3.0e5	1.5e5
1e−5	15	3.9e3	3.8e3	4.0e3	3.9e3	14	1.6e5	1.6e4	3.0e5	1.6e5
1e−8	15	6.6e3	6.5e3	6.7e3	6.6e3	14	1.7e5	2.5e4	3.1e5	1.6e5

Δf	f_2 in 5-D, N=15, mFE=12560 #	ERT	10%	90%	RT_{succ}	f_2 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	2.6e3	2.6e3	2.8e3	2.6e3	8	1.8e6	1.0e6	2.8e6	1.3e6
1	15	3.6e3	3.4e3	3.7e3	3.6e3	8	1.8e6	1.0e6	2.3e6	1.3e6
1e−1	15	4.3e3	4.1e3	4.4e3	4.3e3	8	1.8e6	1.5e6	2.5e6	1.3e6
1e−3	15	6.1e3	5.9e3	6.3e3	6.1e3	8	1.8e6	1.0e6	2.5e6	1.3e6
1e−5	15	8.2e3	8.0e3	8.7e3	8.2e3	8	1.8e6	1.0e6	2.5e6	1.3e6
1e−8	15	1.1e4	1.1e4	1.1e4	1.1e4	8	1.8e6	1.0e6	2.3e6	1.3e6

Δf	f_3 in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_3 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	14	3.7e4	1.9e3	1.1e5	3.7e4	0	*21e+0*	*16e+0*	*56e+0*	5.0e4
1	13	8.9e4	1.3e4	1.6e5	4.9e4	.	.	.	.	.
1e−1	8	4.5e5	2.7e5	5.7e5	2.0e5	.	.	.	.	.
1e−3	8	4.5e5	3.3e5	6.3e5	2.0e5	.	.	.	.	.
1e−5	8	4.5e5	3.3e5	6.4e5	2.0e5	.	.	.	.	.
1e−8	8	4.6e5	3.4e5	7.0e5	2.0e5	.	.	.	.	.

Δf	f_4 in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_4 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	2.4e3	2.3e3	2.6e3	2.4e3	1	2.8e7	2.4e7	3.0e7	2.0e6
1	11	2.3e5	1.3e5	2.8e5	1.8e5	0	*23e+0*	*15e+0*	*36e+0*	5.6e4
1e−1	1	7.0e6	5.5e6	7.5e6	8.1e3	.	.	.	.	.
1e−3	1	7.0e6	6.5e6	7.5e6	1.2e4	.	.	.	.	.
1e−5	1	7.0e6	6.0e6	7.5e6	1.5e4	.	.	.	.	.
1e−8	1	7.0e6	6.1e6	7.5e6	1.8e4	.	.	.	.	.

Δf	f_5 in 5-D, N=15, mFE=320 #	ERT	10%	90%	RT_{succ}	f_5 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	1.0e2	9.7e1	1.1e2	1.0e2	8	1.8e6	1.5e6	2.5e6	1.0e6
1	15	1.5e2	1.3e2	1.6e2	1.5e2	8	1.8e6	1.0e6	2.0e6	1.0e6
1e−1	15	1.6e2	1.5e2	1.7e2	1.6e2	8	1.8e6	1.0e6	2.5e6	1.0e6
1e−3	15	1.6e2	1.4e2	1.8e2	1.6e2	8	1.8e6	7.5e5	2.0e6	1.0e6
1e−5	15	1.6e2	1.5e2	1.8e2	1.6e2	8	1.8e6	1.3e6	2.5e6	1.0e6
1e−8	15	1.6e2	1.6e2	1.7e2	1.6e2	8	1.8e6	1.0e6	2.8e6	1.0e6

