

Black-Box Optimization Benchmarking for Noiseless Function Testbed using Harmony Search

Draft version *

Mohammed El-Abd
American University of Kuwait
melabd@auk.edu.kw

ABSTRACT

This paper benchmarks the harmony search (HS) algorithm using the noise-free BBOB 2010 testbed.

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

1. HARMONY SEARCH

HS [6, 10] was introduced as an imitation of the musical process that searches for a harmony balance. In HS, each variable of the problem is regarded as the pitch of a different musical instrument and the complete solution is referred to as a harmony vector. If the pitch (decision variable value) makes good harmony (good objective function value), it is stored in memory.

The memory part is modeled using a memory structure referred to as the Harmony Memory (HM). Initially, HS is initialized with random harmonies. During the algorithm's progress, when a new harmony is found, it is inserted in HM replacing the worst harmony if it is better than it.

In HS, new harmonies are developed by producing a series of new pitches. Each new pitch is produced following one of three rules. Playing one pitch from memory, picking one pitch from memory and adjusting it to play a new pitch,

playing a totally random pitch. These rules are applied using two parameters known as the Harmony Memory Considering Rate (HCMR) and the Pitch Adjusting Rate (PAR). Algorithm 1 shows the basic steps of the HS algorithm.

If the second rule is applied, a random pitch p_i is picked from memory and adjusted according to the following equation:

$$p_j = p_j + \phi \times b, \quad (1)$$

for $j \in \{1 \dots d\}$ where d is the number of dimensions, ϕ is a random number uniformly distributed in the range $[-1,1]$, and b is an arbitrary distance bandwidth.

If a totally random new pitch is to be generated, it is generated randomly in the allowable domain using the following equation:

$$p_j = lb_j + r \times (ub_j - lb_j), \quad (2)$$

where lb_j and ub_j are the lower and upper domains for decision variable j and r is a random number uniformly distributed in the range $[0,1]$.

Algorithm 1 The HS algorithm

Require: Max_Function_Evaluations, memory size, HCMR, PAR, b

- 1: Initialize the harmony memory
- 2: Evaluate the solutions in the memory
- 3: Max_Iterations = Max_Function_Evaluations
- 4: Iter_number=1
- 5: **while** iter_number \leq Max_Iterations **do**
- 6: **for** each dimension d **do**
- 7: **if** $U(0,1) \leq HCMR$ **then**
- 8: Choose a randomly selected pitch from the memory
- 9: **if** $U(0,1) \leq PAR$ **then**
- 10: Adjust the selected pitch
- 11: **end if**
- 12: **else**
- 13: Improvise a new pitch
- 14: **end if**
- 15: **end for**
- 16: Evaluate the new harmony
- 17: **if** the new harmony is better than the worst one in memory **then**
- 18: Replace the worst harmony in memory
- 19: **end if**
- 20: Iter_number = Iter_number + 1
- 21: **end while**
- 22: **return** best harmony

Recent modifications for the HS algorithm include the proposed Improved HS (IHS) [7], in which the parameters PAR and b are not fixed during the algorithm's progress. PAR

*Submission deadline: March 25th.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GECCO'10, July 7–11, 2010, Portland Oregon, United States of America.
Copyright 2010 ACM X-XXXXXX-XX-X/XX/XX ...\$10.00.

is increased linearly with the iterations while b is decreased in a logarithmic approach. IHS was shown to produce good results, when compared to other algorithms, for some constrained optimization applications including the minimization of the weight of a spring, pressure vessel design and welded beam design.

Another modification is the Global-best HS (GHS) proposed by [8]. The modification was proposed to imitate how the particles in PSO follow the global best in the swarm. The pitch adjustment step is updated so that the new harmony can mimic the best harmony in memory. The GHS was shown to outperform HS and IHS on almost all of the classical benchmark functions tested. GHS also remained as the best performer under the influence of noise or increasing dimensionality.

The Differential HS (DHS) was reported in [1]. In DHS, the pitch adjustment operation is replaced by a mutation strategy borrowed from DE. It was mathematically shown that DHS under certain circumstances can have more population variance over the generations when compared to the classical HS. When applied to a set of classical benchmark functions, DHS outperformed the classical HS, IHS and GHS in terms of the solution reached, the speed of convergence and the robustness.