Δf	f_6 in 5-D, N=15, mFE=19880 #	ERT	10%	90%	RT_{succ}	f_6 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	5.4e2	4.2e2	6.8e2	5.4e2	9	1.4e6	9.8e5	2.2e6	7.2e5
1	15	1.9e3	1.6e3	2.0e3	1.9e3	7	2.4e6	1.6e6	2.6e6	1.2e6
1e−1	15	3.2e3	3.0e3	3.7e3	3.2e3	7	2.4e6	1.6e6	3.5e6	1.2e6
1e−3	15	6.6e3	6.2e3	7.0e3	6.6e3	7	2.6e6	1.8e6	3.3e6	1.3e6
1e−5	15	1.1e4	1.0e4	1.1e4	1.1e4	7	2.8e6	2.3e6	3.4e6	1.4e6
1e−8	15	1.7e4	1.6e4	1.7e4	1.7e4	5	5.0e6	4.2e6	5.5e6	1.7e6

Δf	f_7 in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_7 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	2.7e2	2.0e2	3.6e2	2.7e2	12	5.8e5	1.3e5	9.0e5	5.7e5
1	15	3.1e3	1.5e3	3.6e3	3.1e3	0	*62e−1*	*29e−1*	*17e+0*	1.1e5
1e−1	7	6.9e5	5.7e5	8.7e5	2.9e5	.	.	.	.	.
1e−3	6	8.5e5	7.3e5	1.1e6	2.7e5	.	.	.	.	.
1e−5	6	8.5e5	8.0e5	1.0e6	2.7e5	.	.	.	.	.
1e−8	6	8.5e5	6.9e5	1.0e6	2.7e5	.	.	.	.	.

Δf	f_8 in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_8 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	9.6e2	8.3e2	1.0e3	9.6e2	15	1.8e5	1.5e5	1.9e5	1.8e5
1	14	4.2e4	5.9e3	7.7e4	4.1e4	11	1.2e6	1.0e6	1.3e6	1.0e6
1e−1	14	6.7e4	3.0e4	1.0e5	6.5e4	11	1.4e6	1.2e6	1.7e6	1.2e6
1e−3	14	1.8e5	1.5e5	2.1e5	1.7e5	11	2.0e6	1.8e6	2.2e6	1.6e6
1e−5	14	3.2e5	2.9e5	3.6e5	3.0e5	7	3.9e6	3.8e6	4.1e6	1.9e6
1e−8	7	1.0e6	1.0e6	1.1e6	4.8e5	1	3.0e7	2.9e7	3.0e7	2.0e6

Δf	f_9 in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_9 in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	8.5e2	6.6e2	1.0e3	8.5e2	14	1.2e6	1.0e6	1.5e6	1.1e6
1	13	1.2e5	4.4e4	2.3e5	8.1e4	0	*75e−1*	*45e−1*	*99e−1*	2.0e6
1e−1	12	1.4e5	3.0e4	2.2e5	1.0e5	.	.	.	.	.
1e−3	11	3.4e5	3.3e5	4.2e5	2.6e5	.	.	.	.	.
1e−5	7	7.9e5	6.7e5	9.0e5	4.0e5	.	.	.	.	.
1e−8	5	1.3e6	1.2e6	1.4e6	4.9e5	.	.	.	.	.

Δf	f_{10} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{10} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	8	6.1e5	5.3e5	7.2e5	3.4e5	0	*84e+1*	*33e+1*	*81e+2*	2.0e6
1	4	1.6e6	1.5e6	1.7e6	4.6e5	.	.	.	.	.
1e−1	0	*10e+0*	*31e−2*	*37e+0*	4.5e5	.	.	.	.	.
1e−3	.	.	.	.	.	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{11} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{11} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	1.3e4	9.1e3	1.4e4	1.3e4	15	1.4e5	1.4e5	1.6e5	1.4e5
1	15	4.8e4	3.9e4	5.2e4	4.8e4	15	4.1e5	4.0e5	4.3e5	4.1e5
1e−1	15	9.4e4	7.7e4	1.1e5	9.4e4	15	6.9e5	6.5e5	7.1e5	6.9e5
1e−3	15	1.9e5	1.7e5	2.1e5	1.9e5	15	1.3e6	1.3e6	1.3e6	1.3e6
1e−5	14	3.6e5	3.3e5	3.7e5	3.4e5	15	1.8e6	1.8e6	1.9e6	1.8e6
1e−8	8	8.8e5	8.7e5	9.1e5	4.8e5	0	*21e−7*	*31e−8*	*87e−7*	2.0e6

Δf	f_{12} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{12} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	13	8.1e4	4.3e3	1.6e5	8.0e4	8	1.8e6	1.3e6	2.5e6	1.0e6
1	5	1.0e6	7.1e5	1.2e6	4.0e5	0	*64e−1*	*19e−1*	*80e+4*	1.8e6
1e−1	3	2.0e6	1.5e6	2.2e6	5.0e5	.	.	.	.	.
1e−3	1	7.0e6	6.5e6	7.5e6	5.0e5	.	.	.	.	.
1e−5	0	*41e−1*	*24e−4*	*36e+0*	4.5e5	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{13} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{13} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	11	2.1e5	1.6e5	2.7e5	1.6e5	5	4.0e6	2.8e6	4.8e6	1.2e6
1	3	2.0e6	1.7e6	2.3e6	5.0e5	2	1.3e7	1.1e7	1.4e7	1.8e4
1e−1	1	7.0e6	6.0e6	7.5e6	5.0e5	1	2.8e7	2.6e7	3.0e7	2.0e6
1e−3	0	*57e−1*	*26e−2*	*21e+0*	2.2e4	1	2.8e7	2.6e7	3.0e7	2.0e6
1e−5	.	.	.	.	.	0	*22e+0*	*49e−2*	*21e+1*	7.1e4
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{14} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{14} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	1.8e1	1.2e1	2.6e1	1.8e1	15	5.0e2	4.2e2	5.6e2	5.0e2
1	15	2.3e2	2.0e2	2.6e2	2.3e2	15	3.0e3	2.8e3	3.2e3	3.0e3
1e−1	15	8.6e2	7.9e2	8.9e2	8.6e2	15	6.0e3	5.8e3	6.2e3	6.0e3
1e−3	15	4.1e3	3.8e3	4.3e3	4.1e3	15	5.0e4	4.7e4	5.6e4	5.0e4
1e−5	15	5.5e4	4.4e4	7.3e4	5.5e4	0	*44e−6*	*40e−6*	*55e−6*	2.0e6
1e−8	0	*90e−8*	*63e−8*	*20e−7*	4.5e5	.	.	.	.	.

Δf	f_{15} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{15} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	8.1e3	2.7e3	1.3e4	8.1e3	0	*49e+0*	*20e+0*	*98e+0*	7.1e4
1	3	2.1e6	1.7e6	2.5e6	5.0e5	.	.	.	.	.
1e−1	1	7.1e6	6.7e6	7.5e6	5.0e5	.	.	.	.	.
1e−3	1	7.1e6	6.7e6	7.5e6	5.0e5	.	.	.	.	.
1e−5	1	7.1e6	6.3e6	7.5e6	5.0e5	.	.	.	.	.
1e−8	1	7.1e6	6.3e6	7.5e6	5.0e5	.	.	.	.	.

Δf	f_{16} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{16} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	2.9e2	2.2e2	3.7e2	2.9e2	14	1.5e5	1.1e4	1.6e5	1.5e5
1	15	3.8e3	2.9e3	4.7e3	3.8e3	0	*47e−1*	*18e−1*	*77e−1*	1.4e6
1e−1	12	1.6e5	1.2e5	2.6e5	1.1e5	.	.	.	.	.
1e−3	6	9.4e5	7.5e5	1.2e6	3.8e5	.	.	.	.	.
1e−5	2	3.5e6	3.3e6	3.8e6	2.6e5	.	.	.	.	.
1e−8	0	*16e−4*	*32e−7*	*39e−2*	1.3e5	.	.	.	.	.