1.1 Parameter Tuning

HS code was obtained by private communication with Z. W. Geem. The parameters were set as $HMS = 5$, $HCMR = 0.9$ and $PAR = 0.3$ following [8]. The value of b was reduced from 0.01 to 0.001 as it provided better results in earlier experiments conducted on the CEC05 benchmarks [9]. The HS algorithm is allowed to perform a maximum of $10^5 \times D$ function evaluations for all test functions and no restart mechanism was used.

2. RESULTS

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

3. CPU TIMING EXPERIMENT

For the timing experiment, HS was run on f8 and restarted until at least 30 seconds had passed (according to Figure 2 in [4]). The experiments have been conducted with an Intel Core 2 Quad 2.4 GHz under Windows Vista using the MATLAB-code provided. The results were 3.1×10^{-4} seconds per function evaluation in dimensions 2 up to 20. A dependency of CPU time on the search space dimensionality is not visible.

4. REFERENCES

- [1] P. Chakraborty, G. G. Roy, S. Das, and D. Jain. An improved harmony search algorithm with differential mutation operator. *Fundamenta Informaticae*, 95:1–26, 2009.
- [2] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2010.
- [3] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking

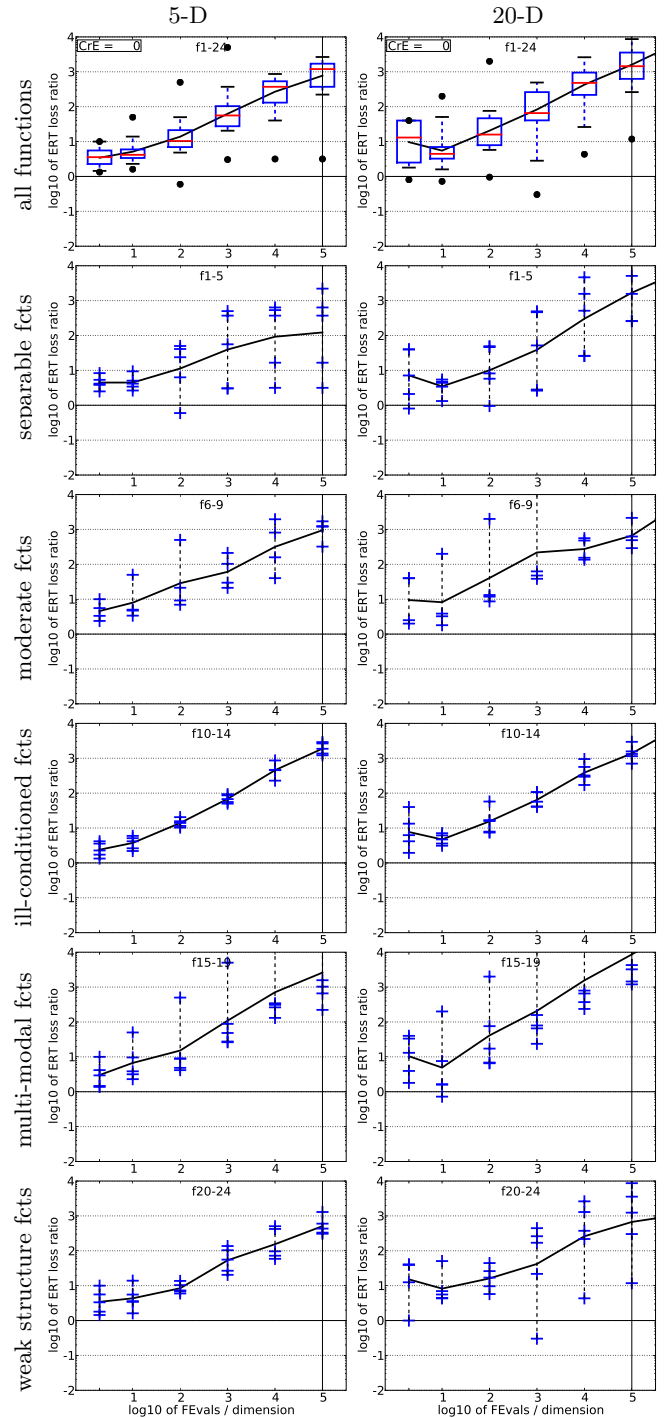


Figure 3: ERT loss ratio versus given budget FEvals. The target value f_t for ERT (see Figure 1) is the smallest (best) recorded function value such that $ERT(f_t) \leq FEvals$ for the presented algorithm. Shown is FEvals divided by the respective best $ERT(f_t)$ from BBOB-2009 for functions f_1 – f_{24} in 5-D and 20-D. Each ERT is multiplied by $\exp(CrE)$ correcting for the parameter crafting effort. 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 this function subset.