Δf	f_{17} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{17} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	1.7e1	8.5e0	2.3e1	1.7e1	15	2.0e2	1.6e2	2.4e2	2.0e2
1	14	3.6e4	6.8e2	7.2e4	3.6e4	7	2.6e6	1.9e6	3.1e6	1.3e6
1e−1	12	1.3e5	4.5e4	1.7e5	1.3e5	0	*10e−1*	*64e−2*	*29e−1*	1.8e6
1e−3	3	2.0e6	1.5e6	2.3e6	3.4e5	.	.	.	.	.
1e−5	2	3.3e6	2.8e6	3.8e6	5.0e5	.	.	.	.	.
1e−8	1	7.1e6	6.4e6	7.5e6	5.0e5	.	.	.	.	.

Δf	f_{18} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{18} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	2.3e2	1.9e2	2.9e2	2.3e2	14	1.5e5	3.2e3	2.9e5	3.6e3
1	15	2.5e3	2.1e3	2.9e3	2.5e3	0	*29e−1*	*16e−1*	*72e−1*	5.6e5
1e−1	8	4.5e5	2.7e5	5.7e5	3.1e5	.	.	.	.	.
1e−3	0	*53e−3*	*62e−4*	*45e−2*	1.6e5	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{19} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{19} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	3.5e1	2.7e1	4.1e1	3.5e1	15	3.8e2	3.2e2	4.0e2	3.8e2
1	15	3.4e3	2.7e3	3.8e3	3.4e3	0	*32e−1*	*29e−1*	*36e−1*	6.3e5
1e−1	8	5.9e5	4.8e5	7.2e5	2.9e5	.	.	.	.	.
1e−3	1	7.2e6	7.0e6	7.5e6	5.0e5	.	.	.	.	.
1e−5	1	7.3e6	6.8e6	7.5e6	5.0e5	.	.	.	.	.
1e−8	0	*94e−3*	*20e−3*	*20e−2*	4.5e5	.	.	.	.	.

Δf	f_{20} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{20} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	1.4e2	1.2e2	1.6e2	1.4e2	15	1.4e3	1.3e3	1.6e3	1.4e3
1	15	2.6e3	2.5e3	2.9e3	2.6e3	7	2.3e6	1.4e6	2.9e6	8.6e5
1e−1	5	1.0e6	7.2e5	1.2e6	4.0e5	0	*11e−1*	*63e−2*	*13e−1*	4.5e4
1e−3	5	1.0e6	7.3e5	1.3e6	4.0e5	.	.	.	.	.
1e−5	5	1.0e6	8.2e5	1.2e6	4.0e5	.	.	.	.	.
1e−8	5	1.0e6	7.2e5	1.3e6	4.0e5	.	.	.	.	.

Δf	f_{21} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{21} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	8.2e1	6.8e1	9.1e1	8.2e1	10	1.0e6	2.1e3	1.4e6	8.0e5
1	8	4.4e5	3.1e5	5.6e5	3.1e5	1	2.8e7	2.4e7	3.0e7	2.5e3
1e−1	8	4.4e5	3.1e5	6.3e5	3.1e5	1	2.8e7	2.8e7	3.0e7	4.1e3
1e−3	8	4.4e5	2.5e5	5.6e5	3.1e5	1	2.8e7	2.2e7	3.0e7	8.6e3
1e−5	8	4.4e5	2.5e5	6.3e5	3.1e5	1	2.8e7	2.6e7	3.0e7	1.2e4
1e−8	8	4.4e5	2.6e5	6.9e5	3.1e5	1	2.8e7	2.6e7	3.0e7	2.1e4

Δf	f_{22} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{22} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	1.8e2	1.5e2	2.2e2	1.8e2	15	2.3e3	1.3e3	3.2e3	2.3e3
1	12	1.3e5	4.3e4	1.7e5	1.3e5	7	2.3e6	2.0e6	2.6e6	1.4e6
1e−1	8	4.4e5	2.5e5	5.6e5	1.9e5	0	*20e−1*	*69e−2*	*51e−1*	4.0e4
1e−3	8	4.4e5	2.6e5	6.3e5	1.9e5	.	.	.	.	.
1e−5	8	4.5e5	3.2e5	6.3e5	1.9e5	.	.	.	.	.
1e−8	8	4.5e5	2.7e5	6.4e5	2.0e5	.	.	.	.	.