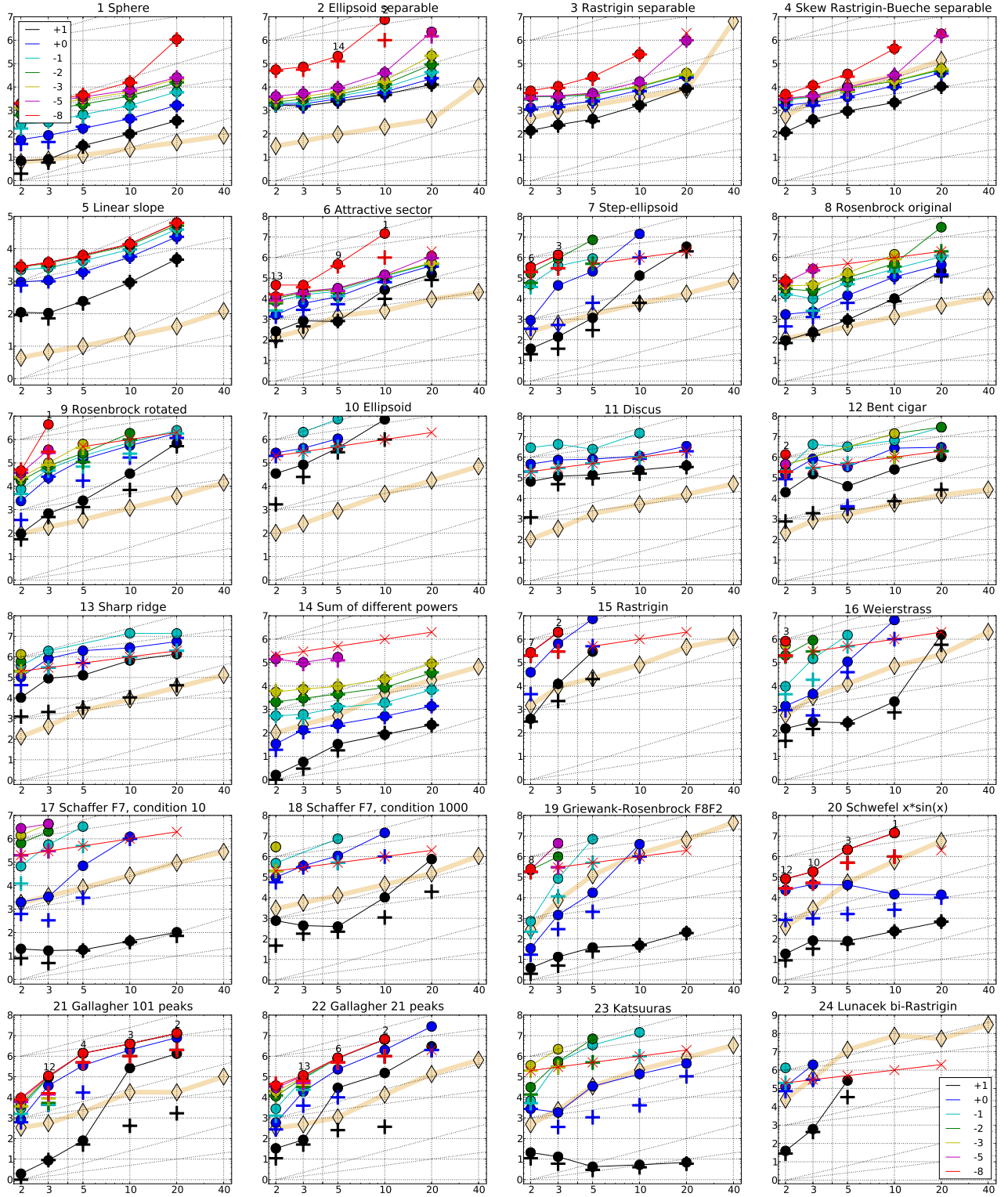


Figure 1: Expected Running Time (ERT, ●) to reach $f_{\text{opt}} + \Delta f$ and median number of f -evaluations from successful trials (+), for $\Delta f = 10^{\{+1, 0, -1, -2, -3, -5, -8\}}$ (the exponent is given in the legend of f_1 and f_{24}) versus dimension in log-log presentation. For each function and dimension, $\text{ERT}(\Delta f)$ equals to $\#FEs(\Delta f)$ divided by the number of successful trials, where a trial is successful if $f_{\text{opt}} + \Delta f$ was surpassed. The $\#FEs(\Delta f)$ are the total number (sum) of f -evaluations while $f_{\text{opt}} + \Delta f$ was not surpassed in the trial, from all (successful and unsuccessful) trials, and f_{opt} is the optimal function value. Crosses (×) indicate the total number of f -evaluations, $\#FEs(-\infty)$, divided by the number of trials. Numbers above ERT-symbols indicate the number of successful trials. Y-axis annotations are decimal logarithms. The thick light line with diamonds shows the single best results from BBOB-2009 for $\Delta f = 10^{-8}$. Additional grid lines show linear and quadratic scaling.