Δf	f_{23} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{23} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	6.7e0	4.5e0	8.4e0	6.7e0	15	7.2e0	6.1e0	8.3e0	7.2e0
1	15	1.0e4	6.7e3	1.2e4	1.0e4	8	2.5e6	1.9e6	2.9e6	1.4e6
1e−1	2	3.5e6	3.3e6	3.8e6	5.0e5	0	*95e−2*	*54e−2*	*13e−1*	5.0e5
1e−3	0	*15e−2*	*81e−3*	*41e−2*	3.2e5	.	.	.	.	.
1e−5	.	.	.	.	.	.	.	.	.	.
1e−8	.	.	.	.	.	.	.	.	.	.

Δf	f_{24} in 5-D, N=15, mFE=500000 #	ERT	10%	90%	RT_{succ}	f_{24} in 20-D, N=15, mFE=2000000 #	ERT	10%	90%	RT_{succ}
10	15	9.2e3	8.2e3	1.2e4	9.2e3	0	*60e+0*	*39e+0*	*91e+0*	7.9e5
1	0	*63e−1*	*55e−1*	*83e−1*	1.6e5	.	.	.	.	.
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 $f_{\mathrm{opt}} + \Delta f$ (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.

```
function PSO(FUN, DIM, ftarget, maxfunevals)

% Set algorithm parameters
popsize = 40;
c1 = 1.4944;
c2 = 1.4944;
w = 0.792;
xbound = 5;
vbound = 5;

% Allocate memory and initialize
xmin = -xbound * ones(1,DIM);
xmax = xbound * ones(1,DIM);
vmin = -vbound * ones(1,DIM);
vmax = vbound * ones(1,DIM);

x = 2 * xbound * rand(popsize,DIM) - xbound;
v = 2 * vbound * rand(popsize,DIM) - vbound;
pbest = x;

% update pbest and gbest
cost_p = feval(FUN, pbest');
[cost,index] = min(cost_p);
gbest = pbest(index,:);

maxfunevals = min(1e5 * DIM, maxfunevals);
maxiterations = ceil(maxfunevals/popsize);

for iter = 2 : maxiterations
    % Update velocity
    v = w*v + c1*rand(popsize,DIM).*(pbest-x) + c2*rand(popsize,DIM).*(repmat(gbest,↙
popsize,1)-x);

    % Clamp veloctiy
    s = v < repmat(vmin,popsize,1);
    v = (1-s).*v + s.*repmat(vmin,popsize,1);
    b = v > repmat(vmax,popsize,1);
    v = (1-b).*v + b.*repmat(vmax,popsize,1);

    % Update position
    x = x + v;

    % Clamp position - Absorbing boundaries
    % Set x to the boundary
    s = x < repmat(xmin,popsize,1);
    x = (1-s).*x + s.*repmat(xmin,popsize,1);
    b = x > repmat(xmax,popsize,1);
    x = (1-b).*x + b.*repmat(xmax,popsize,1);

    % Clamp position - Absorbing boundaries
    % Set v to zero
    b = s | b;
```



```
    v = (1-b).*v + b.*zeros(popsize,DIM);

    % Update pbest and gbest if necessary
    cost_x = feval(FUN, x');
    s = cost_x<cost_p;
    cost_p = (1-s).*cost_p + s.*cost_x;
    s = repmat(s',1,DIM);
    pbest = (1-s).*pbest + s.*x;
    [cost,index] = min(cost_p);
    gbest = pbest(index,:);

    % Exit if target is reached
    if feval(FUN, 'fbest') < ftarget
      break;
    end
end
```

Figure 1: PSO MATLAB-code.