f1 in 5-D, N=15, mFE=5813						f1 in 20-D, N=15, mFE=1.39e6						f2 in 5-D, N=15, mFE=500000						f2 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	15	3.2e1	6.0e0	6.4e1	3.2e1	15	3.6e2	2.0e2	4.8e2	3.6e2	10	15	2.4e3	1.2e3	3.9e3	2.4e3	15	1.3e4	1.1e4	1.5e4	1.3e4		
1	15	1.7e2	5.0e1	3.0e2	1.7e2	15	1.6e3	1.4e3	2.0e3	1.6e3	1	15	3.1e3	2.2e3	4.6e3	3.1e3	15	2.3e4	1.6e4	2.7e4	2.3e4		
1e-1	15	6.7e2	2.9e2	1.0e3	6.7e2	15	6.2e3	5.0e3	7.5e3	6.2e3	1e-1	15	3.9e3	2.9e3	4.9e3	3.9e3	15	4.2e4	3.3e4	5.0e4	4.2e4		
1e-3	15	3.0e3	1.5e3	4.0e3	3.0e3	15	2.2e4	1.9e4	2.5e4	2.2e4	1e-3	15	6.1e3	4.3e3	8.3e3	6.1e3	15	2.2e5	1.4e5	3.3e5	2.2e5		
1e-5	15	3.7e3	2.0e3	4.9e3	3.7e3	15	2.6e4	2.4e4	2.9e4	2.6e4	1e-5	15	9.3e3	6.7e3	1.2e4	9.3e3	10	2.3e6	1.0e6	4.8e6	1.3e6		
1e-8	15	4.5e3	2.8e3	5.5e3	4.5e3	15	1.1e6	8.1e5	1.4e6	1.1e6	1e-8	14	2.1e5	5.2e4	3.6e5	1.7e5	0	58e-7	35e-7	17e-6	2.0e6		
f3 in 5-D, N=15, mFE=82140						f3 in 20-D, N=15, mFE=2.00e6						f4 in 5-D, N=15, mFE=78670						f4 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	15	4.3e2	2.2e2	7.5e2	4.3e2	15	8.8e3	6.5e3	1.2e4	8.8e3	10	15	9.5e2	4.6e2	1.5e3	9.5e2	15	1.1e4	7.2e3	1.3e4	1.1e4		
1	15	2.5e3	1.7e3	3.0e3	2.5e3	15	2.7e4	2.1e4	3.7e4	2.7e4	1	15	3.8e3	2.3e3	6.6e3	3.8e3	15	4.0e4	2.6e4	6.2e4	4.0e4		
1e-1	15	4.6e3	2.8e3	1.1e4	4.6e3	15	3.6e4	2.2e4	6.3e4	3.6e4	1e-1	15	8.6e3	2.9e3	1.8e4	8.6e3	15	5.4e4	3.0e4	8.0e4	5.4e4		
1e-3	15	5.1e3	3.2e3	5.9e3	5.1e3	15	3.9e4	2.6e4	5.9e4	3.9e4	1e-3	15	9.0e3	3.2e3	1.7e4	9.0e3	15	6.1e4	3.4e4	8.4e4	6.1e4		
1e-5	15	5.5e3	3.6e3	6.7e3	5.5e3	15	9.5e5	4.5e5	1.5e6	9.5e5	1e-5	15	9.7e3	3.7e3	1.9e4	9.7e3	12	1.9e6	1.1e6	3.5e6	1.4e6		
1e-8	15	2.8e4	1.4e4	4.2e4	2.8e4	0	37e-7	20e-7	56e-7	2.0e6	1e-8	15	3.6e4	1.6e4	7.9e4	3.6e4	0	63e-7	31e-7	17e-6	2.0e6		
f5 in 5-D, N=15, mFE=11740						f5 in 20-D, N=15, mFE=83550						f6 in 5-D, N=15, mFE=500000						f6 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	15	2.4e2	1.3e2	6.6e2	2.4e2	15	4.9e3	3.5e3	6.4e3	4.9e3	10	15	7.9e2	4.5e2	1.2e3	7.9e2	15	1.6e5	1.2e4	3.3e5	1.6e5		
1	15	1.9e3	8.3e2	2.6e3	1.9e3	15	2.3e4	1.9e4	2.7e4	2.3e4	1	15	1.2e4	1.8e3	4.0e4	1.2e4	15	4.0e5	1.5e5	6.5e5	4.0e5		
1e-1	15	4.2e3	3.0e3	6.0e3	4.2e3	15	4.0e4	3.2e4	4.6e4	4.0e4	1e-1	15	2.3e4	4.8e3	6.0e4	2.3e4	15	5.1e5	2.4e5	7.7e5	5.1e5		
1e-3	15	6.3e3	4.3e3	8.6e3	6.3e3	15	6.1e4	5.1e4	7.5e4	6.1e4	1e-3	15	3.1e4	8.1e3	7.0e4	3.1e4	15	5.6e5	2.5e5	8.1e5	5.6e5		
1e-5	15	6.4e3	4.9e3	8.5e3	6.4e3	15	6.3e4	5.2e4	7.5e4	6.3e4	1e-5	15	3.3e4	1.2e4	6.7e4	3.3e4	13	1.2e6	5.5e5	2.6e6	8.9e5		
1e-8	15	6.4e3	4.4e3	8.5e3	6.4e3	15	6.3e4	5.2e4	7.6e4	6.3e4	1e-8	9	4.7e5	6.2e4	1.0e6	1.4e5	0	40e-7	31e-7	11e-6	2.0e6		
f7 in 5-D, N=15, mFE=500000						f7 in 20-D, N=15, mFE=2.00e6						f8 in 5-D, N=15, mFE=500000						f8 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	15	1.2e3	2.1e1	5.5e3	1.2e3	6	3.4e6	3.3e4	8.8e6	3.7e5	10	15	9.3e2	5.4e2	2.0e3	9.3e2	15	2.2e5	4.0e4	5.4e5	2.2e5		
1	11	2.1e5	5.7e2	6.5e5	3.0e4	0	12e+0	57e-1	17e+0	6.2e5	1	15	1.4e4	1.8e3	3.3e4	1.4e4	15	4.6e6	6.8e4	1.2e6	4.6e5		
1e-1	6	8.9e5	1.0e5	1.9e6	1.4e5	1e-1	15	6.3e4	3.4e3	1.6e5	6.3e4	14	1.1e6	9.1e4	1.9e6	9.6e5		
1e-3	0	17e-2	13e-3	45e-1	1.0e5	1e-3	15	1.8e5	1.0e5	3.1e5	1.8e5	0	31e-3	13e-3	96e-3	2.0e6		
1e-5	1e-5	0	47e-6	34e-6	12e-5	5.0e5		
1e-8	1e-8		
f9 in 5-D, N=15, mFE=500000						f9 in 20-D, N=15, mFE=2.00e6						f10 in 5-D, N=15, mFE=500000						f10 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	15	2.4e3	4.7e2	4.1e3	2.4e3	14	6.9e5	4.2e5	1.0e6	5.5e5	10	11	4.4e5	1.9e5	8.3e5	2.5e5	0	31e+1	70e+0	57e+1	2.0e6		
1	12	1.4e5	2.4e3	5.2e5	1.7e4	11	1.9e6	1.1e6	5.0e6	1.1e6	1	6	1.1e6	3.3e5	2.3e6	3.3e5		
1e-1	12	2.0e5	2.7e4	5.4e5	7.6e4	11	2.5e6	1.6e6	3.8e6	1.7e6	1e-1	1	7.5e6	9.7e5	1.5e7	4.7e5		
1e-3	9	6.4e5	1.7e5	1.6e6	3.1e5	0	80e-3	70e-3	41e-1	2.0e6	1e-3	0	20e-1	16e-2	29e+0	5.0e5		
1e-5	0	87e-5	42e-6	39e-1	5.0e5	1e-5		
1e-8	1e-8		
f11 in 5-D, N=15, mFE=500000						f11 in 20-D, N=15, mFE=2.00e6						f12 in 5-D, N=15, mFE=500000						f12 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	14	1.4e5	1.6e4	4.5e5	1.0e5	15	3.9e5	2.1e5	8.5e5	3.9e5	10	14	3.9e4	2.0e3	4.1e3	3.2e3	10	1.0e6	2.4e4	2.0e6	2.6e4		