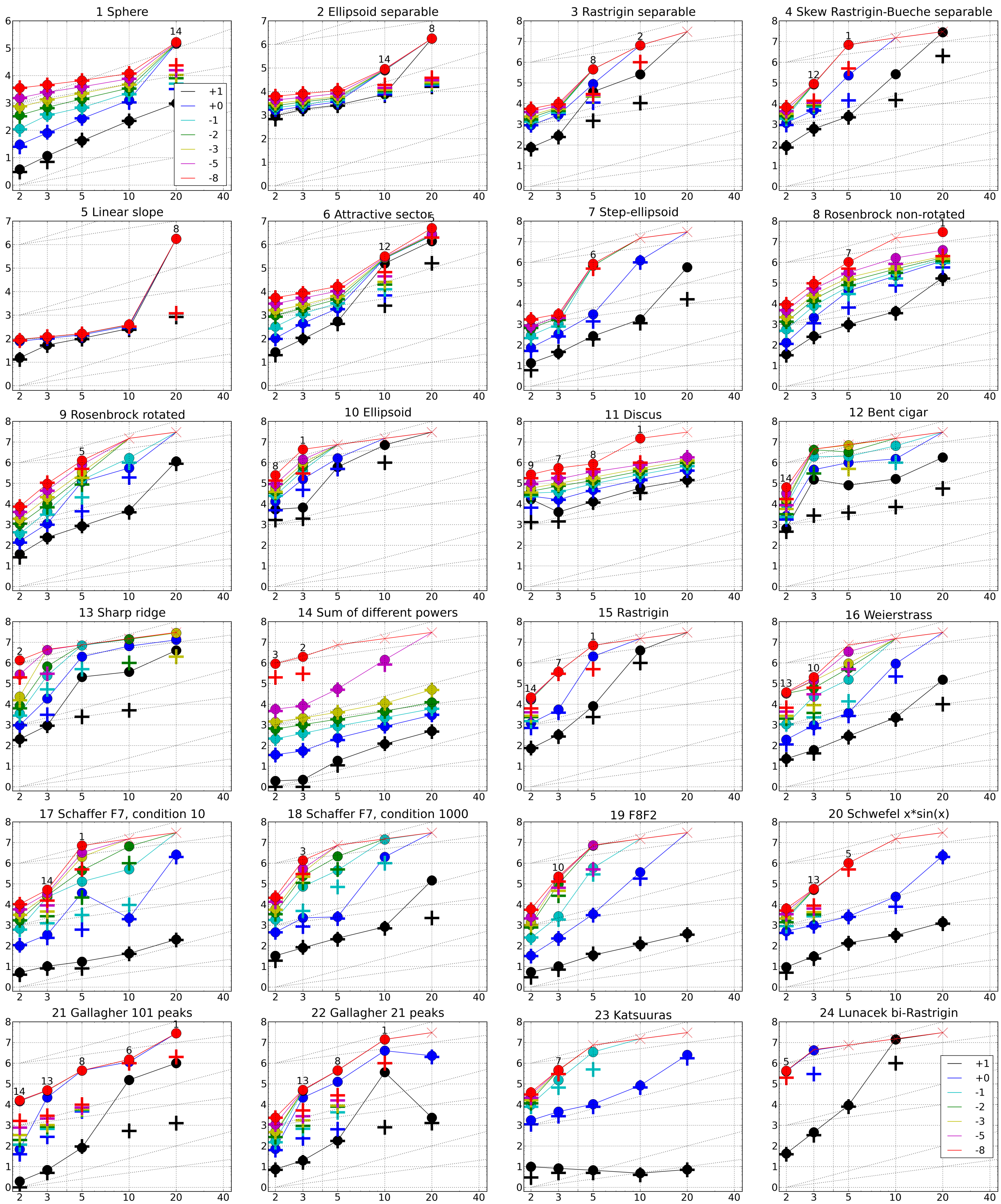

Figure 2: Expected Running Time (ERT, ●) to reach $f_{\mathrm{opt}} + \Delta f$ and median number of function evaluations of successful trials (+), shown for $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log 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 ($\times$) 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.