1	7	8.0e5	1.1e5	2.1e6	2.3e5	8	3.5e6	1.3e6	7.8e6	1.7e6	1	9	3.4e5	2.1e3	1.0e6	3.2e3	6	3.0e6	2.7e4	7.0e6	3.6e4		
1e-1	3	2.5e6	4.7e5	5.0e6	4.6e5	0	95e-2	64e-2	29e-1	2.0e6	1e-1	2	3.3e6	3.6e3	8.0e6	3.3e3	1	2.8e7	2.1e6	6.5e7	1.4e5		
1e-3	0	10e-1	76e-3	48e-1	5.0e5	1e-3	0	65e-2	97e-3	98e-1	5.0e5	0	14e-1	24e-2	31e+0	2.0e6		
1e-5	1e-5		
1e-8	1e-8		
f13 in 5-D, N=15, mFE=500000						f13 in 20-D, N=15, mFE=2.00e6						f14 in 5-D, N=15, mFE=500000						f14 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	12	1.3e5	2.1e3	5.1e5	3.5e3	9	1.4e6	3.0e4	6.0e6	3.6e4	10	15	3.3e1	5.0e0	7.2e1	3.3e1	15	2.2e2	1.1e2	2.9e2	2.2e2		
1	3	2.0e6	7.1e3	3.5e6	5.6e3	4	5.5e6	5.6e4	1.3e7	4.9e4	1	15	2.3e2	1.4e2	4.0e2	2.3e2	15	1.4e3	8.3e2	1.9e3	1.4e3		
1e-1	0	18e-1	34e-2	33e+0	4.2e5	2	1.4e7	1.8e6	3.1e7	7.8e5	1e-1	15	1.2e3	2.9e2	1.9e3	1.2e3	15	6.9e3	4.4e3	9.9e3	6.9e3		
1e-3	0	52e-1	92e-3	28e+0	2.0e6	1e-3	15	9.7e3	4.9e3	1.4e4	9.7e3	15	9.0e4	8.2e4	1.0e5	9.0e4		
1e-5	1e-5	15	1.7e5	9.8e4	3.0e5	1.7e5	0	11e-5	10e-5	12e-5	2.0e6		
1e-8	1e-8	0	30e-7	14e-7	62e-7	5.0e5		
f15 in 5-D, N=15, mFE=500000						f15 in 20-D, N=15, mFE=2.00e6						f16 in 5-D, N=15, mFE=500000						f16 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	10	3.0e5	6.1e2	1.0e6	4.6e4	0	94e+0	60e+0	13e+1	2.0e6	10	15	2.7e2	9.9e1	4.4e2	2.7e2	9	1.5e6	7.8e3	6.0e6	2.1e5		
1	1	7.3e6	8.4e5	1.8e7	3.4e5	1	14	1.1e5	2.1e3	3.3e5	7.4e4	0	83e-1	41e-1	12e+0	1.9e6		
1e-1	0	40e-1	20e-1	18e+0	2.2e5	1e-1	4	1.5e6	7.0e3	3.0e6	1.1e5		
1e-3	1e-3	0	41e-2	60e-3	92e-2	4.9e5		
1e-5	1e-5		
1e-8	1e-8		
f17 in 5-D, N=15, mFE=500000						f17 in 20-D, N=15, mFE=2.00e6						f18 in 5-D, N=15, mFE=500000						f18 in 20-D, N=15, mFE=2.00e6					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
10	15	1.8e1	8.0e0	3.9e1	1.8e1	15	1.0e2	3.9e1	1.9e2	1.0e2	10	15	4.0e2	3.9e1	9.8e2	4.0e2	1	7.5e5	4.8e2	2.0e6	2.3e4		
1	14	7.1e4	6.1e2	4.4e5	3.5e4	0	29e-1	17e-1	46e-1	2.0e6	1	5											