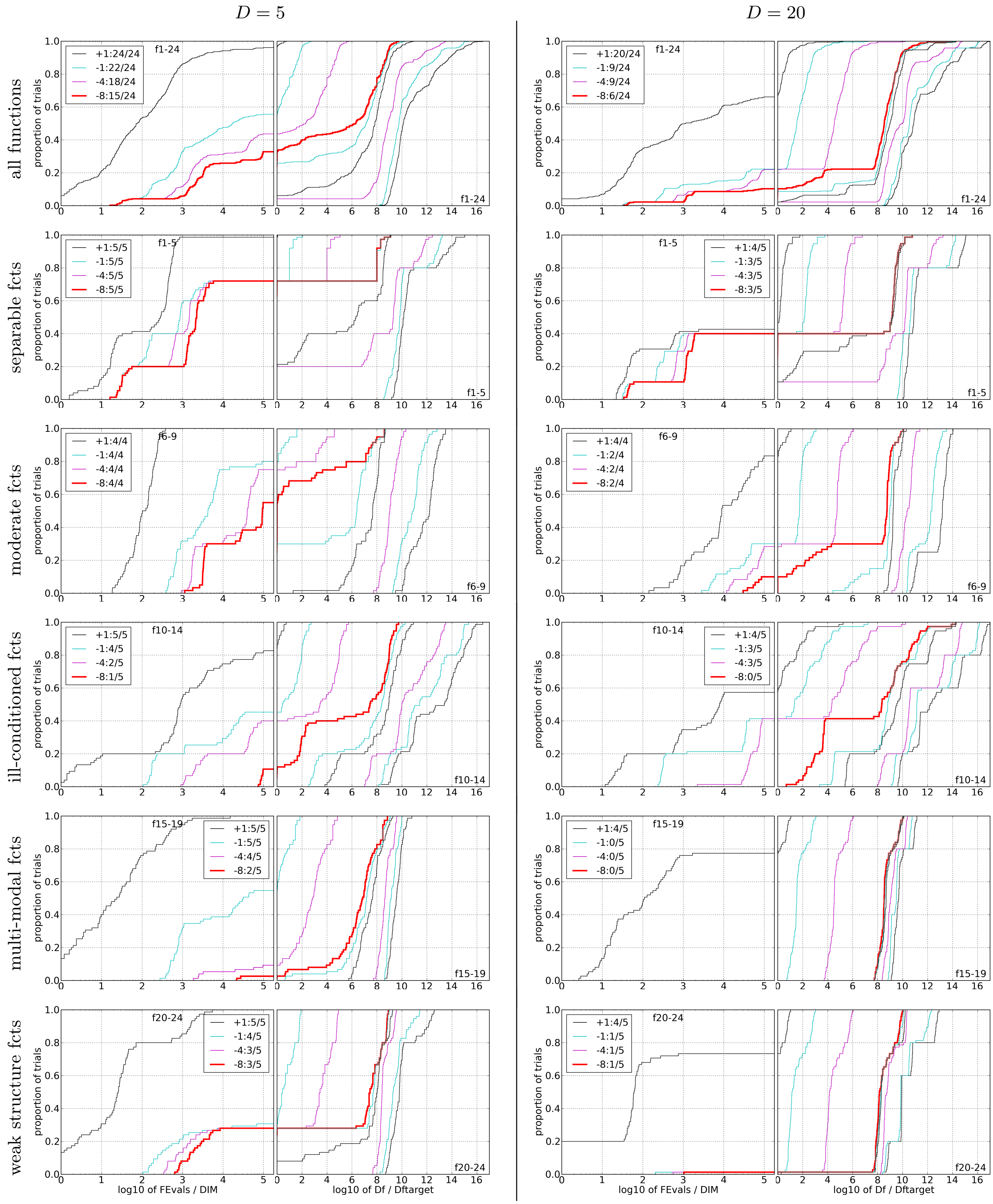

Figure 3: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or Δf. Left subplots: ECDF of the running time (number of function evaluations), divided by search space dimension D, to fall below $f_{\text{opt}} + \Delta f$ with $\Delta f = 10^k$,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