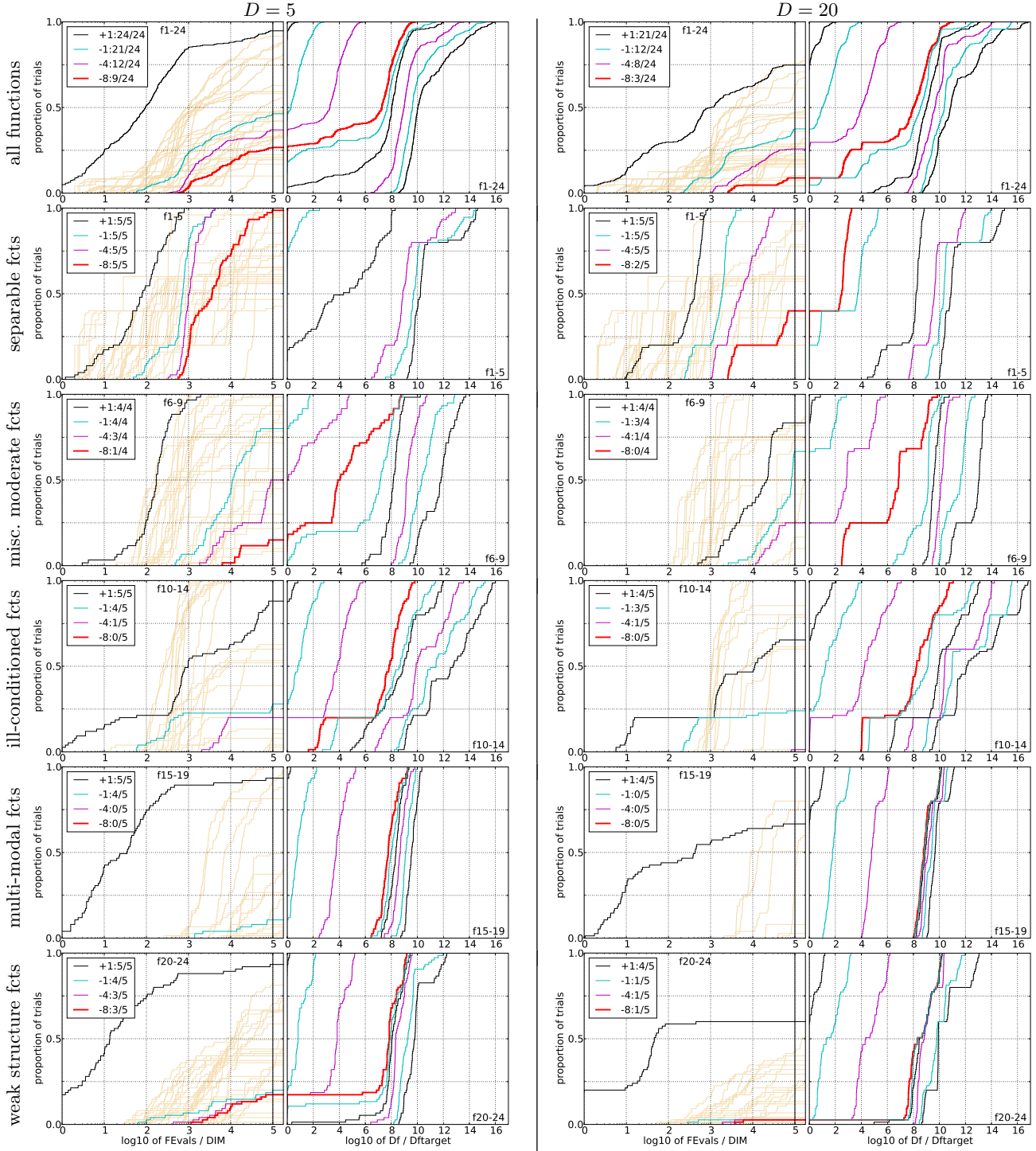


Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left subplots) or versus Δf (right subplots). The thick red line represents the best achieved results. Left subplots: ECDF of the running time (number of function evaluations), divided 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 $D, 10D, 100D \dots$ function evaluations (from right to left cycling black-cyan-magenta). 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. Light brown lines in the background show ECDFs for target value 10^{-8} of all algorithms benchmarked during BBOB-2009.

Table 2: ERT loss ratio (see Figure 3) compared to the respective best result from BBOB-2009 for budgets given in the first column. The last row RL_{US}/D gives the number of function evaluations in unsuccessful runs divided by dimension. Shown are the smallest, 10%-ile, 25%-ile, 50%-ile, 75%-ile and 90%-ile value (smaller values are better).

$f1-f24$ in 5-D, maxFE/D=100000						
#FEs/D	best	10%	25%	med	75%	90%
2	1.3	1.4	2.0	3.5	5.5	10
10	1.6	2.3	3.3	4.2	5.8	18
100	0.60	4.8	6.9	10	21	95
1e3	3.0	19	27	56	1.0e2	3.8e2
1e4	3.2	38	1.1e2	3.6e2	5.2e2	9.7e2
1e5	3.2	2.0e2	3.5e2	1.1e3	1.6e3	2.7e3
RL_{US}/D	1e5	1e5	1e5	1e5	1e5	1e5
$f1-f24$ in 20-D, maxFE/D=100000						
#FEs/D	best	10%	25%	med	75%	90%
2	0.81	1.7	2.3	13	40	40
10	0.72	1.6	3.2	4.4	6.6	66
100	0.95	5.8	7.6	15	45	2.7e2
1e3	0.30	2.8	39	64	2.2e2	2.4e3
1e4	4.3	26	1.9e2	4.3e2	8.7e2	2.8e3
1e5	12	2.6e2	5.6e2	1.4e3	3.4e3	1.0e4
1e6	18	1.5e3	2.9e3	6.6e3	1.2e4	3.5e4
RL_{US}/D	1e5	1e5	1e5	1e5	1e5	1e5

2010: Experimental setup. Technical Report RR-7215, INRIA, 2010.

- [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] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2010: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2010.
- [6] K. S. Lee and Z. W. Geem. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Computer Methods in Applied Mechanics Engineering*, 194:3902–3933, 2005.
- [7] M. Mahdavi, M. Fesanghary, and E. Damangir. An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, 188:1567–1579, 2007.
- [8] M. G. H. Omran and M. Mahdavi. Global-best harmony search. *Applied Mathematics and Computation*, 198:643–656, 2008.
- [9] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization. Technical Report 2005005, ITT Kanpur, India, 2005.
- [10] X. Yang. Harmony search as a metaheuristic algorithm. In Z. W. Geem, editor, *Music-Inspired Harmony Search Algorithm, SCI 191*, pages 1–14. Springer-Verlag:Berlin Heidelberg, 2009.